

Copyright
by
Lu Xia
2014

The Dissertation Committee for Lu Xia
certifies that this is the approved version of the following dissertation:

Recognizing Human Activity Using RGBD Data

Committee:

J.K. Aggarwal , Supervisor

Kristen Grauman

Constantine Caramanis

Inderjit Dhillon

Dana Ballard

Wilson Geisler

Recognizing Human Activity Using RGBD Data

by

Lu Xia, B.E.; M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2014

Dedicated to my parents.

Acknowledgments

First of all, I would like to express my deepest gratitude to my advisor, Prof. J.K. Aggarwal, for giving me the chance to work in this interesting area. I sincerely appreciate his advice and guidance, his continuous support, his encouragement when I am frustrated, and his trust in me. I feel very lucky that he granted me the opportunity to join his group and work on 3D vision in 2010. His wisdom of foreseeing promising topics has offered me good opportunities to produce novel research.

I would like to express my sincere thanks to Dr. Kristen Grauman. I learned from her not only the foundation knowledge and state-of-the-art research, but also her dedication and passion toward research. She gave me many helpful and insightful advice on my research. I really admire her sharpness in research. She is like a role model to me.

I want to thank my committee members, Dr. Wilson Geilso, Dr. Constantine Caramanis, Dr. Dhillon Inderjit, and Dr. Dana Ballard. The courses I have taken from them laid important foundations of my research, and broadened my eyes to many interesting and related research. I thank all of them for the insightful comments and feedback which strengthened this thesis.

I would also like to thank my fellow lab mates and collaborators: Dr. Chia-Chih Chen, Dr. Michael Ryoo, Dr. Changbo Hu, Dr. Josh Harguess, Dr.

Jong Taek Lee, Birgi Tamersoy, Shaohua Wan, and Ilaria Gori. They offered me many helping hands and shared valuable experiences with me. Special thanks go to Ms. Selina Keilani for helping me to revise my paper drafts for conferences and journals.

I feel lucky to have incredibly supportive friends in Texas, California, New York, Washington, Pennsylvania, Beijing and elsewhere. The courage and inspiration they gave enabled me to walk through all the difficulties.

Finally, I owe my thanks to my parents and my brother. Without their continuous support, encourage, and love, this would not have been possible. I attribute all my success to them.

Recognizing Human Activity Using RGBD Data

Publication No. _____

Lu Xia, Ph.D.

The University of Texas at Austin, 2014

Supervisor: J.K. Aggarwal

Traditional computer vision algorithms try to understand the world using visible light cameras. However, there are inherent limitations of this type of data source. First, visible light images are sensitive to illumination changes and background clutter. Second, the 3D structural information of the scene is lost when projecting the 3D world to 2D images. Recovering the 3D information from 2D images is a challenging problem. Range sensors have existed for over thirty years, which capture 3D characteristics of the scene. However, earlier range sensors were either too expensive, difficult to use in human environments, slow at acquiring data, or provided a poor estimation of distance. Recently, the easy access to the RGBD data at real-time frame rate is leading to a revolution in perception and inspired many new research using RGBD data.

I propose algorithms to detect persons and understand the activities using RGBD data. I demonstrate the solutions to many computer vision

problems may be improved with the added depth channel. The 3D structural information may give rise to algorithms with real-time and view-invariant properties in a faster and easier fashion. When both data sources are available, the features extracted from the depth channel may be combined with traditional features computed from RGB channels to generate more robust systems with enhanced recognition abilities, which may be able to deal with more challenging scenarios.

As a starting point, the first problem is to find the persons of various poses in the scene, including moving or static persons. Localizing humans from RGB images is limited by the lighting conditions and background clutter. Depth image gives alternative ways to find the humans in the scene. In the past, detection of humans from range data is usually achieved by tracking, which does not work for indoor person detection. In this thesis, I propose a model based approach to detect the persons using the structural information embedded in the depth image. I propose a 2D head contour model and a 3D head surface model to look for the head-shoulder part of the person. Then, a segmentation scheme is proposed to segment the full human body from the background and extract the contour. I also give a tracking algorithm based on the detection result.

I further research on recognizing human actions and activities. I propose two features for recognizing human activities. The first feature is drawn from the skeletal joint locations estimated from a depth image. It is a compact representation of the human posture called histograms of 3D joint locations

(HOJ3D). This representation is view-invariant and the whole algorithm runs at real-time. This feature may benefit many applications to get a fast estimation of the posture and action of the human subject.

The second feature is a spatio-temporal feature for depth video, which is called Depth Cuboid Similarity Feature (DCSF). The interest points are extracted using an algorithm that effectively suppresses the noise and finds salient human motions. DCSF is extracted centered on each interest point, which forms the description of the video contents. This descriptor can be used to recognize the activities with no dependence on skeleton information or pre-processing steps such as motion segmentation, tracking, or even image de-noising or hole-filling. It is more flexible and widely applicable to many scenarios.

Finally, all the features herein developed are combined to solve a novel problem: first-person human activity recognition using RGBD data. Traditional activity recognition algorithms focus on recognizing activities from a third-person perspective. I proposed to recognize activities from a first-person perspective with RGBD data. This task is very novel and extremely challenging due to the large amount of camera motion either due to self exploration or response of the interaction. I extracted 3D optical flow features as the motion descriptors, 3D skeletal joints features as posture descriptors, spatio-temporal features as local appearance descriptors to describe the first-person videos. To address the ego-motion of the camera, I proposed an attention mask to guide the recognition procedures and separate the features on the ego-motion region

and independent-motion region. The 3D features are useful at summarizing the discerning information of the activities. In addition, the combination of the 3D features with existing 2D features brings more robust recognition results and make the algorithm capable of dealing with more challenging cases.

Table of Contents

Acknowledgments	v
Abstract	vii
List of Tables	xv
List of Figures	xvii
Chapter 1. Introduction	1
1.1 Overview	1
1.1.1 Overview of 3D Sensing	1
1.1.2 Human Detection	3
1.1.3 Human Activity Recognition	6
1.1.4 First-Person Activity Recognition	8
1.1.5 Overview of My Work	10
1.2 Main Contributions	14
1.3 Road Map	16
Chapter 2. Related Work	17
2.1 Human Detection	17
2.2 Activity Recognition	20
2.2.1 Recognition From 3D Silhouettes	21
2.2.2 Recognition From Skeletal Joints or Body Parts Tracking	23
2.2.3 Recognition Using Local Spatio-Temporal Features	26
2.2.4 Recognition Using Local 3D Occupancy Features	28
2.2.5 Recognition From 3D Optical Flow	30
2.3 First-Person Activity Recognition	32

Chapter 3. Human Detection Using a Single Depth Image	34
3.1 Algorithm	35
3.1.1 Preprocessing	35
3.1.2 Regression on the Diameter of the Head	35
3.1.3 2D Template Matching	36
3.1.3.1 2D Chamfer Distance Matching	37
3.1.3.2 Depth Guided Pyramid Matching	39
3.1.4 3D Model Fitting	40
3.1.4.1 Head Radius	41
3.1.4.2 Occlusion Mask	42
3.1.4.3 3D Model Fitting	43
3.1.5 Extract Contours	44
3.1.6 Tracking	47
3.2 Experimental Results	49
3.2.1 Datasets	49
3.2.2 Detection Results	50
3.2.3 Tracking Results	54
3.3 Conclusion	54
Chapter 4. Histogram of Skeletal Joint Feature for Action Recognition	58
4.1 Algorithm	60
4.1.1 Body Part Inference and Joint Position Estimation	60
4.1.2 HOJ3D as Posture Representation	61
4.1.2.1 Spherical Coordinates of Histogram	62
4.1.2.2 Probabilistic Voting	62
4.1.2.3 Feature Extraction	64
4.1.3 Vector Quantization	64
4.1.4 Action Recognition Using Discrete HMMs	65
4.2 Experiments	66
4.2.1 Data	66
4.2.2 Experimental Results	71
4.3 Conclusion	72

Chapter 5. Spatio-Temporal Depth Cuboid Similarity Feature for Action Recognition	74
5.1 Algorithm	75
5.1.1 DSTIP Detection	75
5.1.1.1 Spatio-Temporal Filtering	75
5.1.1.2 Noise Suppression	76
5.1.1.3 Interest Point Extraction	79
5.1.2 Interest Point Description	79
5.1.2.1 Adaptable Supporting Size	80
5.1.3 Action Description	83
5.1.3.1 Cuboid Codebook	83
5.1.3.2 Mining Discriminative Feature Pool	84
5.2 Experimental Results	85
5.2.1 MSRAction3D Dataset	85
5.2.2 MSRDailyActivity3D Dataset	86
5.2.3 UTKinect Dataset	90
5.3 Conclusion	92
Chapter 6. First-Person Activity Recognition Using RGBD Data	94
6.1 Algorithm	95
6.1.1 Feature Extraction	95
6.1.1.1 Motion Descriptor	95
6.1.1.2 Local Appearance Descriptors	98
6.1.1.3 Human Posture Descriptor	99
6.1.2 Separating Ego-Motion from Independent-Motion	100
6.1.2.1 Independent Motion Vectors	101
6.1.2.2 Attention Mask	102
6.1.3 Multiple Channel Kernels	104
6.2 Experiments	105
6.2.1 Dataset	106
6.2.2 2D vs 3D	109
6.2.2.1 Optical Flow	109
6.2.2.2 STIP	110

6.2.3 Mask	111
6.2.4 Single Features vs Concatenated	112
6.3 Conclusion	113
Chapter 7. Conclusion	116
Bibliography	120

List of Tables

3.1	Accuracy of our algorithm on the two datasets	51
3.2	Comparison of performance	53
4.1	The mean and standard deviation of the sequence lengths measured by number of frames at 30 fps.	66
4.2	Recognition rate of each action type	68
4.3	Comparisons on the UTKinect dataset	68
4.4	The three subsets of actions used for the MSR Action3D dataset.	69
4.5	Recognition results of our algorithm on the MSRAction3D dataset, compared with Li et al. [49] and Yang et al. [100] . In test one, 1/3 of the samples were used as training samples and the rest as testing samples. In test two, 2/3 samples were used as training samples. In the cross subject test, half of the subjects were used as training and the rest of the subjects were used as testing.	69
5.1	Comparison of accuracy on MSRAction3D dataset.	86
5.2	Comparison of recognition accuracy on MSRDailyActivity3D dataset.	88
5.3	Comparison of recognition rate on our own dataset. In test one , 1/3 of the samples were used as training samples and the rest as testing samples; in test two , 2/3 samples were used as training samples; In cross subject test , half of the subjects were used as training and the rest as testing.	92
6.1	Sample images of 9 activities in our humanoid first-person dataset. The first row presents RGB images. The second row shows depth images. The last row represents skeleton images. If no skeleton is detected for a particular frame, a black image is shown.	108
6.2	The table illustrates the comparison between $2D$ and $3D$ optical flow descriptors.	109
6.3	Comparison of results for spatio-temporal features.	111
6.4	This table illustrates the comparison between raw descriptors and features extracted using the attention mask.	111

6.5	All the comparisons are illustrated. The first rows are dedicated to the results obtained using state-of-the-art methods. Following that are the performance of the descriptors we have investigated. The results of the proposed features applying the attention mask are shown starting from the fifth row: 3D optical flow features (OF), combination of depth and intensity spatio-temporal features (ST), posture descriptor (P), and different combinations. The last row indicates our best results, attained using the combination of all the features together.	114
-----	--	-----

List of Figures

1.1	Sample RGB image frames from our first-person dataset. . . .	9
3.1	Regression curve of the relation between head diameter (in pixels) and depth value (in millimeters)	36
3.2	Intermediate results of 2D Chamfer distance matching. (a) Pre-processed depth image. (b) Binary edge image calculated by Canny edge detector. (c) Distance map generated from binary edge image (d). The binary head template (e). 2D template matching result (yellow dots indicate the detected locations). .	41
3.3	(a) Input of the 3D fitting algorithm. (b) Fitting process. (c) Output of 3D model fitting. Yellow dots indicate detection points.	44
3.4	(a) Original depth array. Some parts of the body are merged with the ground plane and wall. (b) The input depth array to the region growing algorithm. The ground plane is delineated by the thresholded F filter response. The edges along the feet well separate the persons from the floor.	45
3.5	(a) Result of the region growing algorithm. (b) The extracted whole body contours superimposed on the depth map.	48
3.6	Example images from datasets. Images in the upper two rows are from dataset 1; images in the bottom two rows are from dataset 2.	50
3.7	Examples of the human detection result.	51
3.8	Failure cases.	52
3.9	Detection Results using HOG pedestrian detection algorithm.	55
3.10	Tracking result. Results are shown at every two frames. The original frame rate is about 0.4 spf. The two tracked persons are labeled number one and number two respectively.	56
4.1	(a) Depth image. (b). Skeletal joints locations.	59
4.2	(a) Reference coordinates the HOJ3D. (b) Modified spherical coordinate system for joint location binning.	59
4.3	Voting using a Gaussian weight function.	61
4.4	Example of the HOJ3D of a posture.	61

4.5	Sample images from videos of the 10 activities in the database. Note only depth images are used in the proposed algorithm. Action type from left to right, top to bottom: <i>walk, stand up, sit down, pick up, carry, throw, push, pull, wave hands, clap hands</i>	67
4.6	Different views of the actions are presented in the dataset. . .	67
4.7	The variations of subjects performing the same action.	68
5.1	Temporal evolution of pixel values at different locations in the scene.	77
5.2	DSTIPs projected onto x - y dimensions on top of one frame of the video <i>drink</i>	78
5.3	Illustration of extracting DCSF from depth video	79
5.4	Example of STIPs extracted using our algorithm. They are projected onto x - y dimensions with one depth frame from the video for display. Action type from left to right, up to down: <i>drink-sit, eat, drink-stand, call cellphone, play guitar, sit down, stand up, toss, walk and lay-down</i>	88
5.5	Comparison of our DSTIP detector with Cuboid detector. Example video is action drink. Column from left to right is taken $N_p = 50, 100, 200, 300, 500, 800$ respectively.	90
5.6	STIPs extracted using Harris3D detector [44]	90
5.7	Parameter evaluation around optimum value on the MSRDailyActivity3D dataset. The average accuracy with the standard deviation denoted by error bar is plotted.	91
6.1	Sample frames from the dataset with the extracted features overlaid. The first row shows down-sampled dense optical flow features. The second row depicts the STIP cuboids. The third row illustrates the DSTIP cuboids (red), and skeletal joint locations (green) (skeleton feature for the activity in the third row is missing). The sample frames are extracted from run, stand up, and hug activities respectively.	96
6.2	In the first row, the sparse optical flow is depicted. The second row shows the vectors identified as independent motion. The third row presents the attention window used for the STIP features. In the fourth row, the attention mask used for the optical flow features is represented.	115
6.3	Confusion matrices of the four different features on the humanoid first-person dataset.	115

Chapter 1

Introduction

1.1 Overview

1.1.1 Overview of 3D Sensing

The development of computer vision, the use of a camera and a computer to recognize objects began in the early 1960s. It has matured fairly quickly and contributes to the solution of some of the most serious societal problems. Until now, most of the vision algorithms were built on 2D intensity images. However, 3D geometric structure is important for many computer vision applications such as navigation and object search, and it may bring significant improvement to the current computer vision tasks including object and scene recognition, activity analysis, human-computer interaction and robot vision. The acquisition of 3D geometric structure is a difficult problem. There are basically two ways to address it. The first way is to estimate 3D structures from 2D images. Upon seeing a 2D image, a human usually has little difficulty understanding its 3D structure. Thus, there might be cues embedded in the 2D image to infer 3D. However, it is extremely challenging for current computer vision systems to infer 3D structures from 2D images, due to the considerable loss of information when projecting the 3D scene into a 2D image. Such processes require high computational cost and good-quality images. Even

though, there is still difficulty at estimating the 3D structure of texture-less areas. The second way is to capture 3D structure directly from sensors. However, earlier range sensors were either too expensive, difficult to use in human environments, slow at acquiring data, or provided poor estimation of distance. It was very difficult to acquire two dimensional depth images in real-time in the past. Laser scanners give measurements of one point of the scene at a time, multiple co-planar scanners were sometimes used to generate multiple measurements along a 2D line of the scene. Multiview-stereo systems compute depth by comparing a pair of intensity images acquired by two cameras at a certain distance apart. The computational cost for the stereo geometry is high. Such data are usually used to analyze static objects of scenes. To the best of my knowledge, there was no literature addressing activity recognition using depth videos in the past.

The development of range cameras has progressed rapidly over the past decade. Recently, the advent of depth cameras at relatively inexpensive costs and smaller sizes give us easy access to the 3D data at a higher frame rate and resolution. The easy access to real-time RGBD data is leading to a revolution in computer vision, robotics, and other related fields. Combining the strengths of optical cameras and range sensors, RGBD sensing makes visual perception more robust and efficient, leading to the emergence of systems that reliably recognize everyday objects and daily activities in complex scenes, as well as systems that build detailed 3D models of indoor spaces. The quality of the depth sensing, given the low-cost and real-time nature of the devices,

is compelling, especially when compared with the previous commercial range sensors. The only imperfection is the noise of the data. Depth measurements often fluctuate and depth maps contain "holes" where no estimation of depth are obtained in case of specific material, reflection, interference, or fast motion.

I propose to take advantage of this readily available RGBD data to improve the performance of existing computer vision algorithms. Especially, I propose to address the problem of human activity recognition with this added depth channel. I demonstrate that even under the current constraint of the quality of the depth images, the improvement of the results made by the 3D information is still quite encouraging. With the rapid development of new depth sensors, we believe that the future generations of the depth sensors will bring us better quality images and our 3D algorithms will benefit more applications.

1.1.2 Human Detection

Human detection is an important and basic task for many computer vision systems. It may offer a starting point for pose estimation, action recognition, and human-computer interaction tasks. It is a crucial component for autonomous systems such as intelligent cars and social robots. Detecting humans in images or videos is a challenging problem due to variation in poses, clothing, lighting conditions, and the complexity of the backgrounds. The problem becomes more difficult when there are several persons moving in the same area, or interacting with each other. In these cases, individual persons

may be partially or totally occluded by other persons. There has been a significant amount of research in the past decade on human detection, and various methods have been proposed [19, 20, 72]. Most of the research is based on images captured by visible-light cameras. We may divide the algorithms of finding persons from visible-light images into three categories: detection from tracking, sliding-window approach, and part-based approach. Tracking based algorithms assume that a person is moving in specific patterns, such as walking with a relatively smooth speed. The limitation is that static persons cannot be detected, which happens very often in indoor scenes. The sliding-window approach was first proposed to detect pedestrians [19]. It assumes a more restricted pose of the person and has difficulty generalizing to persons with various poses and rotations. The part-based approaches model the person as a collection of parts, it is more flexible with the pose of the person, and may handle partial occlusions. The parts are usually detected beforehand using texture characteristics. Although these methods give satisfactory detection results under certain scenarios, e.g. pedestrian on the road, RGB image based methods encounter difficulties in perceiving the shapes of the human subjects with articulated poses when the background is cluttered, or when the color of the person is hardly distinguishable from the background (which often happens in poor lighting areas).

Depth information is an important cue for humans to recognize objects, since the objects may consists of many color blocks and various textures but must occupy a continuous region in space. Depth image gives an alternative

to find human in the scene. In the past, there was research on detecting humans using range data, which was single or multiple 2D scan lines of laser measurements. The detection is usually achieved from tracking, the vertical moving blobs are usually considered to be a human. These type of algorithms may not work for indoor person detection when the persons may be static or occluded by objects. In this thesis, I proposed to detect humans from a single depth image. The person may be walking, sitting, dancing, or interacting with other objects or persons. Despite the advantages of depth sensors, it is still quite a challenging task since the "appearance" of a human in a depth image may change drastically as a function of body pose, distance to the sensor, self-occlusions, and occlusion by other objects. Unlike the previous works, I do not rely on tracking to find the person. Furthermore, since I aim to find persons under various poses, the traditional sliding-window approach would not be appropriate to give good performance. I present a novel model based method for human detection from depth images based on the 3D shapes. Our algorithm utilizes depth information only. It can also be combined with traditional gradient based approaches on RGB imagery to give more accurate detection when the visual input is reliable. The detection algorithm may serve as an initial step of the research on pose estimation, tracking, or activity recognition using depth information.

1.1.3 Human Activity Recognition

Recognizing human activity is one of the important areas of computer vision research today. The goal of human activity recognition is to automatically detect and analyze activities from sensors, e.g. a sequence of images, either taken by RGB cameras, range sensors, other sensing modalities, or a combination of a few sensing modalities. Its applications include surveillance systems, video analysis, robotics and a variety of systems that involve interactions between persons and electronic devices including computers. Its development began in the early 1980s. Past research has mainly focused on learning and recognizing actions from video sequences taken by a single visible light camera [1, 86]. The major issue with this type of data is that capturing articulated human motion from monocular video sensors results in a considerable loss of information, which limits the performance of video-based human action recognition. Despite the efforts in the past decades, recognizing human activities from videos is still a challenging task.

Depending on the situation, human activity may have different forms ranging from simple actions to complex activities. They can be conceptually categorized into 4 categories [1]: atomic actions, activities that contains a sequence of different actions, person-object interaction, and person-to-person interaction, ranging from two person interaction to group activities. Research on atomic action recognition from 3D began more than 20 years ago, while complex activities and interactions were studied more recently, especially after easy access of 3D data become available. In this thesis, I will cover atomic

action recognition, daily activity recognition, and person-to-person interaction from a first-person viewpoint.

We enumerate four major challenges to vision based human action recognition. The first is low level challenges. Occlusions, cluttered background, shadows, and varying illumination conditions can produce difficulty for motion segmentation and alter the way actions are perceived. This is one of the major types of difficulty of activity recognition from RGB videos. The introduction of 3D data largely alleviates the low-level difficulties by providing the structure information of the scene. The second challenge is view changes. The same actions can generate different "appearances" from different perspectives. Solving this issue with a traditional RGB camera is done by introducing multiple synchronized cameras, which is not an easy task for some applications. For recognition from range images, this problem is partially alleviated since the "appearance" from a slightly rotated view can be inferred from the depth data. The problem is not totally solved, though, because the range image only provides information on one side of the object in view, nothing is known about the other side. If skeletal joint information can be inferred accurately using a single depth camera, the recognition algorithm which builds upon the skeletal joint information can be view-invariant. The third challenge is scale variance, which can result from a subject appearing at different distances to the camera and subjects of different body size. In RGB videos, this can be solved using windows or filters at multiple scales, which largely increases the computational cost. In depth videos, this can easily be adjusted because the dimension

of the object can be estimated from the depth data. The fourth challenge is intra-class variability and inter-class similarity of actions. Individuals can perform an action in different directions with different characteristics of body part movements, and two actions may be only distinguished by very subtle spatio-temporal details. This remains a difficult problem for most algorithms using various types of data.

1.1.4 First-Person Activity Recognition

Action and activity recognition systems have attained crucial importance in recent years. Most of the works focus on recognizing activities that are not directly performed in relation to the observer: some of them recognize activities and actions that are independently executed by a single person, such as running, drinking, or jumping [11]. Other researchers analyze interactions between two persons [38] or groups of people [18, 43] from a third-person perspective. Works on recognizing first-person human interaction activities, i.e. activities performed by a person that are directly related to the presence or behavior of the explorer is very limited [70]. In particular, the first-person recognition task can be formulated as follows: given a moving subject that is actively exploring the scene, e.g. an observer equipped with a sensor, the interesting activities are those that directly involve the observer. Examples of such activities are punching, shaking hands, and throwing an object at the observer. Analyzing this class of activities enables to understand whether the persons surrounding the observer are friendly or hostile, and whether there will



Figure 1.1: Sample RGB image frames from our first-person dataset.

be a threat. This novel problem is useful in contexts such as video surveillance, where the security camera would need to understand if somebody is trying to damage it. Human-machine interaction, where the machine has to properly react to a person's behavior, is also a domain where classifying first-person activities is fundamental. Even recently developed wearable devices such as the Google Glass [80], may use these features.

In this thesis, I propose to study the problem of human interaction-level activities using RGBD data from a first-person view-point. Several types of features are investigated for this new task including both $2D$ and $3D$ features. To the best of my knowledge, the only previous work on human interaction recognition from a first-person perspective was proposed in [70]; nevertheless, they used simple RGB features to classify activities. I experimentally demonstrate that adjoining depth, skeleton, and $3D$ information significantly

increases the activity classification accuracy.

First-person activity videos are notably different from classic action recognition videos, as the videos present a strong ego-motion component. Sample images recorded from our datasets are shown in Figure 1.1. In this thesis, I proposed an attention mask to help focusing the descriptors and differentiating ego-motion regions from independent-motion regions. Unlike the majority of the works in the literature, which suppresses ego-motion data, I exploit both pieces of information. I show that this technique improves our results significantly.

1.1.5 Overview of My Work

Given a depth image, the first problem to consider is to detect the persons in the scene. Localizing humans from RGB images is limited by the lighting conditions and background clutter. Depth image gives alternative ways to find the human in the scene. In the past, there was research on detecting humans using range data, which was single or multiple 2D scan lines of range data. The detection is usually achieved from tracking, which does not work for indoor person detection where the persons may be static and occluded by objects. In this thesis, I presented an algorithm to detect humans of various appearances and poses from a single depth image. Since the depth images lack of texture, I propose a model based approach to detect the persons using the structural information embedded in the depth images. The most stable feature of a person in the depth image is the head and shoulder part, the structure

of which does not change too much with different poses and body shapes. Also, the shape of the body and shoulder stays similar from frontal, back, and side views. The major detection process contains two stages. Firstly, a 2D chamfer model is matched across the whole image and gives the regions that possibly contain a human head. Secondly, an occlusion mask is extracted for each region. A 3D head model is constructed at the correct scale and fit onto the regions with the occlusion mask, resulting in the final estimation. Both matching stages are guided by the depth value to adjust the model to the correct scale. A region growing algorithm is proposed to find the entire human body, and the body contour is extracted. All planar surfaces in the depth image are extracted to avoid the human region growing onto the planar regions in the scene such as floors and tables. Furthermore, a simple tracking algorithm was developed based on the detection result. The algorithm was tested on 2 datasets captured by a Kinect in two different indoor settings and presented superior results than state-of-the-art works on RGB images or depth images [19, 34, 77].

I further researched on recognizing human actions and activities. I proposed two features for recognizing human activities. The first one is drawn from the skeletal joint locations of the person, which is a high level abstraction of the human posture and can be directly extracted from depth images [75]. I designed a compact representation of the human posture called histograms of 3D joint locations (HOJ3D) from the skeleton. View-invariant property is achieved by building the reference coordinates in the 3D skeleton space.

A spherical coordinate is constructed according to the joints on the torso of the person. Joints on the limbs are casted into the 3D spherical coordinates and concatenated into a histogram, which constitutes the HOJ3D feature. The HOJ3D computed from the depth sequences are reprojected using LDA and then clustered into posture visual words, which represent the prototypical poses of actions. The temporal evolutions of those visual words are modeled by discrete hidden Markov models (HMMs). This feature may benefit many applications to get a fast estimation of the posture and action of the humans in the scene. However, this feature is limited by the dependence of the skeletal joints estimation result. The skeletal estimation algorithm is not reliable or may fail under some real-world scenarios, e.g. when the human body is partly in view, when the person touches the background, when the person is not in an upright position, or when the sensor is mounted on a higher location and angled downwards.

To be able to recognize the activity when the skeleton information is not available/reliable, a more general feature is desired. Inspired by the success of the spatio-temporal interest point method on RGB videos, I develop a spatio-temporal feature to describe the local 3D patches in the depth video. First, interest points are extracted using a filtering algorithm that effectively suppresses the noisy measurements and finds the salient motion in the depth video. Then, a novel depth cuboid similarity feature (DCSF) was proposed to describe the local 3D depth cuboid around the DSTIPs with an adaptable supporting size and handles the noise in the depth video. The DCSF feature

was designed based on self-similarity notion, which has been proved to be more robust on depth data than gradient based feature that has been widely used on RGB data [34]. The DCSF features from all the DSTIPs form the description of a video, and the activity contents of the video can be classified using a bag-of-words approach. We tested this feature using our own dataset and the MSRAction3D and MSRDailyActivity3D public datasets. Experimental evaluations showed that the proposed approach outperforms state-of-the-art features and algorithms on depth videos, and this framework is more widely applicable than existing methods.

Finally, all the features herein developed are employed and combined to solve a novel problem: first-person human activity recognition using RGBD data. This task is very novel and extremely challenging due to the large amount of motions of the camera. I extracted 3D optical flow features as the motion descriptors, 3D skeletal joints features as posture descriptors, and HOG/HOF from RGB channels and DCSF from depth channel as the local appearance descriptors to describe the first-person videos. As mentioned, the skeleton is not available for some sequences or some part of the sequence, in those cases, a symbol for the missing skeleton is inserted. To address the ego-motion of the camera, an attention mask was proposed to guide the recognition procedure and separate features on the ego-motion region and independent-motion region. Studies conducted on primates suggest that ego-motion and independent-motion are perceived by two different areas of the brain [54, 84]. Motivated by these findings, I propose a new version of state-of-the-art descrip-

tors, explicitly differentiating ego-motion regions from independent-motion regions. Unlike the majority of the works in the literature, which suppresses ego-motion data, I exploit both pieces of information. I show that this technique improves our results significantly.

1.2 Main Contributions

I developed algorithms constituting a pipeline from human detection to activity recognition using 3D data. The algorithms developed here are in line with the fast development of RGBD hardware and the rapid growing number of research conducted using RGBD perception in the past several years. The existing resources and related works for each category were limited at the time of developing the algorithms. The proposed work made contributions to the fast growing literature of the related areas. I summarized the contribution into the following 5 aspects.

Human detection algorithm using a single depth image I proposed an algorithm on human detection from a single depth image that is able to detect moving or static persons in various poses and appearances. The model-based approach enables the system to find the persons in the scene, which provides initialization of a variety of tasks such as pose estimation, gesture recognition, activity recognition, human-computer interaction, and so on.

Real-time HOJ3D feature for action recognition I proposed a view-invariant feature for action recognition using skeletal joints information. This algorithm recognizes the action of persons in real-time and independent of the

viewing angle, which is desirable for many applications.

Depth cuboid similarity feature for activity recognition I proposed a novel spatio-temporal feature for depth video that specially handles the noise of the depth video and gives robust and discerning descriptions of the human motion in depth video. This feature offers the possibility to understand the contents of the depth video or the activity of persons without the dependence on the skeleton, which is unreliable or not even available in many real-world applications.

First-person interaction activity recognition with RGBD data I proposed to analyze the activities or interactions of persons from a first-person view-point using RGBD perception. I demonstrated that depth information is helpful at this task where the camera presents significant ego-motion, and combining features extracted from RGB and depth channels gives the optimum result in this challenging task.

RGBD datasets on human detection and activity recognition I made publicly available one depth dataset on human detection, one RGBD dataset on action recognition, and two RGBD datasets on first-person activity recognition using two different first-view settings.

The algorithms proposed here have been cited more than 200 times by Mar. 2014.

1.3 Road Map

In the following Chapter, I describe related work to my thesis. In Chapter 3, I present my human detection algorithm from a depth image. In Chapter 4, I describe the skeletal joint feature for action recognition. In Chapter 5, I give details of extracting spatio-temporal interest points from depth video and the construction of depth cuboid similarity features to describe depth cuboids. In Chapter 6, I describe my work on first-person activity recognition from RGBD data. Finally, in Chapter 7, I conclude the dissertation.

Chapter 2

Related Work

2.1 Human Detection

Human detection has been intensively studied in the past decade. Most of the works focus on pedestrian detection in outdoor scenes for vehicular applications. Both visible light video cameras and range sensors have been explored for this task.

Approaches on visible light images or videos include the following 3 categories. The first category finds humans from tracking. Foreground objects are usually distinguished through a background subtraction process, and the foreground blobs are tracked and verified based on the motion or geometric shape of the blob [30, 35]. Depending on the approach, static background images may be needed to initialize the background model, and a shape model of the pedestrian may be needed for recognition [30]. The limitation of this type of approach is that the humans have to be moving in the desired pattern to be recognized. The second category is sliding-window based methods, which are also popular among researchers. Earlier ones, such as [63], used 2D wavelets (vertical, horizontal, diagonal) as the detection window and Dalal et al. [19] developed the widely known sliding-window HOG feature for pedestrian de-

tection. Simple as it is, the sliding-window approach has the tacit assumption that the human body has a general restricted geometric shape, e.g. a vertical walking person. Special effort has to be made to handle occlusions. The third category is part-based methods. Part based algorithms are more flexible at modeling shape articulations. The individual human is modeled as an assembly of body parts. It handles partial-occlusions efficiently. Different features are selected to detect body parts. Parts are then combined to form a joint likelihood model, and the human detection problem may be formulated as a MAP estimation problem [56, 95]. Alternatively, humans may be detected in a hierarchical way by combining a global template and local parts using a Bayesian approach [50].

Despite the detailed approaches, human detection algorithms from visible light images or videos suffers from difficulties caused by cluttered backgrounds or lighting changes. Due to the information loss from 3D to 2D, it is very hard to perceive the contour of the human when the color of the human is not easily distinguished from the background objects. Lighting may change the appearance of the human body or parts drastically in the image, which may cause difficulty for the gradient features or part detectors.

There are also a number of works on human detection from range data. Early works looked for a moving local minimal in the scan for person detection [27]. Due to the natural performance limit for people detection using a single 2D range sensor, multiple co-planar 2D scanners might be used. Fod et al. [27] used multiple planar laser range finders to build a background/fore-

ground model. Range measurements are grouped into entities such as blobs and objects, and a Kalman filter was employed to estimate trajectories for these objects. With the development of sensors, 3D data was later explored for human detection tasks. Bajracharya et al. [3] detect upright human adults in point clouds from stereo vision by processing vertical objects and considering a set of geometrical and statistical features of the cloud based on a fixed pedestrian model. Navarro-Serment et al. [59] employs 3D LADAR measurements to find salient vertical objects above ground. Motion feature is extracted from the tracked objects to compute the potential of the object being human and each object was then classified using a pattern recognition technique based on geometric features. The above human detection algorithms developed on range data all depend on motion, which may not work when the person is not moving.

There are also researchers who equipped the system with multiple sensors to boost detection performance. Several researchers use depth as a cue to segment foreground blobs and then use visual algorithms to detect humans [71, 78, 99]. Cui et al. [17] employ laser scanners to provide feet trajectory tracking and combine it with visual body region tracking techniques in a Bayesian formulation. Similarly, Bellotto et al. [7] combine laser-based leg detection and visual face detection into a Kalman filter to detect and track the persons from a mobile robot. Choi et al. [16] fuse image-based pedestrian and upper body detectors, a face detector, a skin detector, as well as a depth-based shape detector and motion detector into a sampling framework to construct a

tracking-by-detection formulation. In another way, Rivera-Bautista et al. [69] use a face detector and a skin color detector to find the person and then employ a 3D region growing method to find the full body. Furthermore, Spinello et al. [77] design a counter-part of the HOG feature in depth data called the Histogram of Oriented Depths (HOD) which is combined with HOG to detect pedestrians. All the above methods depend on visual channels for detection, depth information is mostly used as an auxiliary for the visual image detectors. Work on human detection from depth channels only is very limited [34].

Human detection algorithms with depth inputs only are desirable. There may be cases when visual inputs are not available, e.g. when the environment is very dark (surveillance at night). Also, compared to the multi-modal method, depth-only algorithms save the budget of one sensor as well as the computational cost for the additional channels.

In this thesis, I consider human detection in indoor settings. Indoor human detection is more challenging because of the possibilities of various postures. Simplistic assumptions in outdoor pedestrian detection may no longer be valid in indoor settings where people may stand, sit, lean on a wall, interact with objects, and so on. The persons may also get truncated by the image boundary or occluded by furniture or other persons.

2.2 Activity Recognition

Activity recognition has a long history; past research has mainly focused on learning and recognizing actions from video sequences taken by a

single visible light camera. The literature has been surveyed in many publications [1, 86]. Here I will mainly focus on introducing the related works on human activity recognition from depth or RGBD images [2]. Based on the features used, they may be divided into five categories: features from 3D silhouettes, features from skeletal joint or body part locations, local spatio-temporal features, local occupancy patterns, and 3D scene flow features.

2.2.1 Recognition From 3D Silhouettes

Among the early attempts on action recognition from intensity images, researchers have extracted 2D silhouettes as a simple representation of human body shape from the intensity or RGB images and model the evolution of silhouettes in the temporal domain to recognize actions. It was shown that the silhouettes, or, extremities of the silhouettes, carry a great deal of shape information of the body. By tracking the person's silhouette over time, Davis et al. [21] generated a Motion History Image (MHI) which is a scalar-valued image where intensity is a function of recency of motion. Fujiyoshi et al. [28] extracted a "star" skeleton from silhouettes for motion analysis. Yu et al. [103] extracted extremities from 2D silhouettes as semantic posture representation in their application for the detection of fence climbing. However, the silhouettes extracted from intensity images are view-dependent, and only suitable for describing actions parallel to the camera. Also, extracting the correct silhouettes of the actor can be difficult when there is background clutter or bad lighting conditions.

In a depth image, the silhouette of a person can usually be extracted more easily and accurately. In addition, the depth image provides the body shape information not only along the silhouettes, but also the whole side facing the camera. Thus, more information can be acquired from depth images. Many algorithms have been proposed to recognize actions using representations built from 3D silhouettes. Li et al. [48] sample a bag of 3D points on the contours of the planar projections of the 3D depth map to characterize a set of salient postures that correspond to the nodes in the action graph. The number of points can be controlled by the number of projection planes used. Yang et al. [101] also project depth maps onto three orthogonal planes. They propose Depth Motion Maps (DMM) which stack the motion energy through the entire video sequences on each plane. HOG is employed to describe the DMM. Ni et al. [60] propose a Three-Dimensional Motion History Image (3D-MHI) which equip the original MHI with two additional channels, i.e. two depth change induced motion history images (DMHIs): forward-DMHI and backward-DMHI which encode forward and backward motion history. Jalal et al. [36] use Radon transform (\mathcal{R} transform) to compute a 2D projection of depth silhouettes along specified view directions, and employ \mathcal{R} transform to transform the 2D Radon projection into a 1D profile for every frame. Fanello et al. [24] propose a Global Histogram of Oriented Gradient (GHOG) by extending the classic HOG [20] which was designed for pedestrian detection from RGB images. The GHOG describes the appearance of the whole silhouettes without splitting the image into cells. The gradient of the depth stream shows the highest response on the

contours of the person thus indicating the posture of the person. Wu et al. [96] propose extended-MHIs by fusing MHI with gait energy information (GEI) and inversed recording (INV) at an early stage. GEI compensates for non-moving regions and multiple-motion-instance regions. INV provides complementary information by assigning a larger value at initial motion frames instead of the last motion frames. The extended-MHI was proved to outperform the original MHI on an action recognition scenario. Kurakin et al. [42] divide the depth image into sectors and compute the average distance from the hand silhouettes in each sector to the center of the normalized hand mesh as a feature vector to recognize hand gestures.

Current algorithms using 3D silhouettes are suitable for single person action recognition and perform best on simple atomic actions. There is difficulty in recognizing complex activities due to the limitation of the representation. Occlusion and noise can mar the silhouettes dramatically, and the extraction of accurate silhouettes may be difficult when the person interacts with background objects (e.g. sitting on sofa). Furthermore, the depth map only gives the 3D silhouettes of the person facing the camera. Thus, the 3D silhouettes based algorithm are usually view-dependent, even though they are not limited to only modeling parallel motions as in intensity images.

2.2.2 Recognition From Skeletal Joints or Body Parts Tracking

The human body is an articulated system of rigid segments connected by joints, and human action is considered a continuous evolution of the spatial

configuration of these segments. Back in 1975, Johansson's experiment showed that humans can recognize activity with only seeing the light spots attached to a person's major joints [37]. In computer vision, there is plenty of research on extracting the joints or detecting body parts and tracking them in the temporal domain for activity recognition. In intensity images, researchers tried to extract "skeletons" from silhouettes [28], or label main body parts [9] such as arms, legs, torso, and head for activity recognition. Researchers also tried to extract joints or body parts from stereo images or to directly get them from motion capture systems.

In 2011, Shotton et al. [75] propose to extract 3D body joint locations from a depth image using an object recognition scheme. The human body is labeled as body parts based on the per-pixel classification results. The parts include LU/ RU/ LW/ RW head, neck, L/R shoulder, LU/ RU/ LW/ RW arm, L/ R elbow, L/ R wrist, L/ R hand, LU/ RU/ LW/ RW torso, LU/ RU/ LW/ RW leg, L/ R knee, L/ R ankle and L/ R foot (Left, Right, Upper, Lower). This offers us easy access to the skeletal joint locations of the persons with overall better accuracy, and this excited considerable interest in the computer vision society. Many algorithms have been proposed after that recognizing activities using skeletal joint information. The most straight forward feature is the pairwise joint location difference feature, which is a compact representation of the structure of the skeleton posture of the current frame. By computing the difference of the joint positions from the current frame and previous frame, one can get the joint motion between the two frames. Especially, supposing

the first frame is a neutral pose, taking the joint position difference between the current frame and first frame can generate an offset feature. Masood et al. [53], Zhang et al. [106] and Yang et al. [100] concatenates these features and test its effectiveness at recognizing activities.

From the skeletal joint locations, joint orientation can be computed, which is invariant to human body size. Sempena et al. [74] build a feature vector from joint orientation along time series and apply dynamic time warping onto the feature vector for action recognition. Bloom et al. [10] concatenates 5 types of features: pairwise joint position difference, joint velocity, velocity magnitude, joint angle velocity w.r.t. the x-y plane and x-z plane, and 3D joint angle between three distinct joints. In total, 170 features were computed to recognize gaming actions.

Researchers also tried to group the joints and construct planes from joints and measure joint-to-plane distance and motion as features. Yun et al. [104] construct a feature that captures the geometric relationship between a joint and a plane spanned by 3 joints. This feature is intended to describe information such as how far the right foot lies in front of the plane spanned by the left knee, hip, and torso. Sung et al. [81] compute each joint's rotation matrix with respect to the person's torso and hand position as features and use a maximum-entropy Markov model (MEMM) to learn the actions.

The skeleton posture feature I developed came around the same time as these related works, and it offers an alternative method for action recognition with a real-time and view-invariance features.

2.2.3 Recognition Using Local Spatio-Temporal Features

Local spatio-temporal features have been a popular description for action recognition in intensity videos. The video is regarded as a 3D volume along space (x, y) and temporal t axis. Generally, local spatio-temporal interest points (STIPs) are first detected, then descriptors are built around the STIPs on the volume. Classification can be made from the descriptors using, e.g., bag-of-words approach. Many different STIP detectors [22, 44, 61, 94] and descriptors [22, 41, 45, 73, 94] have been proposed in the literature during the past decade. The local spatio-temporal features have demonstrated successful at recognizing a number of action classes with varying difficulties [89].

Encouraged by the success in intensity video, researchers also tried the spatio-temporal features in depth videos. Ni et al. [60] use depth information to partition the space into layers, extract STIPs from RGB channels of each layer using a Harris3D detector [44], and use HOG/HOF [45] to describe the neighborhood of STIPs in the RGB channel. In this approach, depth was only served as a auxiliary for the extraction of STIPs from RGB videos, the detector and descriptor was applied on the RGB channels. Zhang et al. [107] extract a 4D cuboid from RGBD video by calculating a response function from both depth and RGB channels and use the intensity and depth gradients along x, y, t directions as the local feature. Cheng et al. [15] extract STIPs from depth video using a Harris3D detector. They propose a Comparative Coding Descriptor (CCD) feature to describe the $3 \times 3 \times 3$ depth cuboid by comparing the depth value of the center point with 26 nearby points. Sim-

ilarly, Zhao et al. [108] use a Harris3D detector [44] to extract STIPs from RGB or depth channels and combine HOGHOF [45] and the proposed local depth pattern(LDP) feature for representation. The LDP feature is defined by the difference of average depth values between nearby cells of the 3D cuboid. In this paper, IPs extracted from RGB channels perform better than IPs extracted from the depth channel using Harris3D for about 2.5% on the RGBD-HuDaAct dataset [60]. It is reasonable since the Harris3D was designed for intensity videos with rich textures, and the depth videos are usually noisy and have many missing values. To deal with this, I propose in this thesis a filtering scheme to find the STIPs from depth videos with noise suppression functions. Also, I propose a new type of feature which published around the same time as these related works but conveys more information of the local 3D patch than the CCD [15] and LDP [108].

Local spatio-temporal features capture shape and motion characteristics in video and provide independent representation of events. It is invariant to spatio-temporal shifts, scales, and background clutter. Also, it naturally deals with partial occlusions, multiple motions, person-to-person interaction and person-object interaction. Since such features are directly extracted from the video without the need for motion segmentation and tracking, these algorithms are more robust and have a wider range of applications.

2.2.4 Recognition Using Local 3D Occupancy Features

Instead of representing the depth video as 3D spatio-temporal volume, the points may be projected to the 4D (x, y, z, t) space. In this 4D space, some location will be occupied by the data points from the video, i.e. the points that the sensor captured from the real world, those locations will have a value of 1, others 0. In general, the local occupancy pattern is quite sparse, that is, the majority of its elements are zero. The local occupancy pattern has been proposed individually by several researchers for activity recognition. In fact, the local occupancy feature can be defined in the (x, y, z) space or (x, y, z, t) , the former one describes the local depth appearance at a certain time instant while the latter describes the local atomic events within a certain time range. Wang et al. [90] design a 3D Local Occupancy Patterns (LOP) feature to describe the local "depth appearance" at joint locations to capture the information for person-object interactions. The intuition is, when the person fetches a cup, the space around the hand is "occupied" by the cup. The x, y, z space around the joint is partitioned into a $N_x \times N_y \times N_z$ spatial grid, the number of points that fall into each bin are counted and normalized to obtain the occupancy feature of that bin. This work is an example of combining skeleton joints features and local occupancy features to recognize activities and also to model person-object interactions. Wang et al. [91] defined the random occupancy patterns in the (x, y, z, t) domains. Similarly, the occupancy feature is the sum of the pixels in a sub-volume of the 4D space normalized by a sigmoid function. A weighted sampling approach was proposed to sample sub-volumes

from the 4D space, and occupancy patterns were extracted from those locations to give an overall description of the depth video. Instead of random sampling, Vieira et al. [88] divided the whole space-time volume into 4D grids, and extracted occupancy patterns from every partition. Interestingly, a saturation scheme was proposed to enhance the role of the sparse cells, which typically lie on the silhouettes or moving parts of the body. To deal with the sparsity of the feature, a modified-PCA called Orthogonal Class Learning (OCL) is employed to cut the length of the feature to 1/10 of its original.

The local occupancy feature defined in the (x, y, z, t) space is similar to local spatio-temporal features in that they both describe local "appearance" in the space-time domains. Local spatio-temporal features treat the z dimension as "pixel values" in the (x, y, t) volume while local occupancy patterns project the data onto a (x, y, z, t) 4D space containing 0-1 values. They may both be extracted from selected locations or random sampling. However, the local occupancy features can be very sparse while the spatio-temporal feature is not. Furthermore, spatio-temporal features contain information on the background since the cuboid is extracted from the (x, y, t) space, while local occupancy features only contain information around a specific point at a (x, y, z, t) space. This characteristic is not positive or negative as the background is helpful in certain scenarios while disturbing in some other cases.

2.2.5 Recognition From 3D Optical Flow

Optical flow is the distribution of apparent velocities of movement of brightness patterns in an image, which arises both from the relative objects' and the viewer's motion [29]. It is widely used in intensity images for motion detection, object segmentation and stereo disparity measurement [5]. Also, it is a popular feature in activity recognition from videos [14, 102]. When multiple cameras are available, the integration over different viewpoints allows a 3D motion field, the scene flow [87]. However, intensity variations alone are not sufficient to estimate motion and additional constraints such as smoothness must be introduced in most scenarios. Works on estimating 3D scene flow from stereoscopic include [12, 93] and [33]. These algorithms usually have high computational cost due to the fact that they estimate both the 3D motion field and disparity changes at the same time. Depth cameras advantageously provide useful geometric information from which additional consistent 3D smoothness constraints can be derived. With a stream of depth and color images coming from calibrated and synchronized cameras, we have a simpler way of getting optical flow in (x, y, z) space. Among the more straight forward and faster methods, Swadzba et al. [83] and Fanello et al. [24] compute 3D scene flow by transforming the 2D optical flow vectors to 3D using the 3D correspondence information of each point, i.e., each 2D pixel x, y is projected into 3D using the depth value z and the focal length f : $X = (x - x_0)Z/f, Y = (y - y_0)Z/f$. (x_0, y_0) is the principal point of the sensor. They compute the 2D optical flow using traditional methods such as [32] or [52]. The 3D scene flow may be

obtained by differencing the two corresponding 3D vectors in two successive frames F_{t-1} and F_t using equation $\mathbf{D} = (X_t - X_{t-1}, Y_t - Y_{t-1}, Z_t - Z_{t-1})$ [24]. The 3D scene flow estimated using the above method has been proven effective at recognizing arm gestures [31] and upper/full body gestures (ChaLearn dataset) [24]. However, this method is not the best estimation of the 3D scene flow, since only the 2D information is considered when finding the correspondences between frames.

Recently, Letouzey et al. [47] cast the problem of estimating 3D scene flow from a calibrated depth and RGB image sequence as an optimization problem with photometric consistency constraints and motion field regularization. Ballin et al. [4] compute the 3D scene flow from point cloud data using 1-nearest neighbor search driven by both the 3D geometric coordinates and the RGB color information. The 3D scene flow is only computed for relevant portions of the 3D scene. They represent each tracked person by a cluster which is defined as a 4D point cloud. The 3D scene flow vector is then summarized within a 3D grid surrounding each cluster, and 3D average velocity vector is computed for each 3D cube and all these vectors are concatenated into a column vector. This feature is tested on a human action recognition task and shows reasonable performance on a new dataset containing six simple human actions.

Compared to the success of traditional 2D optical flow, the research on scene flow is still in its preliminary stage [57]. Currently, 3D scene flow is often computed for all the 3D points for the subject or scene, resulting

in a large computational cost. Computing the 3D scene flow with real-time performance is a challenging task. We may imagine that after the emergence of more effective ways to compute 3D scene flow, it can be a more popular type of feature for human action recognition and benefit more applications.

2.3 First-Person Activity Recognition

Over the past few years, low-cost high-end wearable cameras have been made available to the public. This resulted in an explosion of first-person viewpoint videos that make the analysis of first person activity an increasingly popular topic within the computer vision community. The majority research in the field of first-person video analysis regards daily household activities from ego-centric videos [26, 39, 55, 65, 79]. These works are usually object-driven and focus on analyzing the relationship between the object and the body parts that manipulate the objects [6]. In a different category, Kitani et al. [40] learn the ego sport activities from first-person videos collected by sports enthusiasts for indexing and retrieval. Lee et al. [46] develop techniques for video summarization by discovering important people and objects in the egocentric videos. All the above mentioned works try to analyze the ego-activity of the person who wears the camera.

In this thesis, I focus on recognizing the activities that a person performs with respect to the explorer. Our task is different from the previous ego-centric activity analysis in that we are trying to answer the question: what are they doing to me, while the previous category of works are trying to

answer the question: what am I doing. Michael et al proposed to study the first-person interaction activity recognition using a webcam last year [70]. In this thesis, I studied this problem using RGBD videos to gain more information which resembles more the human binocular vision system. Additionally, in [70] the motion and appearance descriptors are extracted from the whole scene, therefore the ego-motion and the independent motion components are mixed together. On the contrary, I distinguished between regions that move due to ego-motion, and regions that move independent of camera. In the literature, there are several works that segment the person/body parts from the background [64] to localize the independent motion for activity recognition; Most of these techniques aim to suppress the information from the surrounding regions. In contrast, I demonstrate that, for the first-person task, descriptors extracted from both the areas contribute in a different manner to the recognition procedure; using both of them indeed, is crucial to improving the classification rate.

Chapter 3

Human Detection Using a Single Depth Image

In this Chapter, I will describe the algorithm for detecting humans from a single depth image. A 2D chamfer model is first matched across the whole image and gives the regions that possibly contain a human head. An occlusion mask is extracted for each region. Then, a 3D head model is built at the correct scale and fit onto the regions with the occlusion mask, resulting in the final estimation. Both matching stages are guided by the depth value to adjust to the correct scale of the object in the scene. A region growing algorithm is applied to find the entire human body, and the body contour is extracted. All planar surfaces in the depth image are extracted to avoid the human region growing onto the planar regions in the scene such as floors and tables. Further, a simple tracking algorithm is proposed based on the detection result. The algorithm is tested on 2 datasets captured by a Kinect in two indoor settings and presents superior results than state-of-the-art works.

3.1 Algorithm

3.1.1 Preprocessing

The resolution of the original depth image is 640×480 . To make the detection faster, the images are down sampled by a factor of 2. Simple preprocessing steps are performed to make the depth image less noisy. First, a nearest neighbor interpolation is employed to fill the holes in the depth image. Then, a median filter with a 4×4 window is applied onto the image to smooth the depth values.

3.1.2 Regression on the Diameter of the Head

One of the advantages of the depth data is that true dimensions of the objects may be inferred from the depth value. The variant scales resulting from subjects appearing at different distances to the camera are usually addressed using windows or filters at multiple scales, which largely increases the computational cost. Here, an experiment is conducted to find the relation between the depth value and the scale of the head in the depth image. Head diameters (in pixel) and depth values are manually annotated in a set of images. This information is used to compute a scale-depth regression shown in Fig. 3.1. The regression curve can be expressed by:

$$H(d) = \frac{f \cdot H_r}{d} = \frac{1.3 \times 10^5}{d}. \quad (3.1)$$

Here, d is the depth value of the center of the head in millimeter, H is the diameter of the head in the depth image, measured by pixels, f is the focal

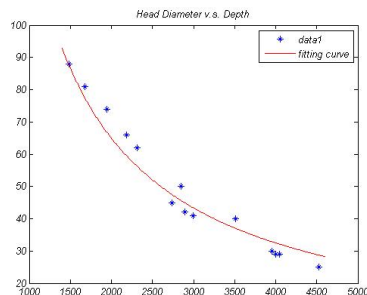


Figure 3.1: Regression curve of the relation between head diameter (in pixels) and depth value (in millimeters)

length of the camera, H_r is the real size of a standard human head in the scene. For any location of the scene in the depth image, the approximate dimension of a standard head can be computed from this equation if a head appears at that location. This reduces the computational cost of the matching process, and might also increase the detection accuracy by neglecting objects of the wrong scale.

3.1.3 2D Template Matching

In this section, we describe our detection process. The first stage is a rough scanning step where a 2D head template is searched throughout the image to locate possible regions that may contain a head. In this stage, only the edge information in the depth image is used, which corresponds to the spatial discontinuities of the scene. A 2D chamfer distance matching algorithm is employed for quick processing.

3.1.3.1 2D Chamfer Distance Matching

A Canny edge detector is applied onto the depth image to find all edges. The parameter for the Canny edge detector is chosen so that the contour of the human is mostly complete and continuous, but fine details are neglected. Then small or short edges are removed to reduce the disturbance from small or irregular shaped objects.

The binary head template is manually generated (shown in Fig. 3.2(d)). To increase the efficiency, a distance transform is computed before the matching process. Distance Transform is a function $D(\cdot)$ that for each image pixel p assigns a non-negative number $D(p)$ corresponding to distance from p to the nearest feature point in the image I . This results in a distance map where pixels contain the distances to the closest edge pixels. The matching process consists of translating and positioning the template at various locations of the distance map. We summarize it as a minimization process:

Given:

- Binary edge image B , where $B(i, j) = 1$ at edges and $B(i, j) = 0$ otherwise
- Binary head edge template, T , of shape we want to match. $T(i, j) = 1$ at edges and $T(i, j) = 0$ otherwise
- Let D_B be the distance map of edge image B , $[X_w, X_h]$ be the size of the size of the depth image, $[a, b]$ be the size of the template.

Goal: Find placement of T in D that minimizes the sum, M , of the distance

transform multiplied by the pixel values in T , i.e.

$$\text{Minimize : } M(i, j) = \frac{1}{|D_T|} \sum D_B \otimes T(i, j) \quad (3.2)$$

$$\text{s.t. } a/2 < i < X_w - a/2 \quad (3.3)$$

$$b/2 < j < X_h - b/2 \quad (3.4)$$

Here, \otimes represent element-wise product (Hadamard product), $|D_T|$ is a normalization term.

$$|D_T| = \sum_{s,t} |D(i + s, j + t)| \quad (3.5)$$

$$-a/2 < s < a/2, -b/2 < t < b/2 \quad (3.6)$$

If T is an exact match to B at location (i, j) then $M(i, j) = 0$. If the edges in B are slightly displaced from their ideal locations in T , we will get a small non-zero number depending on the displacement. Without the normalization term, if the local patch of the image contains dense edges, it will get a close match (small M value) even if the shape does not resemble the template.

As we do not assume one person in each image, we compute $M(i, j)$ for all the locations in the image. The smaller the M values are, the better the match between image and template at this location. If the distance value lies below a threshold τ , the target object is considered detected at this place, which means that a possible head is found. In this stage, a high threshold is set to guarantee a low false negative rate. The result of chamfer distance matching is shown in Fig. 3.2

The head template here is able to find the head of a person in various poses and views only if the person is in an approximate upright position. If the person is lying down or upside down, the algorithm needs to be adjusted by rotating the template and running the same detection process.

3.1.3.2 Depth Guided Pyramid Matching

An important and novel part in this matching process is that we do not run this 2D template matching at multiple scales across the whole image like the normal practice. With equation 3.1, the correct scale of the head that we are looking for at a particular location of the image can be computed. Then we just match at that location using the correct scale, we call it depth guided matching.

We define the level of the pyramid to be L , in each level, the image shrink by a factor of γ ($\gamma = 4/5$ in our experiments). Let the diameter of the head template to be H_0 . First, we find the range of the depth that a person may appear $[d_{min}, d_{max}]$, this can be set manually to the depth range that we are interested in, or simply by finding the maximum and minimum valid depth value in the image. By setting $H_0 = H(d_{min})$, we only need to down sample the image at every level when generating the image pyramid, without the need to up-sample the image. Note the pyramid matching can be done in two equivalent manners: either generate an image pyramid or a template pyramid. Since the template is a binary edge image, the down/up sampled template edge image does not look good when γ is not a integer, so we chose

to generate an image pyramid.

The number of levels of the pyramid can be computed:

$$L = \log_{\gamma} \frac{d_{max}}{d_{min}} \quad (3.7)$$

A table is generated that records the correct template scale $C(i, j) = c, c \in 1, 2, \dots, L$ for all the locations in the image $I(i, j)$. First, we compute the equivalent template scale for each pyramid levels.

$$H_i = \frac{H_{i-1}}{\gamma}, \quad i = 2, \dots, L \quad (3.8)$$

Then the depth value corresponding to the head template scales can be computed from equation (3.1). With this equation, the depth value $d(i)$ corresponded to the template scales $H(i)$ is computed simply by $d(i) = 1.3 \times 10^5 / (H(i)), i = 1, \dots, L$. Then, each image location $I(i, j)$ is assigned to one of the pyramid levels by matching the pixel value to the nearest $d(i)$. Each pixel location is only matched to the template at the recorded pyramid level. This on average reduces the computational cost to $1/L$ compared to traditional pyramid matching. The 2D matching step is a rough scanning process that gives a rough detection result with a very low false negative rate and high false positive rate. In the next stage, each location is further examined to rule out false positives.

3.1.4 3D Model Fitting

In this section we utilize the relational depth information in the depth image to verify the head. We generate a 3D head model to fit onto the image.



Figure 3.2: Intermediate results of 2D Chamfer distance matching. (a) Preprocessed depth image. (b) Binary edge image calculated by Canny edge detector. (c) Distance map generated from binary edge image (d). The binary head template (e). 2D template matching result (yellow dots indicate the detected locations).

The complexity of 3D model fitting is much higher than 2D template fitting. To simplify the process, we expect the model to generalize the characteristics of the head from any view: frontal, back, side, and also higher and lower viewing angles when the sensor is placed higher or lower or when the person is higher or lower. To meet these constraints, we chose a hemisphere as the 3D head model.

3.1.4.1 Head Radius

Instead of taking the result from equation 3.1 as the diameter of the 3D model, we propose to look for the true diameter of the head from the image. This makes the algorithm invariant to scale differences of the head of different persons. Interestingly, the true radius of the head has already been computed in Section 3.1.3.1. Recall that a pixel in the distance map is the distance from the pixel to the closest data pixel in the edge image. Supposing the head is a circular shape, the pixel value at the center of the head (i_0, j_0) on the distance map is just an approximation of the radius of the head, so we

take $R_t = \max D_B(i, j), i_0 - 1 \leq i \leq i_0 + 1, j_0 - 1 \leq j_0 + 1$. The variation of the humans head size is limited, the head grows about 1.67 times in size from infancy to adulthood [92]. With this statistics R_t , some false positives can be removed. We remove the detection if $R_t > 1.4 * H$ or $R_t < 0.5H$, H is the average adult head diameter from section 3.1.2.

A 3D hemisphere model is generated from radius R_t :

$$z = \alpha \frac{\sqrt{R_t^2 - x^2 - y^2}}{R_t} \quad (3.9)$$

$$x \in [-R_t, R_t] \quad (3.10)$$

$$y \in [-R_t, R_t] \quad (3.11)$$

Here, α is a scaler to adjust the depth value according to the standard adult head size, which is 9 inches in diameter.

3.1.4.2 Occlusion Mask

Occlusion relations can be inferred from the depth value. Here, an occlusion mask is generated for every region prior to the 3D model fitting. This will reduce the influence of the objects before and behind the head, and render the algorithm certain robustness against occlusion. Suppose the region center is (i_0, j_0) , the occlusion mask is defined as:

$$O(i_0, j_0) = \begin{cases} 1 & \text{if } I(i, j) < I(i_0, j_0) - \Delta \\ & \text{or } I(i, j) > I(i_0, j_0) + \Delta \\ 0 & \text{else} \end{cases} \quad (3.12)$$

$$i \in [i_0 - R_t/2, i_0 + R_t/2] \quad (3.13)$$

$$j \in [j_0 - R_t/2, j_0 + R_t/2] \quad (3.14)$$

here Δ is the threshold for the depth range. We take $\Delta = 200$ millimeters in our experiments, which is approximately the diameter of a human head.

3.1.4.3 3D Model Fitting

The 3D model is fitted onto every region detected by the previous step. For every location, a patch is extracted centered on that pixel from the preprocessed depth image. The patch is first normalized:

$$d_n(i, j) = d(i, j) - \min_{i,j}(d(i, j)) \quad (3.15)$$

$$i \in [i_0 - R_t, i_0 + R_t] \quad (3.16)$$

$$j \in [j_0 - R_t, j_0 + R_t] \quad (3.17)$$

Here, $d(i, j)$ is the depth value of pixel (i, j) . $d_n(i, j)$ is the normalized depth value. The summed square error between the circular patch and the 3D hemisphere is computed by:

$$E_r = \frac{1}{|O|} \sum_{i,j \in CR} O(i, j) \otimes |d_n(i, j) - T(i, j)|^2 \quad (3.18)$$

Here, \otimes represents an element-wise product (Hadamard product). If $E_r < E_\theta$, we believe a head is found.

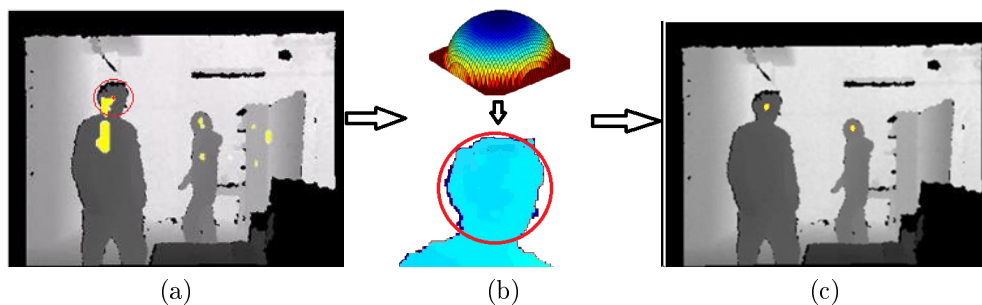


Figure 3.3: (a) Input of the 3D fitting algorithm. (b) Fitting process. (c) Output of 3D model fitting. Yellow dots indicate detection points.

Fig. 3.3. illustrates this stage and shows the result of the 3D matching.

3.1.5 Extract Contours

Up to this point, we have located the head in the depth image. In this section, we further find the whole contour of the person. The body contour may serve as a start point for many algorithms such as human body part/joints estimation, human pose estimation, and so on. In most cases, a human body appears as a continuous region in the depth image. This largely simplifies the processing of extracting the whole body region. The largest difficulty lies where the human body touches the background. In this case, the boundary of the body part cannot be distinguished from the background even with human eyes. As the feet always touch the ground in the image, it is a serious problem to segment the feet from the ground. We propose a simple solution to segment the feet from the floor. Since the floor is horizontal and the leg touches the

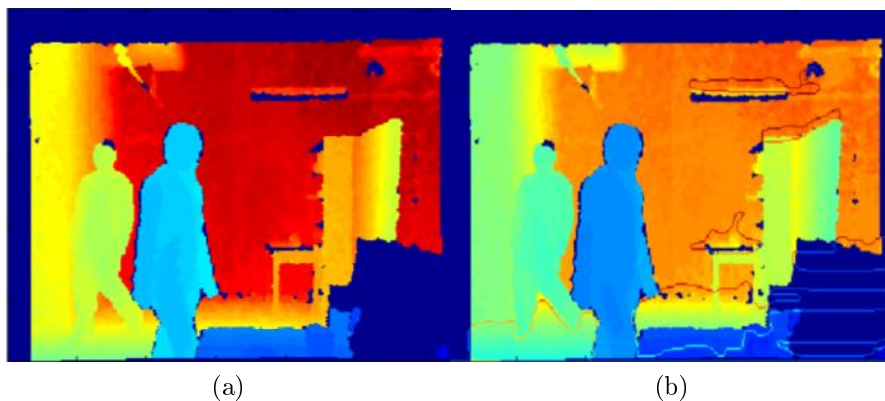


Figure 3.4: (a) Original depth array. Some parts of the body are merged with the ground plane and wall. (b) The input depth array to the region growing algorithm. The ground plane is delineated by the thresholded F filter response. The edges along the feet well separate the persons from the floor.

ground vertically, we propose a simple filter: $F = [1, 1, 1, -1, -1, -1]^T$ to extract the boundary between the legs/feet and the ground.

Since the floor near the bottom of the image is always closer to the camera, pixels on the floor will give a larger response to this filter F than the pixels on the person's legs and feet. With a proper threshold, we can easily find all the planar areas that are parallel to the floor. We add the edges of those planar areas to the original depth image and feed this into the region growing algorithm. Fig. 3.4 shows an example of extracting the edges of the planar surfaces.

We develop a simple and intuitive region growing algorithm to extract the whole body contours from the depth array. It is assumed that the depth values on the surface of a human object are continuous and vary within a

limited range. The algorithm starts with a seed location, which is the centroid of the head detected by 3D model fitting. The rule for growing a region is based on the similarity between the region and its neighboring pixels. The similarity between two pixels (x_1, y_1) and (x_2, y_2) in the depth array is defined as:

$$S(p(x_1, y_1), p(x_2, y_2)) = |d(x_1, y_1) - d(x_2, y_2)| \quad (3.19)$$

Here, S is similarity and $d(\cdot)$ returns the depth value of the pixel. The depth of a region is defined by the mean depth of all the pixels in that region:

$$d(R) = \frac{1}{N} \sum_{p(i,j) \in R} d(i, j) \quad (3.20)$$

The pixel of the highest similarity score with the region is added to the region in every loop until the similarity score exceeds a certain value. To prevent the region from growing onto the background when the person touches the background, we set a limit on the area of the whole region. The pseudocode of the region growing algorithm is summarized in Table 1. The results of the region growing algorithm are shown in Fig. 4.6.

Depending on the image quality and the scene, further refinement of the result can be incorporated after the region growing step to either make the contour better, or further adjust the human detection result. Simple morphological filtering may smooth the irregular contours, and the detection result can be further refined because the whole body of the person is supposed to be known at this stage. Some body properties may be employed to filter out false

Algorithm 1: Region Growing Algorithm

Input: seed s **Output:** region R

```
1 begin
2   Initialize:  $R = s$ ,  $d_{min} = 0$ ,  $area(R) = 1$ ,  $mean(R) = d_s$ 
3   while  $d_{min} < d_\theta \wedge area(r) < S_{max}$  do
4     for  $\{all\ neighboring\ pixels\ of\ region\ R\}$  do
5       Measure the difference of the pixel depth  $d_i$  and the
6       region mean  $mean(R)$ :  $d_1, d_2, \dots$ 
7        $d_{min} \leftarrow min(d_1, d_2, \dots)$ 
8       Add the pixel with the smallest distance  $d_{min}$  to the region
9        $R$ 
10      Update  $mean(R)$ 
11   return  $R$ 
```

positives, e.g. head should be at the top of the body region, the region should have a body part, etc. The criteria we employed are: head width, head upper radius (distance from the center of the detected head to the top of the person's contour), and area of the region. These are all computed from the region R . Note that although head radius is used in the previous 2D matching and 3D matching steps, that radius value may differ from the radius computed from the final contour.

3.1.6 Tracking

A simple tracking algorithm is proposed to track the person based on detection. Tracking in RGB image is usually based on color, the assumption is that the color of the same object in different time frames should be similar. There is no color information in depth images. We propose to utilize the 3D

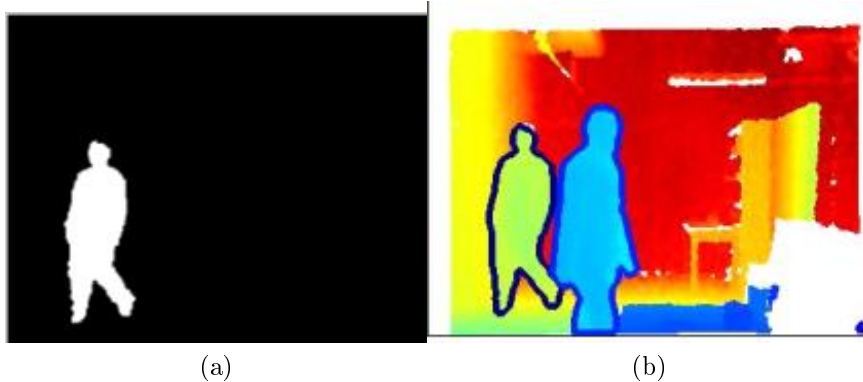


Figure 3.5: (a) Result of the region growing algorithm. (b) The extracted whole body contours superimposed on the depth map.

spatial information to track the persons. We assume the motion of the person is smooth, i.e. with limited acceleration.

The input are the head locations of the persons in each frame $P_t^{(i)}(x, y, Z), i = 1, \dots, N(t), t = 1, \dots, T$. $N(t)$ indicates the total number of persons at frame t . (x, y) are the image coordinates and Z is the depth value at pixel (x, y) . Since the x -axis and y -axis of the location is in image coordinates and the z -axis of the location is in real world coordinates, we transform all the axis to real world coordinates:

$$X = \frac{Z}{f}(x - x_0 + \delta_x) \quad (3.21)$$

$$Y = \frac{Z}{f}(y - y_0 + \delta_y) \quad (3.22)$$

where (X, Y) are the real world coordinates, (x_0, y_0) are the image center, δ_x and δ_y are the correction parameters for lens distortion. We set them to zero for our experiment since the person location does not need to be very accurate

for the tracking scenario.

In the first frame, we label the person in turn according to the detection order. In frame t , we match each person to one of the persons in the previous frame $t - 1$, the total number of possible matching is $C_{\max(N(t), N(t-1))}^{\min(N(t), N(t-1))}$. For each matching, we take the 3D coordinates of the persons in frame t : $P_t^{(i)}(X, Y, Z)$, and frame $t - 1$: $P_{t-1}^{(j)}(X, Y, Z)$, compute the speed of the person's motion $\vec{V}_t^{(ij)} = P_t^{(i)} - P_{t-1}^{(j)}$. We define an energy score E of status transformation as:

$$E_t^{(i,j)} = \sum_i (P_t^{(i)} - P_{t-1}^{(j)})^2 + \alpha(\vec{V}_t^{(ij)} - \vec{V}_{t-1}^{(i)})^2 \quad (3.23)$$

We choose the matching with the minimum energy score as the solution.

3.2 Experimental Results

The algorithm is tested on two datasets and it is compared with state of the art algorithms on depth images [34] and a traditional intensity based human detection algorithm using the HOG descriptor [19]. Both qualitative and quantitative results are given.

3.2.1 Datasets

Two datasets are collected each of which contains 100 depth images. They are captured by the Kinect for XBOX 360 in indoor environments. The resolution of the depth image is 640×480 . The depth value is given in millimeters and the points that failed to be measured are offset to 0. Figure 3.6 shows the image from the two datasets. In the first dataset, 0-2 persons appear in

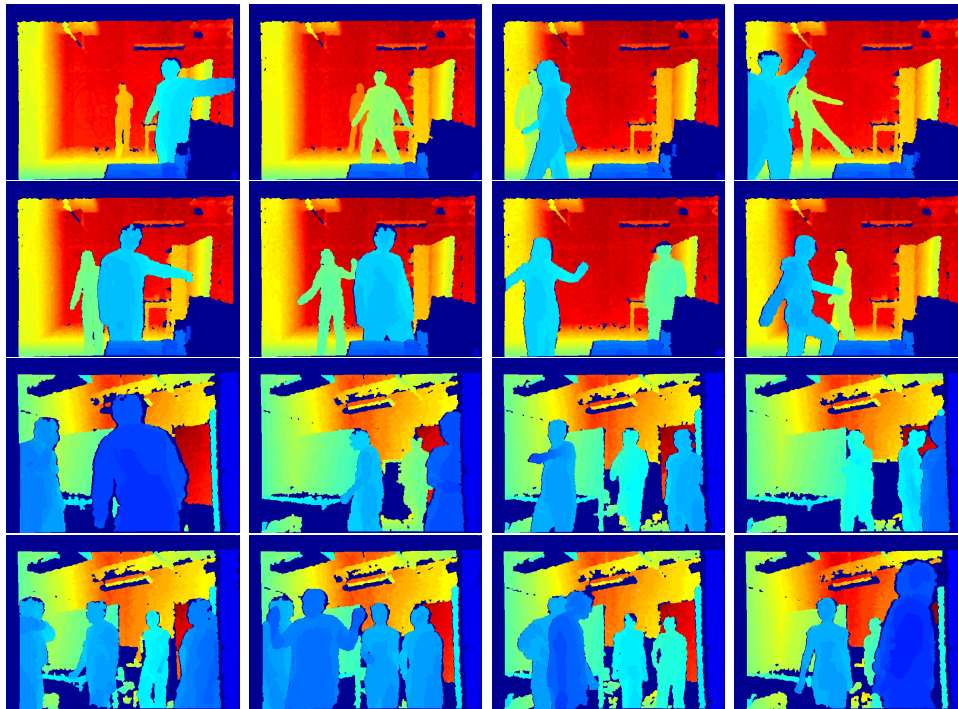


Figure 3.6: Example images from datasets. Images in the upper two rows are from dataset 1; images in the bottom two rows are from dataset 2.

each image. In the second dataset, 1-4 persons appear in each image. The background may contain tables, chairs, shelves, computers, an overhead lamp, and so on. The persons have a variety of poses; they may have interaction with others or the surrounding objects.

3.2.2 Detection Results

Fig. 3.6 shows some of the results of our algorithm. The quantitative result is given in Table 3.1. From the experimental result we can see our detection algorithm detects the person accurately in most cases. The false

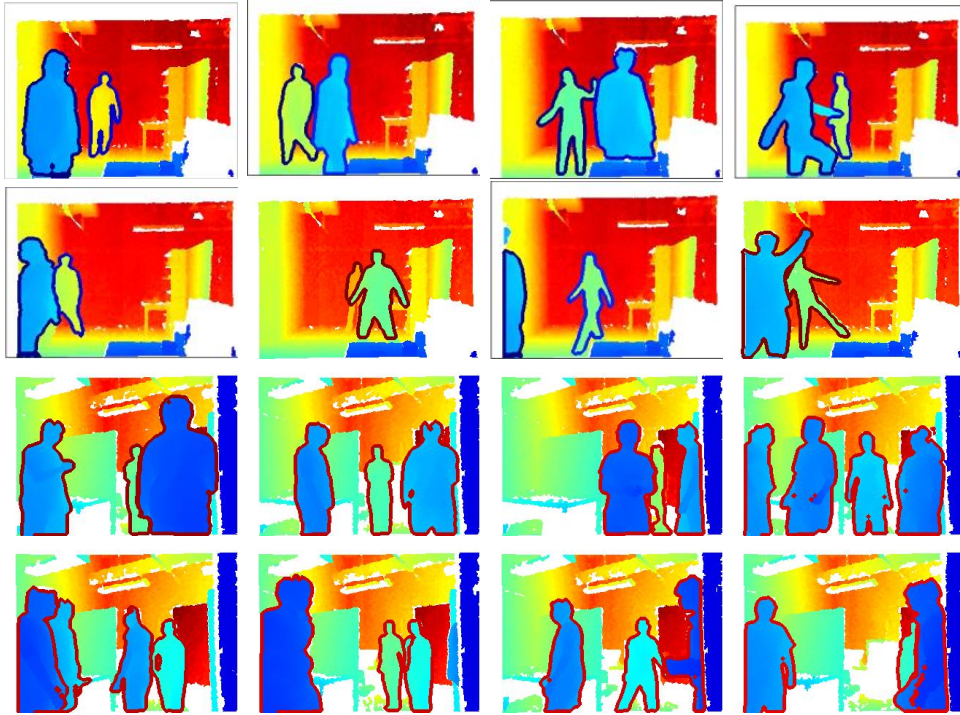


Figure 3.7: Examples of the human detection result.

	True Positive	True Negative	False Positive	False Negative	Precision %	Recall %	Accuracy %
1	169	266	0	7	100	96	98.4
2	251	298	2	20	99.2	92.6	96.1

Table 3.1: Accuracy of our algorithm on the two datasets

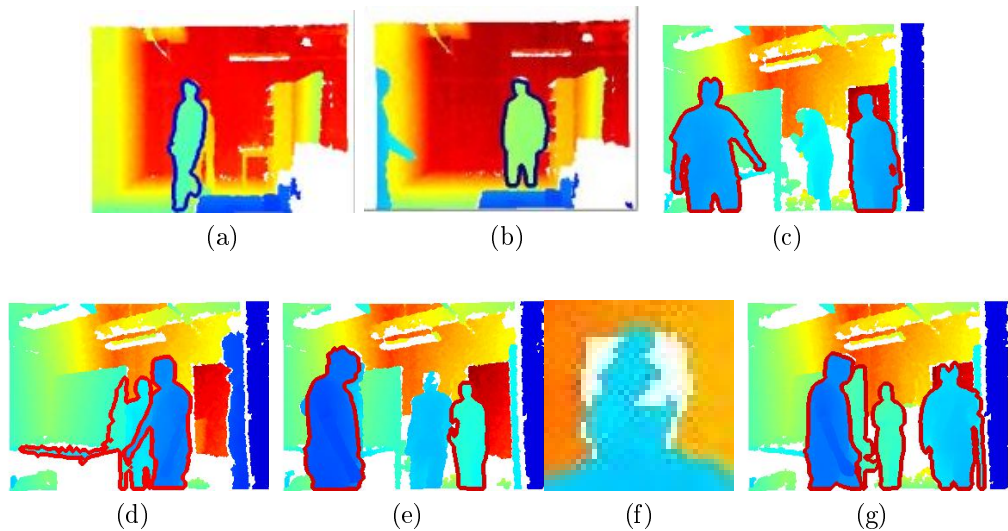


Figure 3.8: Failure cases.

positive rates are very low and false negative happens when the person’s head is occluded or the image quality of the head region is very bad. Figure 3.8 shows typical failure cases. In image (a), the head of the person at the back is occluded by the person in front. In image (b), half of the person/head is out of the frame. In image (c), the person is hiding his head. In image (d), the detection is correct, but the contour is not accurate because that person touches the background. In image (e), the quality of the head region is very bad. An enlarged image is shown in (f). Image (f) shows a false positive, the second detection from the left side is a shelf.

We compare our algorithm with state-of-the-art algorithms on human detection [19, 34, 77]. They are sliding-window based algorithms using HOG features on intensity images [19], HOD features on depth images [77], and

	Precision	Recall	Accuracy
Ikemura [34]	90.0%	32.9%	85.8%
HOD [77]	39.05%	75.86%	71.95%
HOG [19]	51.33%	62.37%	78.12%
Proposed	100%	96.0%	98.4%

Table 3.2: Comparison of performance

relational depth similarity features on depth images [34]. To prove the privilege of using depth data and the effectiveness of our algorithm, we also include the human detection algorithm performed on RGB data [19], and run this algorithm on the RGB images. (Because we did not store the corresponding RGB images when we originally took the dataset, we recaptured the RGB images later in the same room and with the same persons. Even though the RGB images and the depth images are not one to one corresponded, the detection difficulties are similar.) The result of the HOG pedestrian detection is shown in Figure 3.9. The first row shows examples of typical success cases. The second row shows that the background clutter causes confusion for the HOG descriptor. The third row shows that the whole body of the person must be in view to make the pedestrian detection algorithm work. Even though a small portion of the lower leg is out of the field of view, the pedestrian detector cannot detect the person. The fourth row gives examples when the algorithm totally missed the person even though the person is fully in view. The fifth row shows examples when the algorithm totally messed up the detection. It is clear that our algorithm give much better detection.

We perform the same preprocessing on the depth data and then run the

other two algorithms [34, 77]. We manually generated positive and negative windows from our dataset. About 0 to 500 windows are extracted from each frame and we subsample them and use the odd number of frames for training and even number of frames for testing. There are 770 positive examples and 2922 negative examples in the training set and 738 positive examples and 2930 negative examples in the test set. Table 3.2 shows the comparison of performances of all the methods. From table 3.2, we can see that our algorithm outperforms state of the art algorithms on this dataset. The sliding window based algorithm is better at handling the instances when the people in the frame are in an upright position. However, people in this dataset are presented in all kinds of postures and rotations.

3.2.3 Tracking Results

Fig. 3.10 shows the results of the tracking algorithm on dataset 1. 15 consecutive frames are shown, which includes two people walking past each other, one person gets occluded, and appears again.

3.3 Conclusion

In this chapter, I presented a human detection method that takes as input a single depth image. The algorithm outputs the head location and human body contour. This algorithm does not require background subtraction or motion detection. The experimental results show that the algorithm can effectively detect the persons in various poses and appearances from the depth



Figure 3.9: Detection Results using HOG pedestrian detection algorithm.

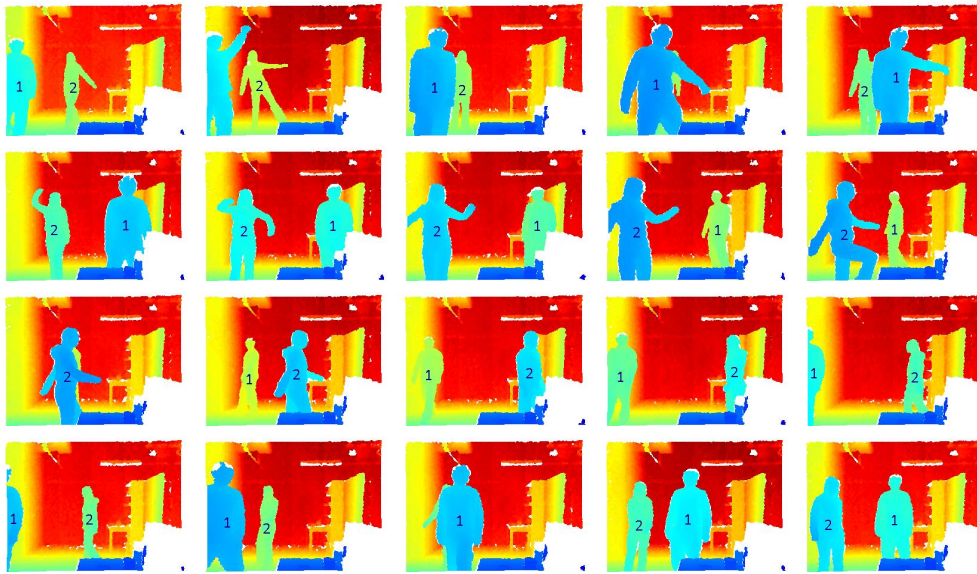


Figure 3.10: Tracking result. Results are shown at every two frames. The original frame rate is about 0.4 spf. The two tracked persons are labeled number one and number two respectively.

images, and it provides an accurate estimation of the whole body contour of the person. In addition, I proposed a tracking algorithm based on the detection results. The algorithm is generally applicable to depth images acquired by other types of range sensors.

The advantages of this algorithm are the following. Firstly, the method may easily adjust to new datasets, no training is needed. Secondly, the algorithm uses a bilayer detection process with 2D chamfer matching in the first layer which largely reduces computational cost. Thirdly, it does not assume certain human poses or motion for accurate detection. Furthermore, this algorithm does not use background subtraction, thus applicable to cases where the camera is non-stationary. Last but not least, due to the nature of the device, the method is generally more robust to illumination changes, and may work in total darkness, as long as the environment does not contain an excessive amount of light of the specific wavelength used by the device. The limitation is the high dependency on accurate head detection, which implies that if the head is totally occluded or if the person is wearing a strange shaped hat, it may not be detected.

If the corresponding RGB imagery is available, this detection process can run parallelly with the detection algorithm on RGB image. The results from the two different channels may be combined to provide more accurate detection results.

Chapter 4

Histogram of Skeletal Joint Feature for Action Recognition

In this Chapter, I describe the proposed 3D skeletal joint feature. A reference coordinate is aligned to the joints on the torso of the person. 12 informative joints are selected, the polar angle and azimuth angle are computed and vote into the bins of every 30 degrees, which generates a compact feature called histograms of 3D joints (HOJ3D). To make the representation robust against minor posture variation, votes of 3D skeletal joints are cast into neighboring bins using a Gaussian weight function. The collection of HOJ3D vectors from training sequences are first reprojected using LDA and then clustered into k posture words. By encoding sequences of skeletons into sequential words, action sequences are classified using HMMs [67]. Experiments show that this algorithm achieves superior results on our challenging dataset and also outperforms the state-of-the art algorithms [49, 100, 109] on activity recognition from depth images.

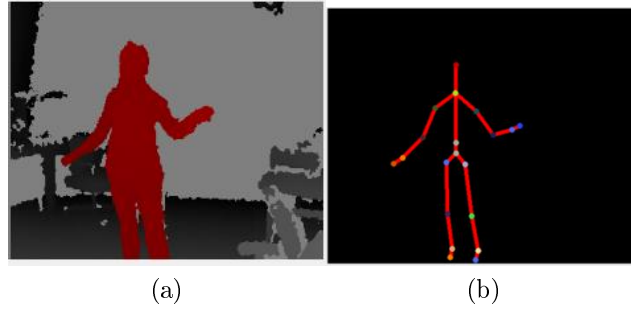


Figure 4.1: (a) Depth image. (b). Skeletal joints locations.

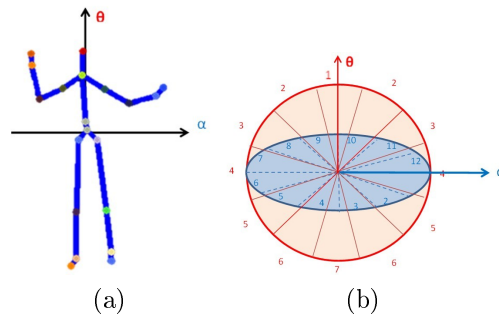


Figure 4.2: (a) Reference coordinates the HOJ3D. (b) Modified spherical coordinate system for joint location binning.

4.1 Algorithm

4.1.1 Body Part Inference and Joint Position Estimation

The human body is an articulated system of rigid segments connected by joints and human action is considered as a continuous evolution of the spatial configuration of these segments (i.e. body postures) [105]. Here, we use 3D joint locations to build a compact representation of postures. 3D joint locations can be extracted from a depth video [75], which include hip center, spine, shoulder center, head, L/ R shoulder, L/ R elbow, L/ R wrist, L/ R hand, L/ R hip, L/ R knee, L/ R ankle and L/ R foot. Fig. 4.1 shows an example of 3D skeletal joint locations of a depth frame. Among these joints, hand and wrist and foot and ankle are very close to each other and thus redundant for the description of body part configuration. In addition, spine, neck, and shoulder do not contribute discerning motion while a person is performing indoor activities. Therefore, I compute the histogram based representation of postures from 12 of the 20 joints, including head, L/ R elbow, L/ R hands, L/ R knee, L/ R feet, hip center and L/ R hip. The hip center is taken as the center of the reference coordinate system, the vector direction from the left hip joint to the right hip joint is defined as the α direction, the normal vector of the floor plane is defined as the θ direction. The rest 9 joints are used to compute the 3D spatial histogram. The estimated joint locations provide information regarding the direction the person. This enables us to compute the reference direction of a person independent of the viewpoints.

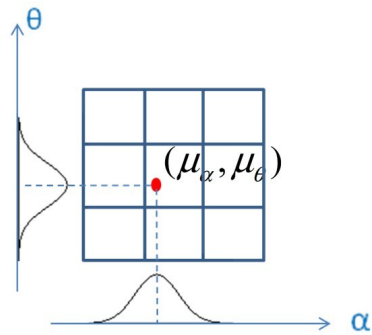


Figure 4.3: Voting using a Gaussian weight function.

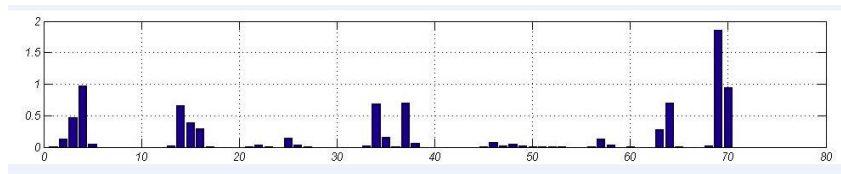


Figure 4.4: Example of the HOJ3D of a posture.

4.1.2 HOJ3D as Posture Representation

The estimation of 3D skeleton from RGB imagery is subject to error and significant computational cost. With the depth image, we may acquire the 3D locations of the body parts in real-time with better accuracy. Eventhough, the joint locations is not perfect, inaccurate estimations occur when parts of the body is occluded. I propose a compact and viewpoint invariant representation of postures based on 3D skeletal joint locations, which also deals with moderate estimation error of the joint locations.

4.1.2.1 Spherical Coordinates of Histogram

The methodology is designed to be view invariant, i.e., descriptors of the same type of pose are similar despite being captured from different viewpoints. This is achieved by aligning a spherical coordinate with the person’s direction, as shown in Fig. 4.2(a). We define the center of the spherical coordinates as the hip center joint. Define the horizontal reference vector α to be the vector from the left hip center to the right hip center projected on the horizontal plane (parallel to the ground), and the zenith reference vector θ as the vector that is perpendicular to the ground plane and passes through the coordinate center.

The 3D space is partitioned into n bins as shown in Fig. 4.2(b) ($n=84$ in the experiment). The inclination angle is divided into 7 bins from the zenith vector θ : $[0, 15]$, $[15, 45]$, $[45, 75]$, $[105, 135]$, $[165, 180]$. Similarly, from the reference vector α , the azimuth angle is divided into 12 equal bins with 30 degrees resolution. The radial distance is not used in this representation to make the method scale-invariant. With our spherical coordinate, each 3D joint falls into a unique bin.

4.1.2.2 Probabilistic Voting

The HOJ3D descriptor is computed by casting the rest 9 joints into the corresponding spatial histogram bins. For each joint location, weighted votes are contributed to the geometrically surrounding 3D bins. To make the representation robust against moderate errors of joint locations, we vote the

3D bins using a Gaussian weight function:

$$p(X, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X-\mu)\Sigma^{-1}(X-\mu)} \quad (4.1)$$

, where $p(X, \mu, \Sigma)$ is Gaussian probability density function with mean vector μ and co-variance matrix Σ (a identity matrix is used for simplicity). Each joint vote into the bin which it falls in and the 8 neighboring bins. We calculate the probabilistic voting on θ and α separately since they are independent. The probabilistic voting for each of the 9 bins is the product of the probability on α direction and θ direction. Let the joint location be (μ_α, μ_θ) . The vote of a joint location to bin $[\theta_1, \theta_2]$ is

$$p(\theta_1 < \theta < \theta_2; \mu_\theta, 1) = \Phi(\theta_2; \mu_\theta, 1) - \Phi(\theta_1; \mu_\theta, 1) \quad (4.2)$$

, where Φ is the CDF of Gaussian distribution. Similarly, the vote of joint location (μ_α, μ_θ) to the bin $[\alpha_1, \alpha_2]$ is

$$p(\alpha_1 < \alpha < \alpha_2; \mu_\alpha, 1) = \Phi(\alpha_2; \mu_\alpha, 1) - \Phi(\alpha_1; \mu_\alpha, 1) \quad (4.3)$$

Then, the probability voting to bin $[\alpha_1, \alpha_2], [\theta_1, \theta_2]$ is:

$$\begin{aligned} & p(\theta_1 < \theta < \theta_2, \alpha_1 < \alpha < \alpha_2; \mu, I) \\ &= p(\theta_1 < \theta < \theta_2, \mu_\theta, 1) \cdot p(\alpha_1 < \alpha < \alpha_2, \mu_\alpha, 1) \end{aligned} \quad (4.4)$$

The votes are accumulated over the 9 joints. A posture is represented by an n-bin histogram. Fig. 4.4 shows an instance of the computed histogram.

4.1.2.3 Feature Extraction

Linear discriminant analysis (LDA) is performed to extract the dominant features. LDA is based on the class specific information which maximizes the ratio of between-class scatter and the within-class scatter matrix. The LDA algorithm looks for the vectors in the underlying space to create the best discrimination between different classes. In this way, a more robust feature space can be obtained that separates the feature vectors of each class. In our experiment, we reduce the dimension of the HOJ3D feature from n dimensions to $n_{\text{Class}}-1$ dimensions.

4.1.3 Vector Quantization

As each action is represented by an image sequence or video, the key procedure is to convert each frame into an observation symbol so that each action may be represented by an observation sequence. Note that the vector representation of postures is in a continuous space. In order to reduce the number of observation symbols, we perform vector quantization by clustering the feature vectors. We collect a large collection of indoor postures and calculate their HOJ3D vectors. We cluster the vectors into K clusters (a K -word vocabulary) using K -means. Then each posture is represented as a single number of the visual word. In this way, each action is a time series of the visual words.

4.1.4 Action Recognition Using Discrete HMMs

Human actions are modeled and recognized by the discrete HMM technique similar to what Rabiner did in speech recognition [11]. In discrete HMM, discrete time sequences are treated as the output of a Markov process whose states cannot be directly observed. Previously, each action sequence has been coded as a vector of posture words, this vector is used to learn the HMM model and this model is used to predict for the unknown sequence.

A HMM that has N states $S = \{s_1, s_2, \dots, s_N\}$ and M output symbols $Y = y_1, y_2, \dots, y_M$ is fully specified by the triplet $\lambda = A, B, \pi$. Let the state at time step t be S_t . The $N \times N$ state transition matrix A is,

$$A = \{a_{ij} | a_{ij} = P(s_{t+1} = q_j | s_t = q_i)\} \quad (4.5)$$

The N times M output probability matrix B is,

$$B = \{b_i(k) | b_i(k) = P(v_k | s_t = q_i)\} \quad (4.6)$$

And the initial state distribution vector π is

$$\pi = \{\pi_i | \pi_i = P(s_1 = q_i)\} \quad (4.7)$$

A HMM model is constructed for each of the actions. Then, I take an action sequence $V = v_1, v_2, \dots, v_T$ and calculate its probability of a model λ for the observation sequence, $P(V|\lambda)$ for every model, which can be solved by using the forward algorithm. Then the action can be classified as the one which has

No.	1	2	3	4	5
Mean	43.60	34.15	25.60	35.50	58.15
SD	8.89	9.40	6.44	11.89	27.04
No.	6	7	8	9	10
Mean	11.95	10.30	15.05	45.70	31.00
SD	4.10	4.24	7.72	16.30	20.14

Table 4.1: The mean and standard deviation of the sequence lengths measured by number of frames at 30 fps.

the largest posterior probability.

$$decision = \arg \max_{i=1,2,\dots,M} \{L_i\} \quad (4.8)$$

$$L_i = Pr(O|H_i) \quad (4.9)$$

Where L_i indicates the likelihood of i -th HMM H_i and M number of activities. This model can compensate for the temporal variation of the actions caused by differences in the duration of performing the actions.

4.2 Experiments

The algorithm is tested on a challenging new dataset I collected and made publicly available. In addition, it is also evaluated on the public MSRAction3D dataset and compared with state-of-the-art algorithms [49, 100, 109].

4.2.1 Data

To test the robustness of the algorithm, we collected a dataset containing 10 types of human actions in indoor settings. We take the sequence using a single stationary Kinect. The RGB images and depth maps were captured

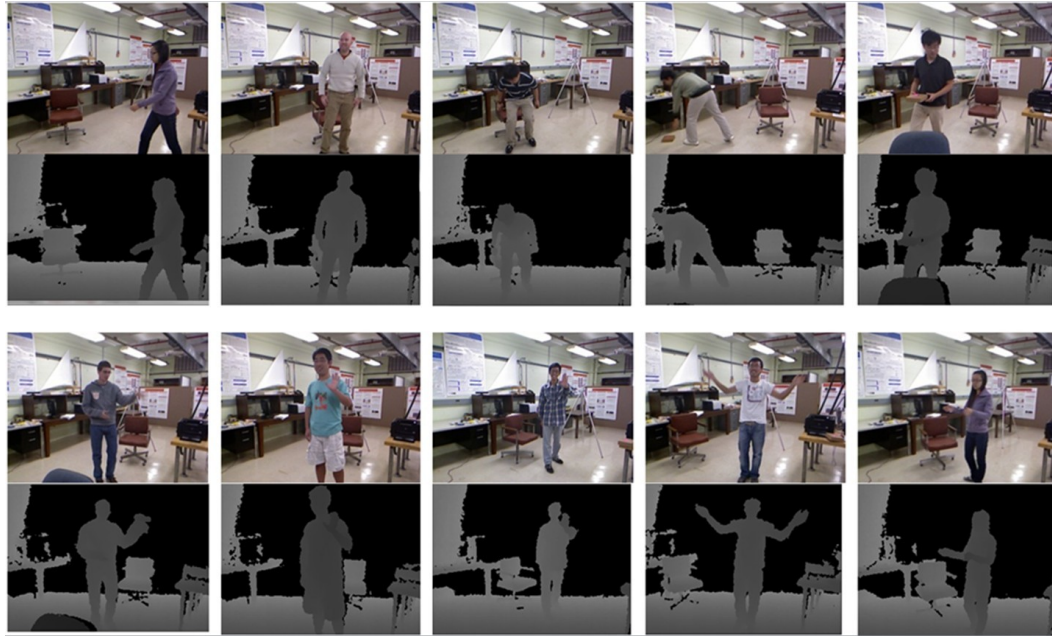


Figure 4.5: Sample images from videos of the 10 activities in the database. Note only depth images are used in the proposed algorithm. Action type from left to right, top to bottom: *walk*, *stand up*, *sit down*, *pick up*, *carry*, *throw*, *push*, *pull*, *wave hands*, *clap hands*.









	Right view	Frontal view	Right view	Back view
No. 5 Carry				
No. 4 Pick up				

Figure 4.6: Different views of the actions are presented in the dataset.

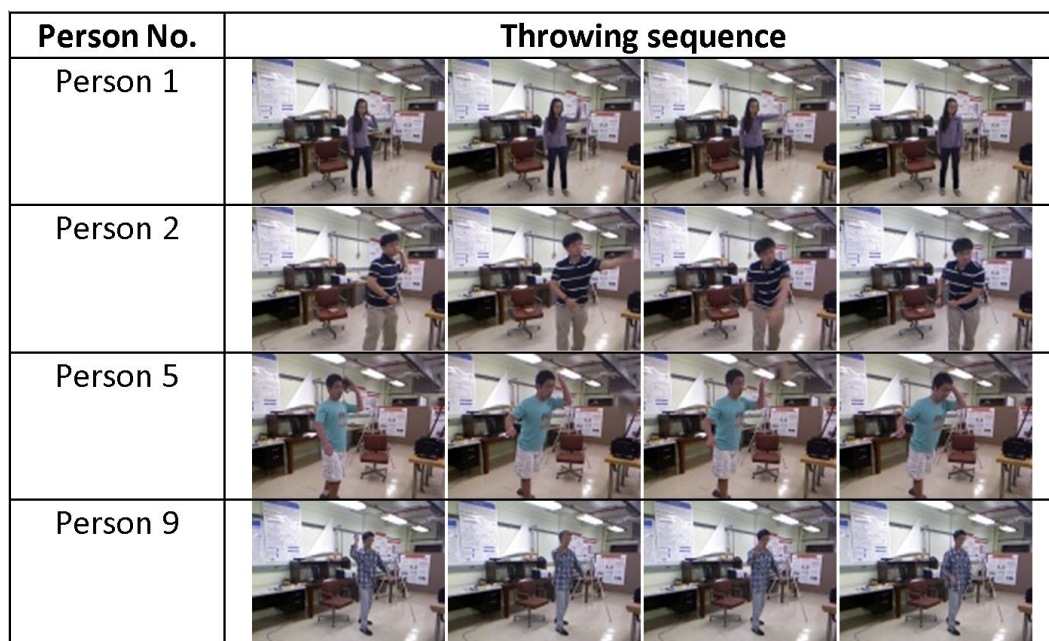


Figure 4.7: The variations of subjects performing the same action.

Action	Accuracy	Action	Accuracy
Walk	96.5%	Throw	59.0%
Sit down	91.5%	Push	81.5%
Stand up	93.5%	Pull	92.5%
Pick up	97.5%	Wave	100%
Carry	97.5%	Clap hands	100%
Overall: 90.92%			

Table 4.2: Recognition rate of each action type

Algorithm	Accuracy
STIP(Harris3D+HOG3D) [109]	80.8%
pair-wise joint distance	83.4%
Skeleton Joint Features [109]	87.9%
Proposed	90.92%

Table 4.3: Comparisons on the UTKinect dataset

AS1	AS2	AS3
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup & throw

Table 4.4: The three subsets of actions used for the MSR Action3D dataset.

	Li [49]	Yang [100]	Proposed
AS1One	89.5	94.7	98.5
AS2One	89.0	95.4	96.7
AS3One	96.3	97.3	93.5
AS1Two	93.4	97.3	98.6
AS2Two	92.9	98.7	97.9
AS3Two	96.3	97.3	94.9
AS1CrSub	72.9	74.5	98.0
AS2CrSub	71.9	76.1	85.5
AS3CrSub	79.2	96.4	79.0

Table 4.5: Recognition results of our algorithm on the MSRAction3D dataset, compared with Li et al. [49] and Yang et al. [100]. In test one, 1/3 of the samples were used as training samples and the rest as testing samples. In test two, 2/3 samples were used as training samples. In the cross subject test, half of the subjects were used as training and the rest of the subjects were used as testing.

at 30 frames per second (FPS). The resolution of the depth map is 320×240 and resolution of the RGB image is 640×480 . The 10 actions include: walk, sit down, stand up, pick up, carry, throw, push, pull, wave and clap hands. Each action was collected from 10 different persons for 2 times: 9 males and 1 female. One of the persons is left-handed. Altogether, the dataset contains 6220 frames of 200 action samples. The length of sample actions ranges from 5 to 120 frames. Sample RGB images from the dataset are shown in Fig. 3.4. Note that we only use the information from the depth image for action recognition in our algorithm; the RGB sequences are just for illustration.

As shown in Fig. 4.6, we took action sequences from different views to highlight the advantages of our representation. In addition to the varied views, our dataset features 3 other challenges which are summarized as follows. First, there is significant variation among different realizations of the same action. For example, in our dataset, some actors pick up objects with one hand while others prefer to pick up the objects with both hands. Fig. 4.7 is another example, individuals can toss an object with either their right or left arm and producing different trajectories. Second, the durations of the action clips vary dramatically. Table 4.1 shows the mean and standard deviation of individual action length. In this table, the standard deviation of the carry sequence lengths is 27 frames, while the mean duration of carry is 48 frames longer than that of push. Third, object-person occlusions and body part out of field of view (FOV) also add to the difficulty of this dataset.

4.2.2 Experimental Results

The proposed algorithm is tested on the new dataset using leave-one-out cross validation (LOOCV). As there is randomness in the initialization of the cluster centroids and the HMM algorithm, 20 experiments is runned and the mean performance is reported in Table 4.2. The number of clusters is $K=125$, and the number of states is $N=6$. By experiments, the overall mean accuracy is 90.92%, the best accuracy is 95.0% and the standard deviation is 1.74%. On a 2.93GHz Intel Core i7 CPU machine, the estimation of 3D skeletal joints and the calculation of HOJ3D is real-time using C implementation. The average testing time of one sequence is 12.5ms using Matlab. The total processing is real-time. I compared the performance with three other features. The first one is spatio-temporal features, which use Harris3D to find the spatio-temporal interest points and use HOG3D feature to describe the local patches [109]. The second feature is the widely used pair-wise joint distance feature. The third feature is consist of three parts [100]: (1) current posture: pair-wise joint distances in current posture; (2) motion: joints difference between current posture and the original (in the first frame); and (3) offset: joints differences between current posture and the previous one. A concatenation of the three feature vectors is used to represent the feature for a specific action. From table 4.3 we can see that our algorithm outperforms the other features on this challenging dataset which contains various viewing angles.

The algorithm is also tested on the public MSRAction3D database that contains 20 actions: high arm wave, horizontal arm wave, hammer, hand catch,

forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing and pickup & throw. As originally proposed in [49], the actions are divided into 3 subsets each comprising 8 actions (see table 4.4). I use the same parameter settings as previously. Each test is repeated 20 times, and the average performance is shown in Table 4.5. I compared the performance with Li et al. [49] and Yang et al. [100]. It can be seen that the proposed algorithm achieves considerably higher recognition rates than Li et al. [49] in all the testing setups on AS1 and AS2. On AS3, our recognition rate is slightly lower. It is stated in [49] that the goal of AS3 was intended to group complex actions together. However, Li et al.'s algorithm actually achieves much higher recognition accuracy on this complex action set while ours have higher accuracy on the other two action set. We conjecture the reason to be that the complex actions effects adversely the HMM classification when the number of training samples is small. Yang et al.'s algorithm is published after my HOJ3D feature, the feature is based on differences of skeleton joints [100]. From table 4.5, we can see that my proposed work perform better on 5 of the 9 action sets.

4.3 Conclusion

This chapter presents a methodology to recognize human action as time series of representative 3D poses. It takes as input 3D skeletal joints locations inferred from depth maps. A compact representation of postures

named HOJ3D is proposed that characterizes human postures as histograms of 3D joint locations within a modified spherical coordinate system. A posture vocabulary is built by clustering HOJ3D vectors calculated from a large collection of postures. Discrete HMMs are learned and used to classify sequential postures into action types. The major components of the algorithm are real-time, which include the extraction of 3D skeletal joint locations, computation of HOJ3D, and classification. Experimental results show the salient advantage of the view invariant representation and the excellent performance of the algorithm.

Chapter 5

Spatio-Temporal Depth Cuboid Similarity Feature for Action Recognition

In this Chapter, I describe the spatio-temporal features I developed for depth video. Local spatio-temporal interest points (STIPs) and the resulting features from RGB videos have been proven successful at activity recognition that can handle cluttered backgrounds and partial occlusions. I design its counterpart in depth video and show its efficacy on activity recognition. A filtering method is employed to extract STIPs from depth videos (called DSTIP) that effectively suppress the noisy measurements and find the salient locations in the video. Further, a novel depth cuboid similarity feature (DCSF) is designed to describe the local 3D depth cuboid around the DSTIPs with an adaptable supporting size. This feature is tested on activity recognition application using the public MSRAction3D, MSRDailyActivity3D datasets and our own dataset. Experimental evaluation shows that this approach outperforms state-of-the-art activity recognition algorithms on depth videos, and the framework is more widely applicable than existing approaches. Detailed comparisons with other features and analysis of choice of parameters are given as a guidance for applications.

5.1 Algorithm

5.1.1 DSTIP Detection

As much of the work on interest point detection, a response function is computed at each pixel in the 3D spatio-temporal volume. Our response function is calculated by application of separable filters.

5.1.1.1 Spatio-Temporal Filtering

First, a 2D Gaussian smoothing filter is applied onto the spatial dimensions:

$$\mathbf{D}_s(x, y, t) = \mathbf{D}(x, y, t) * g(x, y | \sigma) \quad (5.1)$$

where $*$ denotes convolution, \mathbf{D} and \mathbf{D}_s denote the original depth volume and that after spatial filtering respectively. $g(x, y; \sigma)$ is a 2D Gaussian kernel:

$$g(x, y | \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} \quad (5.2)$$

σ controls the spatial scale along x and y . Then we apply a temporal filter along the t dimension:

$$\mathbf{D}_{st}(x, y, t) = \mathbf{D}_s(x, y, t) * h(t | \tau, \omega) \circ \bar{s}(x, y, t | \tau) \quad (5.3)$$

where \mathbf{D}_{st} denotes the depth volume after spatio-temporal filtering. \circ denotes element wise matrix multiplication and $h(t | \tau, \omega)$ is a 1D complex Gabor filter:

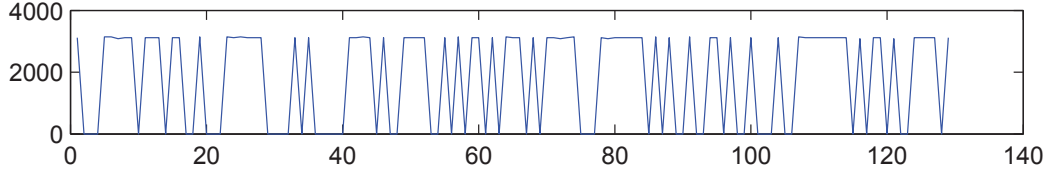
$$h(t | \tau, \omega) = e^{-t^2/2\tau^2} \cdot e^{2\pi i \omega t} \quad (5.4)$$

where τ controls the temporal scale of the filter. We use $\omega = 0.6/\tau$. $\bar{s}(x, y, t | \tau)$ is a correction function for the noise of the depth sequence at location (x, y, t) . τ is the same control parameter as in the Gabor filter. The next section introduces the correction function in detail.

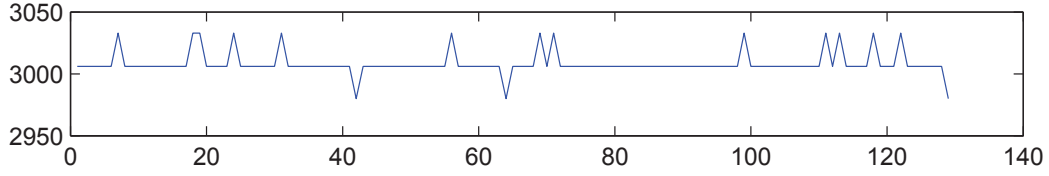
5.1.1.2 Noise Suppression

In RGB videos, smoothing functions usually serve to suppress noise. The reason we choose a correction function instead of using filters is based on the different nature of the noise in depth videos. One may divide the noise in depth videos into three categories: The first category of noise comes from the variation of the sensing device, which is evenly distributed throughout the entire image, the magnitude of which is comparatively small. The second category of noise occurs around the boundary of objects, the values jump from the depth of the background to the depth of the foreground, back and forth frequently. The magnitude of the jump can be a few thousand (mm). The third category of noise is the "holes" that appear in the depth images, caused by special reflectance materials, fast movements, porous surfaces, and other random effects. The magnitude of the noise can be a few thousand (mm) as well. Figure 5.1 gives the temporal evolution of pixel values at different locations in the scene.

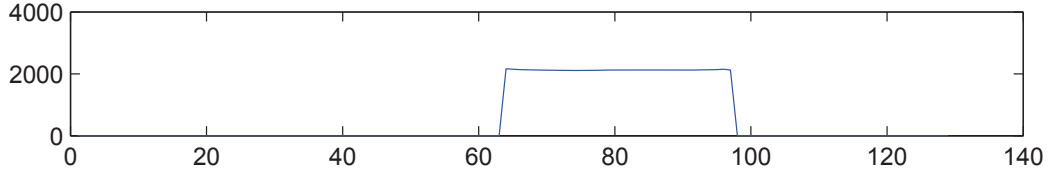
The first category is similar to the noise in RGB images, it is usually less distinguishable than real movements. This noise may be reasonably removed using smoothing filters, but in the second and third categories, the magnitude



(a) Signal from pixel on object boundary: the value flips from 0 to about 3000 (mm) at a high frequency.



(b) Signal from pixel in the middle of a static object, the value fluctuates around $3006 \pm 27(mm)$.



(c) Signal from pixel where movement happens.

Figure 5.1: Temporal evolution of pixel values at different locations in the scene.

of the noise is usually many times larger than real movements. We can hardly smooth out the noise while leaving the real movement signals unaffected.

The flip of the signal caused by sensor noise usually happens much faster than human movements, and it can happen from once to dozens of times during the whole video. In view of this, we calculate the average duration of the flip of the signal, and use it as a correction function:

$$s(x, y, t_0 | \tau) = \frac{\sum_{i=1}^{n_{fp}} \delta t_i(x, y)}{n_{fp}(x, y)} \quad (5.5)$$

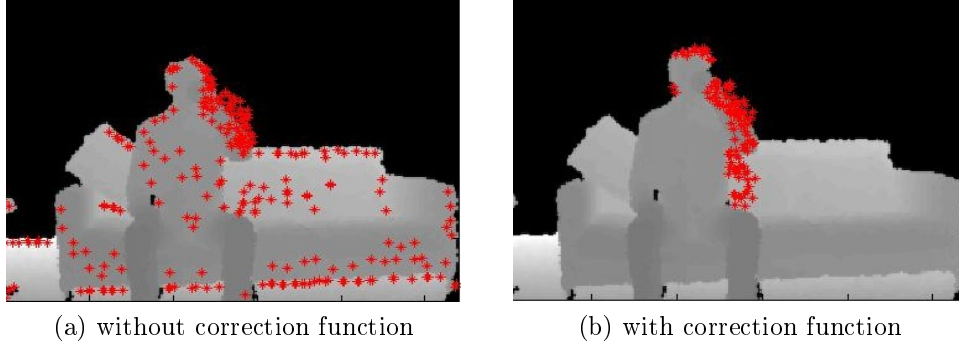


Figure 5.2: DSTIPs projected onto x - y dimensions on top of one frame of the video *drink*

where $n_{fp}(x, y)$ is the total number of flips during the time interval $[t_0 - \tau, t_0 + \tau]$ at location (x, y) , and $\delta t_i(x, y)$ is the duration of the i -th flip. We define the number of flips as the number of zero-crossing of the normalized signal $\tilde{d}(t) = d(t) - (d(t)_{max} + d(t)_{min})/2$.

This correction function is an indicator of the noise-signal ratio of the pixel at location (x, y, t) during interval $[t_0 - \tau, t_0 + \tau]$. It has a higher value at the pixels where real movement happens thus highlight those movements. We take a threshold so that it only affects the noises and does not discriminate between different movements:

$$\bar{s} = \begin{cases} s_0, & \text{if } s > s_0 \\ s, & \text{else} \end{cases} \quad (5.6)$$

where s_0 is selected to best separate the value $s(x, y, t)$ at the location of noises and location of real motions (e.g. $s_0 = 2$). Figure 5.2 shows the DSTIPs before and after the correction function. We can see the correction function effectively removes interest points resulting from noise.

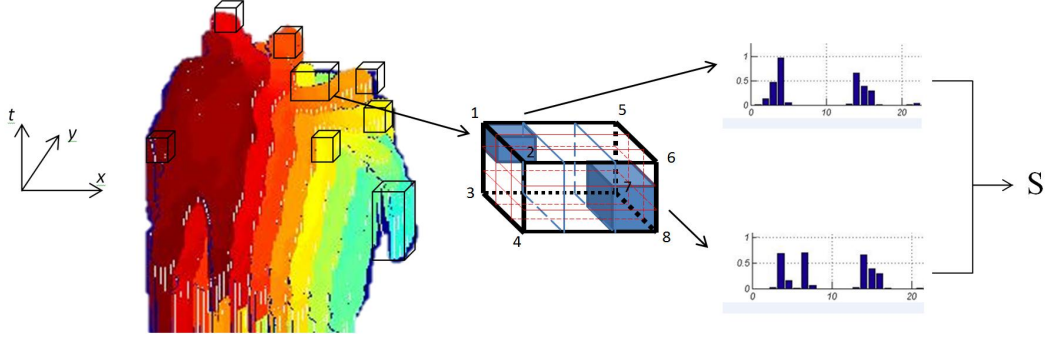


Figure 5.3: Illustration of extracting DCSF from depth video

5.1.1.3 Interest Point Extraction

Finally, we take the response as:

$$\mathbf{R}(x, y, t) = \|\mathbf{D}_{st}(x, y, t)\|_2^2 \quad (5.7)$$

The overall response can be written in a closed form:

$$\mathbf{R}(x, y, t) = (\mathbf{D} * g * h_{ev} \circ \bar{s})^2 + (\mathbf{D} * g * h_{od} \circ \bar{s})^2 \quad (5.8)$$

$$h_{ev}(t | \tau, \omega) = \cos(2\pi\omega t) e^{-t^2/2\tau^2} \quad (5.9)$$

$$h_{od}(t | \tau, \omega) = \sin(2\pi\omega t) e^{-t^2/2\tau^2}$$

DSTIP is selected at the local maximum of \mathbf{R} in spatio-temporal domains and also in scale domain. We take the local maximum with top N_p largest response value as the DSTIPs for each video.

5.1.2 Interest Point Description

Here we propose a descriptor for the local 3D cuboid centered at DSTIP. Note it is 3D instead of 4D because the depth image is a function of x and y ,

not all 3D points $\{x, y, z\}$, but it still provides useful information along the z dimension.

5.1.2.1 Adaptable Supporting Size

We extract a 3D cuboid which contains the spatio-temporally windowed pixel values around the DSTIP. Considering objects appear smaller in the image at a farther distance, we design the cuboid size to be adaptable to the depth. We define the spatial size of the cuboid to be proportional to the scale at which it was detected and inversely proportional to the depth at which it locates:

$$\Delta_x^{(i)} = \Delta_y^{(i)} = \sigma \frac{L}{d^{(i)}} \quad (5.10)$$

where σ is the scale at which the i -th cuboid was detected, and $d^{(i)}$ denotes the depth of the i -th cuboid. Notice that we do not take the depth pixel value at the interest point $\mathbf{D}(x_i, y_i, t_i)$ as $d^{(i)}$, because the DSTIP sometimes lands at the edge of body parts. Instead, we compute the minimum non-zero depth value in the 2τ time interval round the location (x_i, y_i, t_i) , i.e. $\{\mathbf{D}(x_i, y_i, t_i - \tau), \dots, \mathbf{D}(x_i, y_i, t_i + \tau)\}$. This usually gives the depth we want for the cuboid locations. In this way, the size of the cuboid is adjusted according to the real-world size of the object, which corresponds to smaller pixel-size at farther distances and vice-versa. This renders noticeable improvement as compared to a fixed pixel size in our experiments.

The side length of the temporal dimension of a cuboid is simply defined

as:

$$\Delta_t^{(i)} = 2\tau \quad (5.11)$$

cuboid similarity feature

Different from RGB data, depth data lacks texture, and is inherently noisy. We define a DCSF feature based on the self-similarity to encode the spatio-temporal shape of the 3D cuboid, and we show in Section 5.2 that this feature is better than other commonly used features.

As shown in Fig. 5.3, we divide the cuboid into $n_{xy} \times n_{xy} \times n_t$ voxels. (We cut the borders when needed to make sure each voxel contains an integer number of pixels). We define the block as containing $1 \times 1 \times 1$ to $n_{xy} \times n_{xy} \times n_t$ voxels.

We compute a histogram of the depth pixels contained in each block, normalize them to make the total value of every histogram to be 1. Let the histogram calculated from block p and q be h_p and h_q respectively, we use the Bhattacharyya distance to define the similarity:

$$S(p, q) = \sum_{n=1}^M \sqrt{h_p^{(n)} h_q^{(n)}} \quad (5.12)$$

which describes the depth relationship of the two blocks. M denotes the number of histogram bins. Note in this definition, the length of the feature depends on n_{xy} and n_t only, it does not relate to the actual size of the cuboid which offers greater freedom for the interest point detection and the cuboid extraction process.

We generate a feature vector by concatenating the similarity scores for all combinations of blocks. Varying spatial-size from 1×1 to $n_{xy} \times n_{xy}$ gives $n_{xy}(n_{xy} - 1)(2n_{xy} - 1)/6$ possibilities, varying temporal-size from 1 to n_t gives $n_t(n_t + 1)/2$ possibilities. In total, the number of blocks N_b generated by varying the number of voxels it contains is at the order of $n_t^2 n_{xy}^3 / 6$, and the total length of the DCSF feature is $C_{N_b}^2$.

To reduce computational cost, we use integral histograms [66] to compute the depth histograms rapidly. We quantize the depth pixels into M bins, $M = (d_{max} - d_{min})/\Delta d$, where Δd is chosen according to the spatial level of movements to recognize, e.g. $\Delta d = 100mm$. Then we generate M quantized video volumes $\mathbf{Q}^{(n)}, n = 1, \dots, M$, corresponding to the M bins:

$$\mathbf{Q}^{(n)}(x, y, t) = \begin{cases} 1, & \text{if } (n-1)\Delta d + 1 \leq \mathbf{D}(x, y, t) \leq n\Delta d \\ 0, & \text{else} \end{cases} \quad (5.13)$$

We compute an integrated video volume $\mathbf{I}^{(n)}, n = 1, \dots, M$ for each of the quantized video volume $\mathbf{Q}^{(n)}$:

$$\begin{aligned} r^{(n)}(x, y, t) &= r^{(n)}(x, y - 1, t) + \mathbf{Q}^{(n)}(x, y, t) \\ c^{(n)}(x, y, t) &= c^{(n)}(x - 1, y, t) + r^{(n)}(x, y, t) \\ \mathbf{I}^{(n)}(x, y, t) &= \mathbf{I}^{(n)}(x, y, t - 1) + c^{(n)}(x, y, t) \end{aligned} \quad (5.14)$$

where $r^{(n)}(x, y, t)$ denotes the sum of pixels in the rows of $\mathbf{Q}^{(n)}(x, y, t)$, $c^{(n)}(x, y, t)$ denotes the sum of pixels in the columns of $r^{(n)}(x, y, t)$, and $\mathbf{I}^{(n)}(x, y, t)$ denotes the sum through the temporal dimension of $c^{(n)}(x, y, t)$. The calculation of the

histogram of a block at bin n can be obtained using only 7 add operations:

$$\begin{aligned} \mathbf{B}^{(n)} = & \{\mathbf{I}^{(n)}(p_8) - \mathbf{I}^{(n)}(p_7) - \mathbf{I}^{(n)}(p_6) + \mathbf{I}^{(n)}(p_5)\} \\ & - \{\mathbf{I}^{(n)}(p_4) - \mathbf{I}^{(n)}(p_3) - \mathbf{I}^{(n)}(p_2) + \mathbf{I}^{(n)}(p_1)\} \end{aligned} \quad (5.15)$$

the label of the locations p_1, \dots, p_8 is given in Figure 5.3. The integral video volume is computed once for each video, and the histogram of each block is computed with $7M$ add operations.

Note the histogram technique renders invariants to small translation and rotations. We intentionally do not rotate the cuboid itself to retain the direction of the movements so that we can distinguish between actions such as *stand up* and *sit down*. The local feature captures characteristic shapes and motion, thus it provides robust representation of events that is invariant to spatial and temporal shifts, scales, background clutter, partial occlusions, and multiple motions in the scene.

5.1.3 Action Description

5.1.3.1 Cuboid Codebook

Inspired by the successful bag-of-words approach at RGB image classification and retrieval, we build a cuboid codebook by clustering the DCSF using K-means algorithm with Euclidean distance. The spatio-temporal codewords are defined by the center of the clusters and each feature vector can be assigned to a codeword using Euclidean distance or rejected as an outlier. Thus, each depth sequence can be represented as a bag-of-codewords from the codebook. These bag-of-codewords describe what’s happening in the depth

sequences in a simple yet powerful way. To incorporate the positional information of the cuboid, we concatenate the spatio-temporal information x, y, z, t with the DCSF feature before clustering. This gives small improvements under our experimental settings. Dimension reduction methods such as PCA can be incorporated before clustering without sacrificing the performance when choosing a suitable number of dimensions while making the clustering process much faster. We use a histogram of the cuboid prototypes as the action descriptor and SVM [13] for classification with histogram intersection kernel:

$$\mathbf{K}(a, b) = \sum_{i=1}^n \min(a_i, b_i), a_i \geq 0, b_i \geq 0 \quad (5.16)$$

5.1.3.2 Mining Discriminative Feature Pool

Not all the cuboid prototypes give the same level of discrimination among different actions, some cuboids may be related with movements that do not offer good discrimination among different actions, e.g. the sway of the body. To select the discriminative feature set from the pool, we use F-score. In a binary class case, given training vectors $x_k, k = 1, \dots, m$, if the number of positive and negative instances are n_+ and n_- respectively, the F-score of the i -th feature $F(i)$ is defined as:

$$\frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1} \sum_{k=1}^{n_+} (x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n_- - 1} \sum_{k=1}^{n_-} (x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2} \quad (5.17)$$

where $\bar{x}_i, \bar{x}_i^{(+)}, \bar{x}_i^{(-)}$ are the average of the i -th feature of the whole, positive, and negative data. $x_{k,i}^{(+)}$ is the i -th feature of the k -th positive instance, and $x_{k,i}^{(-)}$ is the i -th feature of the k -th negative instance. The F-score indicates

the discrimination between the positive and negative sets. We rank the cuboid prototypes by their F-scores and select features with high F-scores. The threshold is manually selected to cut between low and high F-scores. The number of features to keep generally depends on how good the STIPs are. In our experiments, small improvement is observed by deleting 1-2% cuboid prototypes. We also tested the well-known TF-IDF weighting or stop-words, it turns out it does not give noticeable improvement in our experiments.

5.2 Experimental Results

We test our algorithm on two public datasets: MSRAction3D dataset [48] and MSRDailyActivity3D dataset [90], and our own dataset. We compare our algorithm with state-of-the-art methods on activity recognition algorithms from depth videos [48, 88, 90, 91, 97, 100]. Experimental results show that our algorithm gives significantly better recognition accuracy than algorithm based on low-level features and gives even better results than algorithm using high-level joint features. We also give detailed comparisons on other choices of detectors or features and evaluation of parameters on our model. We take support region size $L = 6$ in all experiments.

5.2.1 MSRAction3D Dataset

The MSRAction3D dataset [48] mainly collects gaming actions. The depth image is clean, there are no background objects, and the subjects appear at the same depth to the camera. On this dataset, we take $\sigma = 5, \tau =$

Method	Accuracy
High Dimensional Convolutional Network [91]	72.5%
Bag of 3D points [48]	74.7%
HOJ3D feature [97]	79.0%
STOP [88]	84.8%
Eigenjoints [100]	82.3%
Random Occupancy Pattern [91]	86.50%
Actionlet [90]	88.2%
Ours	89.3%

Table 5.1: Comparison of accuracy on MSRAction3D dataset.

$T/27, T/17$ (T denotes the duration of the action sequence) and $N_p = 160$ for DSTIP extraction, and take the number of voxels for each cuboid to be $n_{xy} = 4, n_t = 2$. We fix the cuboid spatial size $\Delta_x = \Delta_y = 6\sigma$ because all actions take place at the same depth.

Table 5.1 shows the comparison of our algorithm with state-of-the-art algorithms on the MSRAction3D dataset. All algorithms are tested on the 20 actions, and we select half of the subjects as training and the rest as testing. Our algorithm outperforms the algorithms based on 3D silhouette features [48], skeletal joint features [90, 100] and local occupancy patterns [88, 91].

5.2.2 MSRDailyActivity3D Dataset

The MSRDailyActivity3D dataset collects daily activities in a more realistic setting, there are background objects and persons appear at different distances to the camera. Most action types involve human-object interaction. In our testing, we removed the sequences in which the subject is almost still (This may happen in action type: sit still, read books, write on paper, use

laptop and play guitar). Note that Li et al.’s algorithm [48] cannot work without segmenting out the human subjects from the depth image, which is not a trivial work considering the human appears at different depths and interacts with objects. Such dependence on important preprocessing largely limits the application of this algorithm. Here, we compare to Wang et al. [90] and other choices of STIP detectors and features, and we show the evaluation of parameters on this dataset.

Table 5.2 shows the accuracy of different features and methods. We take $\sigma = 5, 10, \tau = T/17, N_p = 500$ for DSTIP extraction and take the number of voxels for each cuboid to be $n_{xy} = 4, n_t = 3$. Wang et al.’s low-level feature LOP only achieves 42.5% while our DCSF feature achieves 83.6%, which is also better than Wang’s high-level joint position feature. When concatenate our DCSF feature with joint position feature, it presents an accuracy of 88.2% which is higher than LOP combined with Joint position feature reported in [90] 85.75%.

We also compared our DCSF descriptor with widely used descriptors in RGB images: Cuboid descriptor and HOG descriptor. To control the variables, we use the same set of DSTIP locations detected by our DSTIP detector at $\sigma = 5, \tau = T/17$ for all the descriptors and perform no feature selection. For the Cuboid descriptor, we use a fixed cuboid size $\Delta_x = \Delta_y = 6\sigma$, because it does not handle different sizes. For the HOG descriptor, we incorporate the adaptable cuboid size and take $n_{xy} = 6, n_t = 4$ and use 4-bin histograms of gradient orientations, which is the best parameter for HOG on this dataset.

Method	Accuracy
LOP feature [90]	42.5%
Joint position feature [90]	68.0%
DSTIP(ours)+Cuboid descriptor [22] (on depth)	73.6%
DSTIP(ours)+HOG [45] (on depth)	79.1%
Cuboid detector + Cuboid descriptor [22] (on RGB)	77.3%
DSTIP(ours)+DCSF(Ours) (on depth)	83.6%
LOP+Joint [90]	85.75%
DCSF+Joint(Ours)	88.2%

Table 5.2: Comparison of recognition accuracy on MSRDailyActivity3D dataset.

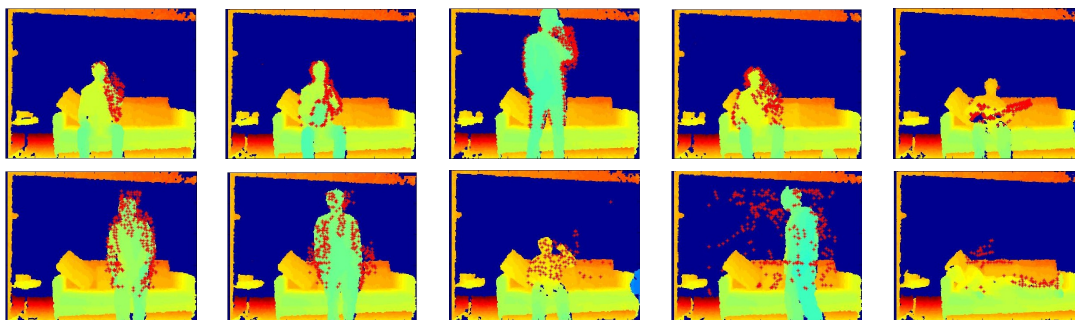


Figure 5.4: Example of STIPs extracted using our algorithm. They are projected onto x - y dimensions with one depth frame from the video for display. Action type from left to right, up to down: *drink-sit*, *eat*, *drink-stand*, *call cellphone*, *play guitar*, *sit down*, *stand up*, *toss*, *walk and lay-down*

Our DCSF descriptor performs significantly better than the Cuboid descriptor or gradient based descriptor even with adaptable cuboid size.

Figure 5.4 shows some examples of extracted DSTIPs on the MSRDailyActivity dataset using our detector. We also compared our DSTIP detector with widely used detectors in RGB images, including the Harris3D detector [44] and Cuboid detector [22]. We implemented the Cuboid detector and keep the same setting of spatial and temporal scale with our DSTIP detector. Figure 5.5 shows the STIPs extracted by the Cuboid detector and

our DSTIP detector when take the STIPs at local maximum with the top 50,100,200,300,500,800 response values. As we can see, the Cuboid detector first captures the noise in the background, then gradually begins to capture a few points around the moving arm at $N_p = 200$, but those informative points are overwhelmed by the large number of noisy points. This also suggests that the noise is at a larger magnitude than the real movements. Our DSTIP detector effectively captures the movement of the arm, and noisy points begin to appear as late as $N_p = 800$, but the majority of the STIPs still gather around the person.

For the Harris3D detector, we use the code on-line¹ and use the standard parameters: number of spatial pyramid equals 3 combined with $\sigma^2 = 4, 8$, $\tau^2 = 2, 4$, $k = 0.0005$. For the tool to work, we smooth and scale the depth pixels to 0-255. Figure 5.6 shows the STIPs extracted. Only a small fraction of STIPs locates around the moving body parts, most of them appear near edges or static objects. We tried varying the parameters but it gives similar results.

Figure 5.7 shows the influence of parameters on the average accuracy of our algorithm. The parameter tested are No. of STIPs per video N_p , No. of bins for the depth histogram M , No. of voxels for a cuboid n_{xy}, n_t , support region L , and codebook size k .

¹<http://www.di.ens.fr/laptev/interestpoints.html>

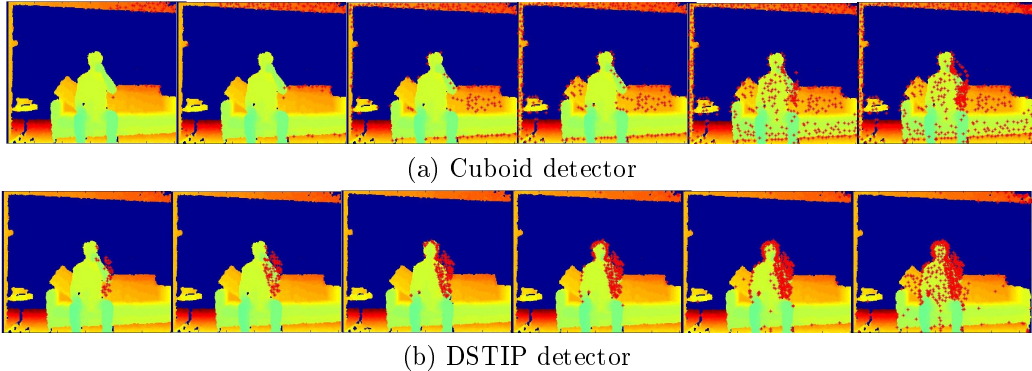


Figure 5.5: Comparison of our DSTIP detector with Cuboid detector. Example video is action drink. Column from left to right is taken $N_p = 50, 100, 200, 300, 500, 800$ respectively.

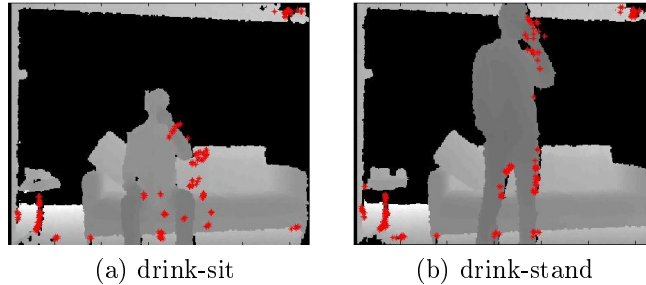


Figure 5.6: STIPs extracted using Harris3D detector [44]

5.2.3 UTKinect Dataset

Our dataset contains 10 actions: *hello*, *push*, *pull*, *boxing*, *step*, *forward-kick*, *side-kick*, *wave hands*, *bend*, and *clap hands*. These actions cover the movements of hands, arms, legs, and upper torso. Each action was collected from 10 different persons each performing the actions 3 times. The resolution of the depth map is 320×240 . Each action sample spans about 8 – 46 frames. We take $\sigma = 5, 10$, $\tau = T/8, T/5, T/3$ when filtering and take the number of voxels for each cuboid to be $n_{xy} = 4, n_t = 2$.

There is no skeleton information recorded so skeleton feature based

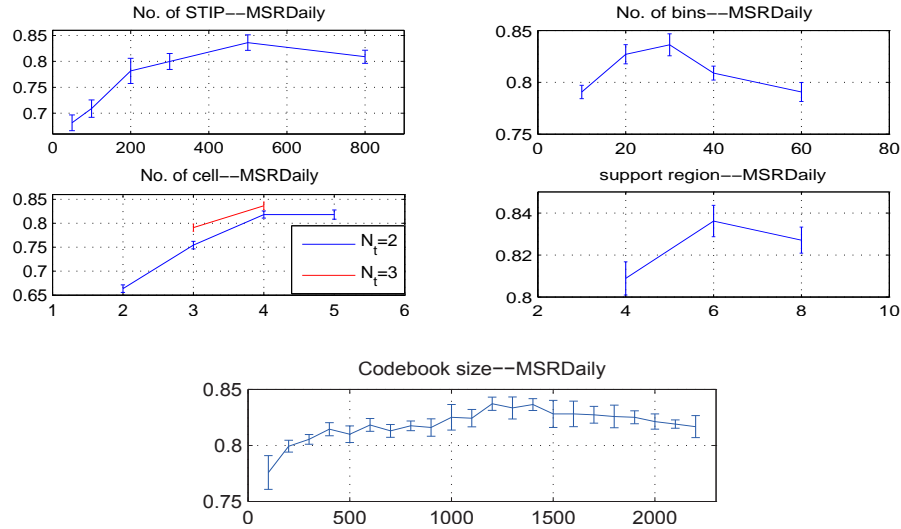


Figure 5.7: Parameter evaluation around optimum value on the MSRDailyActivity3D dataset. The average accuracy with the standard deviation denoted by error bar is plotted.

algorithms [90,100] cannot be applied onto it. On this dataset, we tried another method in which we take the 3D point clouds of the whole body in each frame and map it to a posture word. Then each action is represented by a sequence of posture words and we classify upon that (we refer to it as the "posture word method"). Table 5.3 gives the results of the two algorithms on three testing cases. The proposed DSTIP+DCSF pipeline performs significantly better than posture words method in that it focuses on the location of movement instead of trying to model the whole body, and the DSTIP pipeline automatically finds the movements without requiring segmentation of the human body as the posture word method does.

Notice from the experiments that our algorithm does not depend on

	Test One	Test Two	Cross Subject
DSTIP+DCSF	93.5%	96.7%	85.8%
Posture Word	83.89%	75.65%	79.57%

Table 5.3: Comparison of recognition rate on our own dataset. In **test one**, 1/3 of the samples were used as training samples and the rest as testing samples; in **test two**, 2/3 samples were used as training samples; In **cross subject test**, half of the subjects were used as training and the rest as testing.

the availability of skeleton information or preprocessing as other methods do. By this means, our algorithm is a more general approach to processing depth videos and recognizing activities, which may also be used for a wider variety of settings, e.g. group activities, local body parts activities, or non-human behavior studies.

5.3 Conclusion

This chapter presents algorithms to extract DSTIPs from depth videos and calculate descriptors for the local 3D depth cuboid around the DSTIPs. The descriptor may be used to recognize activities with no dependence on skeleton information or preprocessing such as human detection, motion segmentation, tracking, or image denoising or hole-filling. It is more flexible than existing algorithms. It has been applied on three different datasets and presents better recognition accuracy than other state-of-the art algorithms based on either low-level features or high-level features.

As shown in the experiment, there is rich possibility for extensions. When skeletal joint information is available, the DCSF feature can be com-

bined with the joint features to bring more accurate recognition results. Or, joint locations can be regarded as a type of interest points and cuboids can be extracted from those locations. On the other hand, when the corresponding RGB video is available, the DCSF features can be easily combined with STIP features from RGB videos to integrate information from two sources. Additionally, the STIP locations extracted from the depth videos and RGB videos can be combined or filtered to provide more stable and discriminate interest point locations and render better recognition performance.

Chapter 6

First-Person Activity Recognition Using RGBD Data

In this chapter, I describe the my work on first-person activities recognition using multi-modal data. The goal is to analyze the reactions and interactions of persons with a moving robot (the explorer) that wears a RGBD camera. This allows understanding whether the persons surrounding the explorer are friendly or hostile, and whether there will be a threat. We recorded two multi-modal first-person interaction datasets using a humanoid and non-humanoid robots bundled with Kinect. Multiple *2D* and *3D* descriptors are investigated and evaluated on our datasets; it is demonstrated that *3D* information renders significant improvement to this recognition task. Furthermore, the videos contain a high percentage of ego-motion due to the robot self-exploration as well as its reactions to the persons' interactions. It is shown that separating the descriptors extracted from ego-motion and independent motion areas, and using them both, allows us to achieve superior results. Experiments show that the proposed algorithm recognizes the activities effectively and outperforms other state-of-the-art methods.

6.1 Algorithm

6.1.1 Feature Extraction

It is widely known from the neuroscience literature that the body structure is learned in the early stages of human development [58] and that adults have prior knowledge of body appearance. Johansson [37] has demonstrated that for humans the movement of the main body joints are sufficient to discriminate among different action patterns. Given this evidence, we can consider motion and body appearance to be suitable to classify activities. The Kinect device already provides skeleton joint positions and orientations. This data is not always accurate or available though, especially if the camera is moving or the person that is performing the activity is very close to the camera. Therefore, we cannot rely on the skeleton data only; we need to define additional features that represent motion or body appearance. In particular, we select four different descriptors that have shown to perform well in classic activity recognition tasks: 3D optical flow, spatio-temporal interesting points, depth spatio-temporal interesting points, and body posture descriptors. At the same time, their combination aims at reproducing a mechanism similar to that we humans experience when recognizing activities. Some examples of the mentioned features are depicted in Fig. 6.1.

6.1.1.1 Motion Descriptor

We use histograms of 3D optical flow as our motion descriptors. Specifically, each RGB frame is divided into $c \times c$ cells, in order to explicitly capture

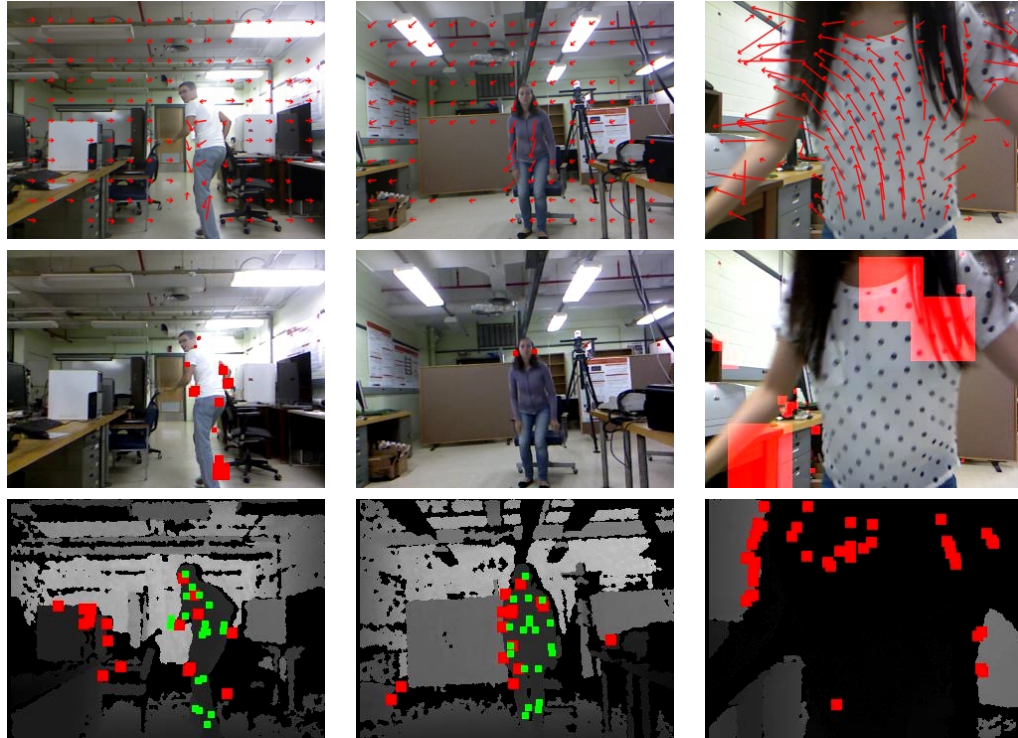


Figure 6.1: Sample frames from the dataset with the extracted features overlaid. The first row shows down-sampled dense optical flow features. The second row depicts the STIP cuboids. The third row illustrates the DSTIP cuboids (red), and skeletal joint locations (green) (skeleton feature for the activity in the third row is missing). The sample frames are extracted from run, stand up, and hug activities respectively.

local motion. For each frame F_t and its consecutive frame F_{t+1} , we compute the dense $2D$ optical flow [25]. It returns, for every pixel (x_t, y_t) in frame F_t , its velocity along the x and the y components, necessary to reach its new position (x_{t+1}, y_{t+1}) in frame F_{t+1} . To benefit from depth information, we proceed in projecting every pixel in frame F_t , in $3D$ [23]:

$$\begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} = \begin{pmatrix} \frac{(x_t - x_0)d(x_t, y_t)_t}{f} \\ \frac{(y_t - y_0)d(x_t, y_t)_t}{f} \\ d(x_t, y_t)_t \end{pmatrix}, \quad (6.1)$$

where (x_t, y_t) is the pixel in $2D$ at time t , (X_t, Y_t, Z_t) is the pixel in $3D$ at time t , $d(x_t, y_t)_t$ is the depth of pixel (x_t, y_t) obtained from the depth image, f is the focal length, and $(x_0, y_0)^T$ is the principal point of the sensor. We then project in $3D$ all the pixels in F_{t+1} , obtaining for each pixel the optical flow vector projected in $3D$ as $(X_{t+1}, Y_{t+1}, Z_{t+1})^T - (X_t, Y_t, Z_t)^T$. At this point, each $3D$ vector so computed is converted in spherical coordinates $(r, \theta, \phi)^T$. We drop the norm r , and we model each vector as its direction $\mathbf{v} = (\theta, \phi)^T$; this way, the descriptor will be invariant to the speed of the action (represented by r). At this stage, we have retrieved a set of vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^2$. We now group these vectors with respect to the specific cell from which they were extracted. For each cell, we compress the vector directions into a $2D$ histogram. After the whole procedure, we provide for each frame a histogram $\mathbf{h} \in \mathbb{R}^{c \times c \times b \times b}$, where b is the number of bins that represent the possible angle directions, and $c \times c$ is the number of the cells in the current frame.

In order to compact motion information over the entire video, we use a Vector Quantization (VQ) approach: all the possible motion descriptor vectors are clustered into k groups via a simple k-means procedure, and the centroids of the k groups represent the atoms of a codebook. Then, each frame is coded as an occurrence of a visual word, hence we obtain a new descriptor $\mathbf{z} \in \mathbb{R}^k$: its components are all 0 with the exception of one 1, in the position correspondent to the selected visual word. We finally sum all the frame descriptors, obtaining a histogram that contains, for each atom, the number of its occurrences in the video.

6.1.1.2 Local Appearance Descriptors

Sparse spatio-temporal features are employed to describe the local appearance of the videos. This representation has been found to be suitable for activity recognition tasks, as it handles cluttered backgrounds and partial occlusions in both RGB and depth videos [8, 45, 98]. Specifically, each video is represented as a 3D X-Y-T volume by concatenating the 2D image frames along the temporal axis T . For RGB videos, we use the gray-scale intensity value of the RGB channels, so each pixel $p(x, y, t)$ inside the 3D volume $I(x, y, t)$ corresponds to the intensity value of the pixel (x, y) at time t . For depth videos, each value $q(x, y, t)$ of the 3D volume $D(x, y, t)$ corresponds to the depth value of the pixel (x, y) at time t . We use Harris3D to detect the sparse spatio-temporal interesting points (STIPs), and HOGHOF to consequently describe the 3D cuboids extracted from the intensity videos [45]. In a

similar manner, we use the method in [98] to detect the local spatio-temporal interest points (DSTIPs) from the depth volume, and use the depth cuboid similarity feature (DCSF) to describe the cuboids extracted from the depth videos.

As for the optical flow descriptors, we build codebooks to obtain a single descriptor for each video. In particular, we build two separate codebooks for intensity and depth features. We consequently generate two bag-of-words histograms for each video. Since the bag-of-words model omits spatio-temporal information, we concatenate scaled spatial and temporal data $(\alpha_1x, \alpha_2y, \alpha_3t)$ to each feature vector before the clustering stage, i.e. $\bar{\mathbf{F}} = [\mathbf{F}, \alpha_1x, \alpha_2y, \alpha_3t]$, where (x, y) is the position of the pixel and t is the time instant. This expedient produces noticeable improvements.

6.1.1.3 Human Posture Descriptor

We use the skeletal joints information estimated from the depth images as a compact representation of the human posture [76]. This type of information has been employed in many frameworks [90] and gives promising results. Unlike the traditional third-person view settings, the skeleton in our first-person scenario is often missing or subject to noise and errors, especially when the person is too close or too far from the camera. Nevertheless, the absence of human detection, or the skeleton confused position, can still be indicative for our task. For example, if a skeleton is not detected, usually the person is very close to the camera, i.e. he is performing activities such as hug or punch, or

very far from it, i.e. he is running away. To handle noise and also incorporate the information from the missing and corrupted skeleton data, we employ a skeletal joint voting scheme. In particular, we use the *hip center* joint as the origin O of a $3D$ reference system, and we transform the other joint positions in spherical coordinates (r, θ, ϕ) with respect to O . Finally, we calculate an equally spaced $2D$ histogram on θ and ϕ , and we compress them into a single histogram. We construct a posture codebook using all the "good" postures from the training set, collecting a bag-of-words histogram for each video. In the meantime, we keep track of the number of frames that contain corrupted skeleton data, and devote a bin h' of the final histogram to this information $H = [h_1, h_2, \dots, h_n, h']$.

6.1.2 Separating Ego-Motion from Independent-Motion

Ego-motion can be defined as the camera motion; in our context, the ego-motion is mainly due to the robot's autonomous movements as well as the consequences of the performer's interactions with the robot – e.g. a *punch* action may drive the robot in a different position. On the contrary, the real motion happening in the scene is defined as **independent motion**: the person that moves to punch the robot is an example of independent motion. In nature, mammals' high-resolution fovea is usually driven towards objects that move with independent motion [85]; this mechanism allows them to process images fast, and improve their recognition capabilities. We aim at reproducing the same behavior, building an attention mask around the movements interpreted

as independent motion, and considering the rest of the scene as subject to ego-motion. We then extract motion and appearance descriptors from the two areas separately, assuming that ego-motion and independent motion regions give different contributions to the recognition procedure. In the following, we propose a simple motion-based segmentation algorithm to separate independent motion areas from ego-motion ones. As opposed to the other works in the literature, which tend to suppress background regions, we demonstrate that using both ego-motion and independent motion areas to recognize first-person interaction activities is crucial to improve the overall accuracy. Our independent-motion separation method does not rely on person/body part detector, therefore it is more flexible and particularly suitable for our task, where the person may be very close or seriously occluded.

6.1.2.1 Independent Motion Vectors

We can assume that the largest part of the independent motion is generated by the person that is performing the activity. A person detector thus, could implicitly catch the likely independent motion regions in a dataset where the person stands at a reasonable distance from the camera. A first-person activity dataset though, contains many videos where the person is extremely close, or very far from the camera, therefore some body parts such as the head are not visible. In these specific situations, person detectors are not always reliable. Therefore, instead of segmenting the person, we propose a new algorithm that explicitly seeks for independent motion regions. Specifically, we rely on

the fact that ego-motion typically induces only coherent motion on the image plane, whereas independent motion is usually very different. We use sparse optical flow, which focuses only on motion vectors between pixels that are easily detected in the image. For each frame, we compute the Lucas-Kanade sparse optical flow [51]; we then build a Multivariate Gaussian model on the pixel velocities so obtained, retrieving a couple of variables (μ, Σ) that represent respectively the mean and the covariance matrix. At this point we compute, for each pixel \mathbf{p}_i in the frame, the Mahalanobis distance between its velocity \mathbf{v}_i and the Gaussian model previously estimated: $D_M = \sqrt{(\mathbf{v}_i - \mu)^T \Sigma^{-1} (\mathbf{v}_i - \mu)}$. If $D_M > \epsilon$, where ϵ is a parameter experimentally chosen, the vector is considered independent motion, otherwise it is considered ego-motion. In Fig. 6.2 second row, the independent motion is represented, whereas in the first row the original sparse optical flow is depicted.

6.1.2.2 Attention Mask

The proposed procedure may suffer from outliers, such as motion vectors detected due to sudden changes of lighting. In order to avoid such false detections, we process the motion vectors to obtain a reliable attention mask. We first use k-means to cluster the motion vectors with respect to their depth; the maximum density cluster is selected, and it represents our focus of attention along the depth component. Since the independent motion is generated by a person, we measured the proportions of humans with respect to the depth to build a spatial x-y window around the focus of attention. In particular, given a

certain depth value d , we compute a windows height h and width w as follows: $h = 5.2 \times 10^5/d, w = 3h/4$. The windows so generated are represented in Fig. 6.2, third row. We finally perform an *AND* operation between the depth value retrieved by the focus of attention and the spatial window, obtaining the final attention mask (Fig. 6.2 fourth row). Notably, our algorithm is able to handle situations where the person is very close or very far from the camera.

The attention mask is able to capture precisely the contour of the human body, therefore it is particularly useful for separating the optical flow generated by the person from the optical flow caused by the camera. Given a set of $3D$ motion vector directions expressed in spherical coordinates $\mathbf{v}_1, \dots, \mathbf{v}_n$, as explained in Sec. 6.1.1.1, we split them into two groups: the first group \mathbf{I} represents the optical flow related to pixels belonging to the attention mask, whereas the second group \mathbf{E} represents the ego-motion optical flow vectors. At this point, for both sets we apply the rest of the procedure explained in Sec. 6.1.1.1, computing two codes z_I and z_E for each frame. We sum the codes of all the frames belonging to a video, obtaining two histograms h_I and h_E that count the visual words occurring in the video. We finally concatenate the two descriptors.

Differently, for the appearance descriptor extraction, a wider mask is needed. Our appearance descriptor is extracted by analyzing the video for a certain duration of time, thus pixels very close to the person are actually "affected" by the person's movement and, in this context, they can be considered as independent motion. To separate the independent pixels from the

others, we directly use the attention window (Fig. 6.2, third row). As for the optical flow descriptors, we split the interesting points into two groups: the first group contains descriptors extracted from the window and the second one contains the remaining pixels. We obtain a histogram of each group applying the procedure explained in Sec. 6.1.1.2, and we concatenate them into the final descriptor of a video.

6.1.3 Multiple Channel Kernels

So far we have generated a Bag-of-Words histogram for each type of feature. We would like the classifier (e.g. SVM) to integrate all the descriptors in an effective way, being able, in specific situations, to privilege a descriptor with respect to another one. A promising approach aiming to assign different weights to different typologies of features is the multi-channel kernel. We define the multi-channel kernels that integrate the aforementioned features as follow:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\sum_{m=1}^M d_m K_m(\mathbf{x}, \mathbf{x}')\right) \quad (6.2)$$

$$\sum_{m=1}^M d_m = 1, d_m \geq 0, \forall m$$

where each basis kernel K_m uses a subset of variables stemming from different data sources, and M is the total number of kernels. We select the following basis kernel:

$$K_m(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^n \frac{2x_i x'_i}{x_i + x'_i}, \quad (6.3)$$

We are now looking for a decision function in the form $f(x) + b = \sum_{m=1}^M f_m(x) + b$, where each function f_m is associated with a kernel K_m . We

use Support Vector Machines (SVM) as our classifiers. Both the coefficients of the problem and the weights d_m can be solved via a single optimization problem:

$$\begin{aligned}
 & \min_{\{f\}, b, \epsilon, d} \quad \frac{1}{d_m} \|f_m\|_H^2 + C \sum_i \epsilon_i \\
 \text{s.t.} \quad & y_i \sum_m f_m x_i + y_i b \geq 1 - \epsilon_i, \forall i \\
 & \epsilon_i \geq 0, \forall i \\
 & \sum_m d_m = 1, d_m \geq 0, \forall m
 \end{aligned} \tag{6.4}$$

Equation 6.4 can be transformed into a constrained optimization problem, and solved by a simple gradient method [68].

6.2 Experiments

There are several public datasets on RGBD human activity recognition [60, 82, 90]. These datasets though, focus on 3rd-person recognition and the observer/camera itself is not involved in interactions. In our setting, the camera shows a salient amount of ego-motion due to interactions, unlike any previous RGBD datasets. To the best of our knowledge, there does not exist any RGBD dataset with any human-camera physical interactions, and it is not meaningful to test our algorithm on a dataset that does not have camera motion. For this reason, we propose two new datasets for 3D first-person activity recognition. We evaluate other state-of-the-art algorithms on the two datasets noticing that they do not perform well; this confirms that our scenario is different from the traditional 3rd-person view problem and classic methods are not suitable for the 1st-person tasks.

Here we first describe the datasets used in this study, then provide

detailed experimental results. In particular, we carried out three different experimental sessions:

- In the first experiment, we investigate the influence of $2D$ and $3D$ features on our recognition task. We show that adjoining $3D$ information to the RGB stream significantly improves the classification results.
- In the second experiment, we show that descriptors extracted from regions that move of independent motion and regions that move of ego-motion provide different contributions in the recognition of activities. In particular, explicitly separating the two components and using both of them enables the achievement of significantly higher accuracy.
- In the third experiment, we present the results combining different types of descriptors. We finally compare our results to [70] and some other approaches that show to perform very well on classic activity recognition tasks.

6.2.1 Dataset

We collect two benchmark datasets for first-person human interaction activity recognition. We record data from a Kinect device mounted on top of the first-person. We use a humanoid first-person (a Teddy Panda bundled on a wheelchair) and a non-humanoid autonomous first-person. They are both able to move and rotate horizontally, but the ego-motion appears in different patterns. The humanoid first-person has head and arm and it is able

to react to a variety of motion patterns derived from the interaction, i.e. it may shake when a person shakes hands with it, and it may fall down when a person punches it. The non-humanoid first-person more closely resembles a battle-field robot and it has more steady self-motions. Sample images of the 9 activities taken from the humanoid first-person dataset are shown in figure 6.1. The 9 classes in the non-humanoid first-person dataset are: *ignore*, *pass by the first-person*, *point at the first-person*, *reach an object*, *run away*, *stand up*, *stop the first-person*, *throw at the first-person*, and *wave to the first-person*.

For each dataset, we invited 8 subjects, between the ages of 20 to 80, to perform a variety of reactions and interactions with our explorer. We ask each subject to perform 7 – 9 different continuous sequences of activities, in a few different background settings. Each group of activities performed by one subject forms a set. Some examples of sequences may be:

- *wave hands* → *approach* → *shake hands* → *hug*
- *stand up* → *reach an object* → *throw [something] at the explorer*
- *approach* → *pass by the explorer*

The continuous sequences are then segmented so that each video represents a single activity. Each set contains around 20 – 35 samples of the 9 activities, with at least one sample for each activity. In total, we collect 8 sets and 177 single activity samples for the humanoid first-person dataset, and 8 sets and 189 single activity samples for the second dataset.

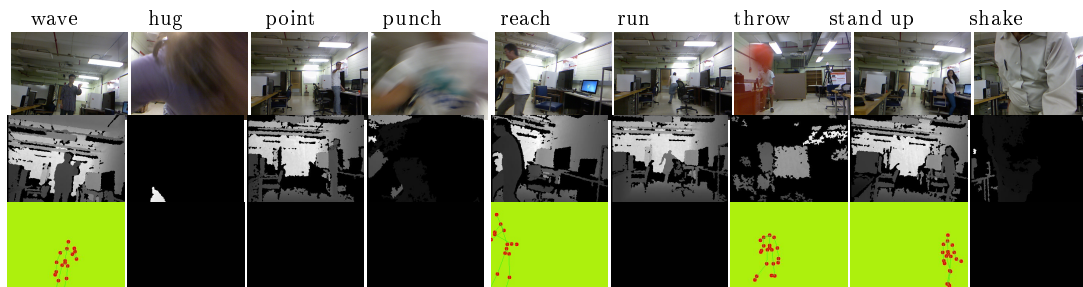


Table 6.1: Sample images of 9 activities in our humanoid first-person dataset. The first row presents RGB images. The second row shows depth images. The last row represents skeleton images. If no skeleton is detected for a particular frame, a black image is shown.

We record the RGB video, the depth video, and the 3D skeleton joint locations from the Kinect simultaneously; the frame rate is about 30fps. The depth image is a 16-bit single channel image of resolution 320×240 . We use the full range of the Kinect (0.8m to 8m) to record more information. The RGB image is an 8-bit 3 channel lossy compressed image of resolution 640×480 .

For all the experiments, we used a cross-subject test: in particular, we use subject No. 1-4 as training and No. 5-8 as testing. In order to also account for randomness due to the clustering of the codebooks, we assessed our algorithm over 10 – 20 different codebook trials. We report, for each experiment, the mean accuracy and the maximum accuracy over all the trials. The first two experiments are conducted on our humanoid first-person dataset. Finally, we evaluate our complete features on both the described datasets.

Feature	Mean Accuracy	Max Accuracy
2D HOF	45.92%	56.79%
3D HOF	55.22%	61.72%

Table 6.2: The table illustrates the comparison between $2D$ and $3D$ optical flow descriptors.

6.2.2 2D vs 3D

In this first experiment, we show that depth and $3D$ cues are fundamental to obtain superior results on our first-person activity dataset.

6.2.2.1 Optical Flow

In this section, $2D$ and $3D$ optical flow performance is compared. Both the descriptors are extracted without using the attention mask. For the $2D$ optical flow, we use the equivalent of our $3D$ descriptor: we compute the dense optical flow motion vectors, then model their directions as the arctangent of their velocities, and divide the frame in $c \times c$ cells. We compute the histogram of the directions of the flow vectors extracted from every cell, and we concatenate them. Finally, we use the Bag of Words technique to retrieve a single descriptor for the entire video. We used 8 main motion directions (i.e. number of bins) and 9 cells for the $3D$ optical flow, obtaining a frame descriptor $\mathbf{h} \in \mathbb{R}^{576}$. For the $2D$ version, we tested histograms with 8, 16 and 64 bins, and we divide each frame in 9 cells. We finally build, for both the $3D$ and $2D$ optical flow, 10 different codebooks of visual words. In Table 6.2, the final comparison between $2D$ and $3D$ optical flow descriptors is presented. We only show the best $2D$ results, which are achieved using 16 bins. Notably, the ac-

curacy of the $3D$ optical flow is always higher in comparison to the $2D$ motion descriptors. These results show that $3D$ information allows obtaining a boost in the recognition rate.

6.2.2.2 STIP

In this section, we compare the performance of the spatio-temporal features from $2D$ intensity image sequences (STIP+HOGHOF) [45] to the spatio-temporal features from depth image sequences (DSTIP+DCSF) [98]. The descriptors evaluated here are computed without applying our attention mask. Some activity segments are very short and may not contain salient intensity changes, resulting in 0 STIPs for several sequences; we use a 0-histogram to represent this case. For the sake of completeness, we report the performance including those 0-word videos (96 training, 81 testing, indicated in Table 6.3 as "all"), and without including them (93 training, 76 testing, labeled as "non-empty"). In this experiment, DSTIP features perform significantly better than STIPs, while usually, when there is little camera motion, the performance of the two descriptors are similar. A possible reason may be that the lighting changes due to the camera motion has a great impact on the STIPs, whereas DSTIPs are very robust against it. When combining the STIPs with DSTIPs though (see Table 6.3), we achieve better results than single STIP or DSTIP features. This experiment demonstrates that adding the depth dimension in first-person activity tasks significantly improves the performance.

Feature(s)		Mean Accuracy	Max Accuracy
STIP	all	48.46%	50.62%
	non-empty	57.63%	61.84%
DSTIP		72.83%	79.01%
STIP	single kernel	74.07%	76.54%
+DSTIP	multi-kernel	77.47%	80.25%

Table 6.3: Comparison of results for spatio-temporal features.

Feature(s)	No Mask		Ind-motion		Ego-motion		Ind+ego	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max
3D optical Flow	55.22%	61.72%	54.07%	60.49%	46.34%	56.79%	59.25%	69.13%
STIP	57.63%	61.84%	57.63%	60.53%	28.55%	32.89%	62.63%	69.74%
DSTIP	72.83%	79.01%	74.69%	77.78%	39.14%	45.68%	75.93%	80.24%

Table 6.4: This table illustrates the comparison between raw descriptors and features extracted using the attention mask.

6.2.3 Mask

In this section, we apply our attention mask to the optical flow and appearance features, and compare the performance (table 6.4). In order to show that the cause of the improvement is the combination of the two components, we also show the accuracy achieved by the descriptors extracted from ego-motion and independent motion regions singularly (table 6.4, second and third column). It is worth noticing that single descriptors extracted from ego-motion or independent motion do not necessarily obtain superior results. On the contrary, when we use the attention mask to combine the contribution of ego-motion and independent motion regions, we obtain higher accuracy on both motion and appearance descriptors (table 6.4 last column).

6.2.4 Single Features vs Concatenated

In this section, we present the results using multiple combinations of features to give an insight on the contributions and characteristics of each descriptor (Table 6.5). Results are reported using both the described datasets.

Due to the different characteristics of the two datasets, single features perform differently. For instance, in the humanoid first-person dataset, spatio-temporal appearance features give better results over motion or posture descriptors. Differently, in the non-humanoid first-person dataset, the subjects are usually at a further distance from the explorer; in this case, the quality of the depth images deteriorate, and all our depth-based features experience a decrease in the results. The quality of the skeleton data instead, improves with respect to the previous dataset, where the person is very close to the explorer. Therefore, we obtain superior results using posture features. Finally, the performance of the combined descriptors using the multiple-channel kernels give similar results on the two datasets. This indicates that the proposed work may constitute a stable framework for first-person activity recognition.

We also compared our results with the algorithm developed by Ryoo et al. [70], which, to the best of our knowledge, is the only work on first-person human interaction activity recognition. Table 6.5 also summarizes the accuracy achieved by other methods [45, 62, 98] that demonstrated to be suitable for general activity recognition tasks. It is possible to notice that the best results are obtained using the combination of our features.

Figure 6.3 presents the confusion matrix of the 4 features and the combination of them using our multi-channel kernels on our humanoid first-person dataset. The 4 features offer different discerning abilities for the activity classes. For example, optical flow and STIPs show a very low accuracy in the recognition of the "wave" activity, whereas the posture descriptor achieves high results. This behavior, noticed for several activities, explains the great improvement gained by combining optical flow with posture features.

6.3 Conclusion

This Chapter presents a framework and algorithm to recognize first-person activities using multi-modal data. The proposed algorithm helps the robot to gain consciousness of the surrounding environment, to be aware of the intention of the persons around it, and to take action in case of a threat. This kind of frameworks can also be embedded into wearable cognitive assistant systems to give instructions or warning the person.

I propose and make publicly available two new first-person activity datasets, which incorporate RGBD and skeleton data. This additional information allows us to extract *3D* cues, which meaningfully increases the classification rate. I investigate several intensity, depth and skeleton features, evaluating their contributions and their combinations on our new task. I also separate the regions of ego-motion and independent motion and utilize them both. I demonstrate that descriptors extracted from the foreground and background give different contributions to the recognition, and their combination

Feature(s)	Humanoid robot		Non-humanoid robot	
	Mean	Max	Mean	Max
Ryoo et al. [70]	57.1%	64.19%	58.48%	63.33%
Laptev et al. [45]	48.46%	50.62%	50.83%	57.14%
Xia et al. [98]	72.83%	79.01%	53.25%	57.14%
Liu al. [62]	52.54%	52.54%	45.55%	45.55%
OF	59.25%	69.13%	52.07%	57.77%
ST	76.85%	80.25%	64.38%	67.78%
P (all)	56.79%	60.49%	70.0%	75.56%
P (non-empty)	62.66%	70.31%	75.74%	81.82%
OF+P	78.60%	80.25%	80.41%	84.44%
OF+ST	77.98%	81.48%	65.54%	68.89%
ST+P	83.88 %	85.60%	80.94%	84.44%
OF+ST+P	85.60%	86.42%	83.70%	87.78%

Table 6.5: All the comparisons are illustrated. The first rows are dedicated to the results obtained using state-of-the-art methods. Following that are the performance of the descriptors we have investigated. The results of the proposed features applying the attention mask are shown starting from the fifth row: 3D optical flow features (OF), combination of depth and intensity spatio-temporal features (ST), posture descriptor (P), and different combinations. The last row indicates our best results, attained using the combination of all the features together.

notably improves the results. The presented methodology has never been utilized in the literature to the best of my knowledge.

The contribution of this work is threefold. Firstly, we propose and make publicly available the first datasets for first-person interaction activity recognition that provide RGB, depth and skeleton data. Secondly, we show that additional 3D information is fundamental to achieving improved performance. Thirdly, we propose a new concept: ego-motion and independent motion regions are both important to improve the recognition results when ego-motion is present.

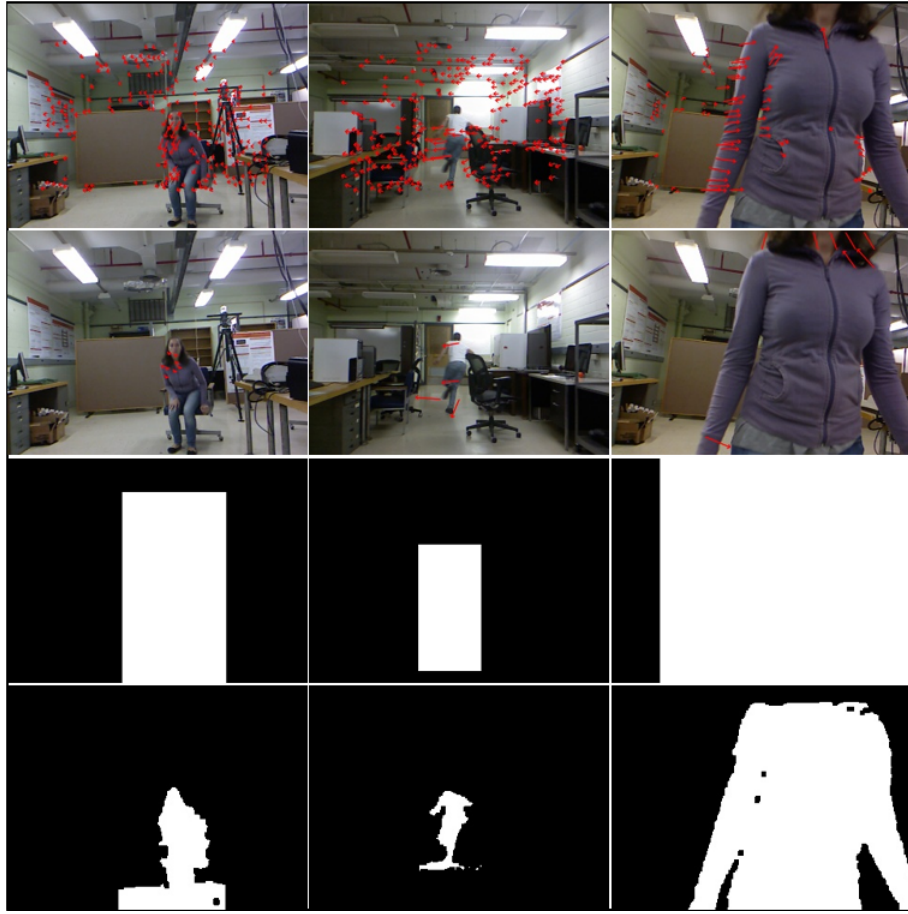


Figure 6.2: In the first row, the sparse optical flow is depicted. The second row shows the vectors identified as independent motion. The third row presents the attention window used for the STIP features. In the fourth row, the attention mask used for the optical flow features is represented.

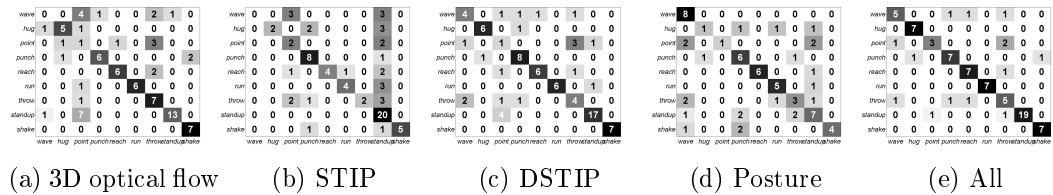


Figure 6.3: Confusion matrices of the four different features on the humanoid first-person dataset.

Chapter 7

Conclusion

My thesis presented algorithms to find persons in the scene and recognize the actions and activities of the persons using RGBD imagery. Furthermore, I proposed to study a novel problem of recognizing human interactions from a first-person perspective using RGBD data and gave a robust solution.

I first presented a model-based algorithm to localize the persons in an indoor scene. Unlike the previous works, I did not assume certain poses or motion patterns of the person. I proposed a 3D shape model to find the humans in the depth image, it is more robust to lighting changes and background clutter. It may work alone on a depth image, or it can be combined with detection algorithms from RGB imagery to compliment the drawbacks in both data sources. Instead of running the detection window at multiple scales to find the body part as is usually done in the RGB images, I took advantage of the depth information to estimate the scale, and adjust the model to the correct scale for detection. The detection process involves two models: a 2D edge template, and a 3D shape model. The 2D edge template summarizes the contour shape of the head and upper shoulder part, which is a relatively stable shape from either a front, back, or side view. Then, a 3D model is

fitted onto the regions returned by the 2D template fitting process to verify if the object is of a spherical shape. An occlusion mask is extracted before fitting the 3D model to remove the objects that occlude or occluded by the head. After localizing the head, a region growing algorithm is employed to find the whole body of the person and a contour is extracted. Furthermore, a tracking algorithm is proposed based on the detection result. This algorithm was tested on two datasets and outperformed state-of-the-art algorithms on RGB imagery and depth imagery.

Furthermore, I proposed a view-invariant posture feature from the human skeletons extracted from a depth video. I constructed a reference coordinate in the 3D skeleton space, which rotates according to the direction of the person and makes the representation view-invariant. The polar angle and azimuth angle of a joint are computed on this reference coordinates and casted into 30 degrees bins, the votes are then concatenated into a feature vector called HOJ3D. This feature is a good abstraction of the posture of the person and the computation is real-time. It offers a simple and effective feature for the real-time systems to recognize human actions.

Since skeleton is not always available for real applications, I designed a more general feature for activity recognition. It describes local contents of a depth video using spatio-temporal concepts. The image frames in a video are concatenated along the time dimension, interesting locations are extracted from this chunk of data, and local voxels around these interesting locations are extracted and described using the proposed feature. Considering the charac-

teristics of depth video, I designed a filtering approach with a noise suppressor to localize the points around human motions. A new descriptor was proposed to describe the local 3D voxels around the interest points with a self-similarity notion that offers flexibility and also handles the noise. This feature may be directly extracted from the depth video without the need for human detection, skeleton estimation, background subtraction, or motion tracking. It is widely applicable for a variety of scenarios.

Finally, I proposed a novel problem of recognizing human interaction-level activities from a first-person perspective combining RGB and depth data. This problem is novel in the sense that the activities are recorded from the perspective of one of the persons involved in the interaction, while traditional computer vision algorithms recognize activities from a 3rd-view camera irrelevant to the activity. First-person activity recognition is a challenging problem due to the presence of a significant amount of ego-motion in the video. Research on this topic came out very recently and previous researchers addressed this problem using only a RGB camera. I proposed to use RGBD videos to solve this problem and gave a more robust solution. The independent-motion and ego-motion regions in the video are separated with the help of the depth channel. Then, features are extracted from the two different regions for analyzing the ongoing activity. 3D optical flows are computed to estimate the motion of the scene/person, spatio-temporal features are extracted from RGB channels and depth channel to describe the local contents, and skeletal features are built to provide information about the posture of the person. I compared

the proposed algorithm with state-of-the-art approaches on activity recognition from 3rd-person view or 1st-person view. Results showed that adding the depth channel significantly improved the performance.

In summary, the main impact of my thesis is that I developed several robust features on the depth imagery for activity recognition and addressed the novel problem of first-person activity recognition using RGBD data. I made publicly available several RGBD datasets. This thesis showed that depth information is very useful for activity recognition tasks in computer vision.

At the same time with the rapid development of the depth sensor, a growing number of research projects are being conducted using RGBD data. I believe this is just the beginning of the low-cost high-end range sensors and the related research. With future developments, range sensors will have a higher resolution, less noise, and an extended sensing range. Furthermore, depth sensors accompanied by traditional cameras on laptops and cell phones are coming out in the near future, which will provide broader computer vision research topics and applications. I believe that my thesis will contribute to many of the real-world applications and also open the doors to more interesting problems related to RGBD vision and activity recognition.

Bibliography

- [1] JK Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.
- [2] JK Aggarwal and Lu Xia. Human activity recognition from 3d data: A review. *Pattern Recognition Letters Special Issue*, 2014.
- [3] Max Bajracharya, Baback Moghaddam, Andrew Howard, Shane Brennan, and L Matthies. Results from a real-time stereo-based pedestrian detection system on a moving vehicle. In *Workshop on People Detection and Tracking, IEEE ICRA*, 2009.
- [4] Gioia Ballin, Matteo Munaro, and Emanuele Menegatti. Human action recognition from rgb-d frames based on real-time 3d optical flow estimation. In *Biologically Inspired Cognitive Architectures 2012*, pages 65–74. Springer, 2013.
- [5] John L Barron, David J Fleet, and SS Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.
- [6] Ardhendu Behera, David C Hogg, and Anthony G Cohn. Egocentric activity monitoring and recovery. In *ACCV*, pages 519–532. 2013.
- [7] Nicola Bellotto and Huosheng Hu. Multisensor-based human detection and tracking for mobile service robots. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):167–181, 2009.
- [8] S. Belongie, K. Branson, P. Dollar, and V. Rabaud. Monitoring animal behavior in the smart vivarium. *Measuring Behavior*, 2005.
- [9] Jezekiel Ben-Arie, Zhiqian Wang, Purvin Pandit, and Shyamsundar Rajaram. Human activity recognition using multidimensional indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1091–1104, 2002.

- [10] Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou. G3d: A gaming action dataset and real time action recognition evaluation framework. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 7–12. IEEE, 2012.
- [11] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *PAMI*, 23(3):257–267, 2001.
- [12] Jan Cech, Jordi Sanchez-Riera, and Radu Horaud. Scene flow estimation by growing correspondence seeds. In *CVPR*, pages 3129–3136. IEEE, 2011.
- [13] C.C. Chang and C.J. Lin. Libsvm: a library for support vector machines. *TIST*, 2(3):27, 2011.
- [14] Rizwan Chaudhry, Avinash Ravichandran, Gregory Hager, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1932–1939. IEEE, 2009.
- [15] Zhongwei Cheng, Lei Qin, Yituo Ye, Qingming Huang, and Qi Tian. Human daily action analysis with multi-view and color-depth data. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 52–61. Springer, 2012.
- [16] Wongun Choi, Caroline Pantofaru, and Silvio Savarese. Detecting and tracking people using an rgb-d camera via multiple detector fusion. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1076–1083. IEEE, 2011.
- [17] Jinshi Cui, Hongbin Zha, Huijing Zhao, and Ryosuke Shibasaki. Tracking multiple people using laser and vision. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2116–2121. IEEE, 2005.
- [18] Peng Dai, Huijun Di, Ligeng Dong, Linmi Tao, and Guangyou Xu. Group interaction analysis in dynamic context. *SMC*, pages 275–282, 2008.

- [19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [20] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. *Computer Vision–ECCV 2006*, pages 428–441, 2006.
- [21] James W Davis and Aaron F Bobick. The representation and recognition of human movement using temporal templates. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 928–934. IEEE, 1997.
- [22] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *PETS*, pages 65–72, 2005.
- [23] S. R. Fanello, I. Gori, G. Metta, and F. Odone. Keep it simple and sparse: Real-time action recognition. *JMLR*, 2013.
- [24] Sean Ryan Fanello, Ilaria Gori, Giorgio Metta, and Francesca Odone. Keep it simple and sparse: Real-time action recognition. *Journal of Machine Learning Research*, 14:2617–2640, 2013.
- [25] Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. *Lecture Notes in Computer Science*, pages 363–370, 2003.
- [26] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In *ICCV*, pages 407–414, 2011.
- [27] Ajo Fod, Andrew Howard, and MAJ Mataric. A laser-based people tracker. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 3024–3029. IEEE, 2002.
- [28] Hironobu Fujiyoshi and Alan J Lipton. Real-time human motion analysis by image skeletonization. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 15–21. IEEE, 1998.

- [29] James J Gibson. The perception of the visual world. 1950.
- [30] Ismail Haritaoglu, David Harwood, and Larry S Davis. Hydra: Multiple people detection and tracking using silhouettes. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 280–285. IEEE, 1999.
- [31] Michael Boelstoft Holte, Thomas B Moeslund, and Preben Fihl. View-invariant gesture recognition using 3d optical flow and harmonic motion context. *Computer Vision and Image Understanding*, 114(12):1353–1361, 2010.
- [32] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1):185–203, 1981.
- [33] Frederik Huguet and Frédéric Devernay. A variational method for scene flow estimation from stereo sequences. In *ICCV*, pages 1–7. IEEE, 2007.
- [34] Sho Ikemura and Hironobu Fujiyoshi. Real-time human detection using relational depth similarity features. *Computer Vision-ACCV 2010*, pages 25–38, 2011.
- [35] Sumer Jabri, Zoran Duric, Harry Wechsler, and Azriel Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 627–630. IEEE, 2000.
- [36] A. Jalal, MD. Zia Uddin, Jeong Tai Kim, and T. S. Kim. Recognition of human home activities via depth silhouettes and R transformation for smart homes: Indoor and built environment. pages 1–7, 2011.
- [37] Gunnar Johansson. Visual motion perception. *Scientific American*, 1975.
- [38] S.-W. Joo and R. Chellappa. Attribute grammar-based event recognition and anomaly detection. In *CVPRW*, pages 107–107, 2006.
- [39] Takeo Kanade and Martial Hebert. First-person vision. *Proceedings of the IEEE*, 100(8):2442–2453, 2012.

- [40] Kris Kitani. Ego-action analysis for first-person sports videos. *Pervasive Computing*, 11(2):92–95, 2012.
- [41] A. Klaser and M. Marszalek. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.
- [42] A Kurakin, Z Zhang, and Z Liu. A real time system for dynamic hand gesture recognition with a depth sensor. 2012.
- [43] Tian Lan, Leonid Sigal, and Greg Mori. Social roles in hierarchical models for human activity recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1354–1361. IEEE, 2012.
- [44] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.
- [45] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.
- [46] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, pages 1346–1353, 2012.
- [47] Antoine Letouzey, Benjamin Petit, Edmond Boyer, and Morpheo Team. Scene flow from depth and color images. In *Jesse Hoey, Stephen McKenna and Emanuele Trucco, Proceedings of the British Machine Vision Conference*, pages 46–1, 2011.
- [48] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. *CVPR Workshop*, 1(5):9–14, 2010.
- [49] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–14. IEEE, 2010.

- [50] Zhe Lin, Larry S Davis, David Doermann, and Daniel DeMenthon. Hierarchical part-template matching for human detection and segmentation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.
- [51] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Image Understanding Workshop*, 1981.
- [52] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.
- [53] Syed Z Masood, Chris Ellis, Adarsh Nagaraja, Marshall F Tappen, JJ LaViola, and Rahul Sukthankar. Measuring and reducing observational latency when recognizing actions. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 422–429. IEEE, 2011.
- [54] J. Maunsell and D. V. Essen. The connections of the middle temporal visual area in the macaque monkey and their relationship to a hierarchy of cortical areas. *Journal of Neuroscience*, 1983.
- [55] Tomas McCandless and Kristen Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. 2013.
- [56] Krystian Mikolajczyk, Cordelia Schmid, and Andrew Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Computer Vision-ECCV 2004*, pages 69–82. Springer, 2004.
- [57] Amar Mitiche and Jagdishkumar Keshoram Aggarwal. *Computer Vision Analysis of Image Motion by Variational Methods*. Springer, 2013.
- [58] D. L. Mumme. Early social cognition: understanding others in the first months of life. *Infant and Child Development*, 2001.
- [59] Luis E Navarro-Serment, Christoph Mertz, and Martial Hebert. Pedestrian detection and tracking using three-dimensional ladar data. *The International Journal of Robotics Research*, 29(12):1516–1528, 2010.

- [60] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *ICCV Workshop*, pages 1147–1153, 2011.
- [61] Antonios Oikonomopoulos, Ioannis Patras, and Maja Pantic. Spatiotemporal salient points for visual recognition of human actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(3):710–719, 2005.
- [62] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. *CVPR*, 2013.
- [63] Constantine Papageorgiou and Tomaso Poggio. Trainable pedestrian detection. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 4, pages 35–39. IEEE, 1999.
- [64] Alonso Patron-Perez, Marcin Marszalek, Ian Reid, and Andrew Zisserman. Structured learning of human interactions in tv shows. 2012.
- [65] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, pages 2847–2854, 2012.
- [66] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *CVPR*, volume 1, pages 829–836, 2005.
- [67] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [68] Alain Rakotomamonjy, Francis Bach, Stéphane Canu, Yves Grandvalet, et al. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.
- [69] Juan-Alberto Rivera-Bautista, Antonio Marin-Hernandez, Ana-Cristina Ramirez-Hernandez, and Luis F Marin-Urias. Detection of 3d human torsos using multiple data for human-robot interaction. 2011.
- [70] Michael S Ryoo and Larry Matthies. First-person activity recognition: What are they doing to me? *CVPR*, 2013.

- [71] Joaquín Salas and Carlo Tomasi. People detection using color and depth images. In *Pattern Recognition*, pages 127–135. Springer, 2011.
- [72] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S Davis. Human detection using partial least squares analysis. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 24–31. IEEE, 2009.
- [73] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360. ACM, 2007.
- [74] S Sempena, NU Maulidevi, and PR Aryan. Human action recognition using dynamic time warping. In *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, pages 1–5. IEEE, 2011.
- [75] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and Andrew Blake. Real-time human pose recognition in parts from a single depth image. *CVPR*, 2011.
- [76] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.
- [77] Luciano Spinello and Kai O Arras. People detection in rgb-d data. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3838–3843. IEEE, 2011.
- [78] Luciano Spinello and Roland Siegwart. Human detection using multimodal and multidimensional features. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3264–3269. IEEE, 2008.
- [79] Ekaterina H Spriggs, Fernando De La Torre, and Martial Hebert. Temporal segmentation and activity classification from first-person sensing. In *CVPRW*, pages 17–24, 2009.

- [80] Thad Starner. Project glass: An extension of the self. *Pervasive Computing, IEEE*, 12(2):14–16, 2013.
- [81] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Human activity detection from RGBD images. *PAIR*, 2011.
- [82] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 842–849. IEEE, 2012.
- [83] Agnes Swadzba, Niklas Beuter, Joachim Schmidt, and Gerhard Sagerer. Tracking objects in 6d for reconstructing static scenes. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008.
- [84] K. Tanaka. *High-level motion processing : computational, neurobiological, and psychophysical perspectives*, chapter Representation of visual motion in the extrastriate visual cortex. 1998.
- [85] V. J. Traver and A. Bernardino. A review of log-polar imaging for visual perception in robotics. In *RAS*, pages 378–398, 2010.
- [86] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.
- [87] Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 722–729. IEEE, 1999.
- [88] Antonio Vieira, Erickson Nascimento, Gabriel Oliveira, Zicheng Liu, and Mario Campos. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 252–259, 2012.

- [89] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, Cordelia Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, 2009.
- [90] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.
- [91] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu. Robust 3d action recognition with random occupancy patterns. In *ECCV*, pages 872–885. 2012.
- [92] LM Wang and Z Zhuang. [measurement and analysis of human head-face dimensions]. *Zhonghua lao dong wei sheng zhi ye bing za zhi= Zhonghua laodong weisheng zhiyebing zazhi= Chinese journal of industrial hygiene and occupational diseases*, 26(5):266–270, 2008.
- [93] Andreas Wedel, Thomas Brox, Tobi Vaudrey, Clemens Rabe, Uwe Franke, and Daniel Cremers. Stereoscopic scene flow computation for 3d motion understanding. *IJCV*, 95(1):29–51, 2011.
- [94] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. *Computer Vision–ECCV 2008*, pages 650–663, 2008.
- [95] Bo Wu and Ramakant Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 90–97. IEEE, 2005.
- [96] Di Wu, Fan Zhu, and Ling Shao. One shot learning gesture recognition from rgb-d images. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 7–12. IEEE, 2012.
- [97] L. Xia, C.C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPRW*, 2012.

- [98] Lu Xia and JK Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *CVPR*, 2013.
- [99] Fengliang Xu and Kikuo Fujimura. Human detection using depth and gray images. In *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pages 115–121. IEEE, 2003.
- [100] Xiaodong Yang and YingLi Tian. Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor. In *CVPRW*, pages 14–19, 2012.
- [101] Xiaodong Yang, Chenyang Zhang, and YingLi Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. 2012.
- [102] Angela Yao, Juergen Gall, and Luc Van Gool. A hough transform-based voting framework for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2061–2068. IEEE, 2010.
- [103] Elden Yu and JK Aggarwal. Human action recognition with extremities as semantic posture representation. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2009.
- [104] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 28–35. IEEE, 2012.
- [105] Vladimir M Zatsiorsky. *Kinematics of human motion*. Human Kinetics 1, 1998.
- [106] Chenyang Zhang and Yingli Tian. Rgb-d camera-based daily living activity recognition.
- [107] H. Zhang and L.E. Parker. 4-dimensional local spatio-temporal features for human activity recognition. In *IROS*, pages 2044–2049, 2011.

- [108] Yang Zhao, Zicheng Liu, Lu Yang, and Hong Cheng. Combing rgb and depth map features for human activity recognition. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–4. IEEE, 2012.
- [109] Yu Zhu, Wenbin Chen, and Guodong Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 486–491. IEEE, 2013.