

Copyright

by

Lu Xia

2012

**The Thesis Committee for Lu Xia
Certifies that this is the approved version of the following thesis:**

**Human Detection and Action Recognition Using Depth Information by
Kinect**

**APPROVED BY
SUPERVISING COMMITTEE:**

Supervisor:

J.K. Aggarwal

Kristen Grauman

**Human Detection and Action Recognition Using Depth Information by
Kinect**

by

Lu Xia, B.E.

Thesis

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

Master of Science in Engineering

The University of Texas at Austin

May 2012

Dedicated to my family

Acknowledgements

First of all, I would like to express my sincere appreciation to my advisor, Dr. J.K. Aggarwal, for giving me the chance to work in this interesting area. I thank him for his continuous support, guidance and his trust in me. His wisdom of foreseeing promising topics has offered me great opportunities to produce novel research.

I also offer my deep thank to Dr. Kristen Grauman for introducing me to the world of computer vision. I learnt from her not only the foundation knowledge and state-of-the-art research, but also her dedication and persistence toward research.

I would like to thank my fellow graduate students, especially Chia-Chih Chen, Josh Harguess, Birgi Tamersoy, Suyog Jain, Jong Taek Lee and Dr. Changbo Hu. They offered me lots of help and shared valuable experiences with me.

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

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

Abstract

Human Detection and Action Recognition Using Depth Information by Kinect

Lu Xia, M.S.E.

The University of Texas at Austin, 2012

Supervisor: J. K. Aggarwal

Traditional computer vision algorithms depend on information taken by visible-light cameras. But there are inherent limitations of this data source, e.g. they are sensitive to illumination changes, occlusions and background clutter. Range sensors give us 3D structural information of the scene and it's robust to the change of color and illumination. In this thesis, we present a series of approaches which are developed using the depth information by Kinect to address the issues regarding human detection and action recognition.

Taking the depth information, the basic problem we consider is to detect humans in the scene. We propose a model based approach, which is comprised of a 2D head contour detector and a 3D head surface detector. We propose a segmentation scheme to segment the human from the surroundings based on the detection point and extract the whole body of the subject. We also explore the tracking algorithm based on our detection result. The methods are tested on a dataset we collected and present superior results over the existing algorithms.

With the detection result, we further studied on recognizing their actions. We present a novel approach for human action recognition with histograms of 3D joint locations (HOJ3D) as a compact representation of postures. We extract the 3D skeletal joint locations from Kinect depth maps using Shotton et al.'s method. The HOJ3D computed from the action depth sequences are reprojected using LDA and then clustered into k 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). In addition, due to the design of our spherical coordinate system and the robust 3D skeleton estimation from Kinect, our method demonstrates significant view invariance on our 3D action dataset. Our dataset is composed of 200 3D sequences of 10 indoor activities performed by 10 individuals in varied views. Our method is real-time and achieves superior results on the challenging 3D action dataset. We also tested our algorithm on the MSR Action3D dataset and our algorithm outperforms existing algorithm on most of the cases.

Table of Contents

List of Tables	x
List of Figures	xi
Chapter 1: Introduction	1
1.1 Motivation.....	1
1.2 Human Detection	4
1.3 Action Recognition	5
1.4 Contributions.....	6
Chapter 2: Related Work	8
2.1 Human Detection and Pose Estimation From 3D Data	8
2.2 Representation and Recognition of Human Actions.....	8
Chapter 3: Human Detection	12
3.1 Overview of the Method	12
3.2 2D Chamfer Distance Matching	13
3.2.1 Preprocessing.....	13
3.2.2 2D Chamfer Distance Matching	13
3.3 3D Model Fitting.....	15
3.3.1 Compute Head Parameters.....	15
3.3.2 Generate 3D Model.....	17
3.3.3 Fitting.....	17
3.4 Extract Contours	18
3.5 Tracking	21
3.6 Experimental Results	22
3.6.1 Dataset.....	22
3.6.2 Experimental Results	23
3.7 Conclusions.....	29
Chapter 4: View Invariant Action Recognition Using HOJ3D	30
4.1 Overview of The Method.....	30

4.2	Body Part Inference and Joint Position Estimation	31
4.3	HOJ3D as Posture Representation	32
4.3.1	Spherical Coordinates of Histogram	33
4.3.2	Probabilistic Voting	34
4.3.3	Feature Extraction	36
4.4	Vector Quantization	36
4.5	Action Recognition Using Discrete HMM	37
4.6	Experiments	38
4.6.1	Data	38
4.6.2	Experimental Results	41
4.7	Conclusions	44
	Chapter 5: Conclusion & Future Work	45
	Bibliography	46
	Vita	52

List of Tables

Table 3.1:	Region growth algorithm	21
Table 3.2:	Accuracy of our human detection algorithm.	25
Table 3.3:	Comparison of our algorithm with Ikemura	26
Table 4.1:	Different views of the actions.	40
Table 4.2:	The variations of subjects performing the same action.	40
Table 4.3:	The mean and standard deviation of the sequence lengths measured by number of frames at 30 fps.	41
Table 4.4:	Recognition rate of each action type.....	41
Table 4.5:	The three subsets of actions used in the experiments.	43

List of Figures

- Figure 1.1: Depth map and corresponding RGB image of an indoor scene. On the left is the depth image (brighter colors correspond to closer to the camera.) On the right is the RGB image.....3
- Figure 3.1: Over view of the human detection method12
- Figure 3.2: Intermediate results of 2D Chamfer Distance Matching. (a) shows the depth array after noise reduction. (b) gives the binary edge image calculated using Canny edge detector and then eliminate small edges. (c) shows the distance map generated from edge image (b). Match the binary head template (d) to (c) gives the head detection result (e). Yellow dots indicate the detected locations.....14
- Figure 3.3: Regression result of head height and depth.....15
- Figure 3.4: 3D head model. (a) illustrates the demands of the head model: the model should invariant to different views. (b) shows the hemisphere model we used as the 3D head model.....17
- Figure 3.5: (a) illustrates the process of estimating the true parameter of the head from the distance map. Input of the 3D model fitting is the output of the 2D Chamfer Distance Matching in Figure 3(e). Output of 3D model fitting is shown in (b). Yellow dots indicate the center of the head detected.18

Figure 3.6: (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 growth algorithm. The ground plane is delineated by the thresholded F filter response. The edges along the feet well separate the persons from the floor.....	19
Figure 3.7: (a) Result of our region growth algorithm. (b) The extracted whole body contours are superimposed on the depth map.....	20
Figure 3.8: (a) a patch of the depth array (b) the depth array showed using color map JET	23
Figure 3.9: Examples of the human detection result.....	23
Figure 3.10: Tracking result. 15 consecutive frames are shown.....	24
Figure 3.11: Examples of false negative detections. In group (a) the person that got occluded is not detected. In group (b) the person that is half way out of the frame is not detected.	24
Figure 4.1: Overview of the method.....	30
Figure 4.2: (a) Depth image. (b) Skeletal joints locations by Shotton et al.....	32
Figure 4.3: Reference coordinates of HOJ3D.....	34
Figure 4.4: Modified spherical coordinate system for joint location binning.....	34
Figure 4.5: Voting using a Gaussian weight function.....	36
Figure 4.6: Example of the HOJ3D of a posture.	36
Figure 4.7: Sample images from videos of the 10 activities in our database. Depth and RGB images are shown. Note only depth images are used in the 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>	39

Figure 4.8 Sample depth images from the MSR Action3D dataset. One frame for each of the 20 actions is shown. Action type (from left to right, up to bottom): *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, pickup & throw*.....43

Chapter 1: Introduction

1.1 MOTIVATION

The development of computer vision, the use of a camera and a computer to recognize objects, began in the early 1960s. Although it has matured fairly quickly and contributes to the solution of some of the most serious societal problems, most of the vision algorithms are built on 2D intensity images. But the world we live in is a 3D world. Upon seeing a 2D image, a human has no difficulty understanding its 3D structure. However, inferring such 3D structures remains extremely challenging for current computer vision systems. Such 3D geometric structure is necessary for most computer vision applications such as navigation and object search, and it may bring significant improvement to a lot of the current computer vision algorithms. But the acquiring of such 3D geometric structures is a difficult problem. There are basically two ways to obtain it. The first method is to estimate 3D structures from 2D images. But it remains an extremely challenging task for current computer vision systems, because there is a great deal of information lost when we project the 3D scene into a 2D image. Indeed, in a narrow mathematical sense, it is impossible to recover 3D depth from a single image, since we can never know if it is a picture of a painting or if it is an image of an actual 3D environment. Yet, in practice we consider it possible to estimate the 3D structure from the 2D image. Because humans can easily infer 3D structures from a 2D image, there must be some cues embedded in the 2D image that allow us to recover the 3D structure of the scene. Although much effort has been devoted to this issue, the result is still not as good as one would expect. The second method is to get the 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. For example, sonar sensors are relatively low cost, but have poor angular resolution and are susceptible to false echoes and reflections. Infrared and laser range finders are again relatively low cost, but typically provide measurements from only one point in the scene. At the other end of the spectrum, LIDAR and radar systems provide accurate distance measurements with good angular resolution across a plane in the scene, but are considerably more expensive and typically have higher power consumption requirements. The advent of low-cost digital cameras has therefore generated renewed interest in vision-based systems, but the disadvantage of this approach is that distance has to be inferred either from stereoscopic cameras, or from the motion of objects within the image (e.g. optical flow).

The research on range data dates back from 1980s, when people used laser range sensors which is expensive, slow at acquiring data and difficult to use on human subjects. Among the precursory attempts, Gil et al. [57] have done experiments at combining intensity and range edge maps. To describe 3D structures of objects, Magee et al. [58] have research on employing specialized range sensing hardware together with 2D intensity images of a scene to build descriptors of objects. Vemuri et al. [59] studied on calculating representation of objects from range data. To overcome the defect of the slow speed of laser range sensor at that time, Magee et al. [60] have explored in intensity guided range sensing recognition of three-dimensional objects. Furthermore, Vemuri et al. [61] have studied on obtaining representation of 3D objects from range data using intrinsic surface properties. Later on, Chu et al. [62] have studied on image interpretation using multiple sensing modalities.

While depth cameras are not conceptually new, the recent release of the Kinect has made such sensors accessible to all and received great deal of attention from the public. The Kinect provides both an RGB image and a depth image. Although intended primarily for the entertainment market, the Kinect has excited considerable interest within the vision and robotics community for its broad applications [4]. The Kinect camera uses a structured light technique [5] to generate real-time depth maps containing discrete range measurements of the physical scene. The quality of the depth sensing, given the low-cost and real-time nature of the device, is compelling, especially when compared with the previous commercial range sensors. However, it is still inherently noisy. Depth measurements often fluctuate and depth maps contain numerous ‘holes’ where no estimations of range are obtained. See Figure 1.1.

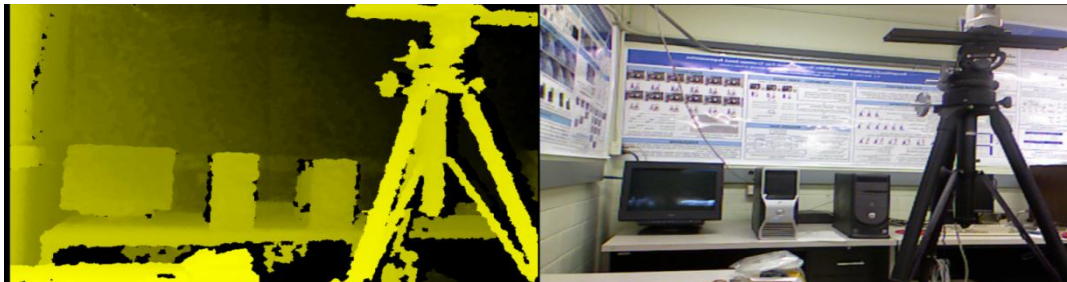


Figure 1.1: Depth map and corresponding RGB image of an indoor scene. On the left is the depth image (brighter colors correspond to closer to the camera.) On the right is the RGB image.

There is significant number of research studies using the Kinect during the last two years. Projects and applications can be found on facial expression analysis and synthesis [6], indoor navigation [7], gaming [8] and robotics [9, 10]. From last year, there is a body of research on the problems of human/parts detection [11], pose estimation [2],

tracking from 3D data [12] and activity recognition from depth images [3, 13] or combined with RGB images [14].

In this thesis, we consider using Kinect depth data to understand human activities. Depth images have several advantages over 2D intensity images: they are robust to the change in color and illumination. Also, depth images are simple representations of 3D information. To this end, our first task is to detect the humans in the scene. Then we track the persons and recognize their basic actions.

1.2 HUMAN DETECTION

Detecting human in images or videos is a challenging problem due to variations in pose, clothing, lighting conditions and complexity of the backgrounds. There has been much research in the past few years in human detection and various methods are proposed [15, 16, 17]. Most of the research is based on images taken by visible-light cameras, which is a natural way to do it just as what human eyes perform. Some methods involve statistical training based on local features, e.g. gradient-based features such as HOG [15], EOH [18], and some involve extracting interest points in the image, such as scale-invariant feature transform (SIFT) [19], etc. Although lots of reports showed that these methods can provide highly accurate human detection results, RGB image based methods encounter difficulties in perceiving the shapes of the human subjects with articulated poses or when the background is cluttered. These will result in the drop of accuracy or the increase of computational cost.

In this thesis, we present a novel model based method for human detection from depth images captured by Kinect. We detect people using a 2-stage detection process, which includes a 2D edge detector and a 3D shape detector to make use of both the edge

information and the relational depth change information embedded in the depth image. We propose a segmentation scheme to segment the human from the background and extract the overall contour of the subject accurately. We also present a simple algorithm for tracking in depth image. The method is evaluated on a 3D dataset taken indoor using the Kinect and achieves excellent results.

1.3 ACTION RECOGNITION

Human action recognition is a widely studied area in computer vision. Its applications include surveillance systems, video analysis, robotics and a variety of systems that involve interactions between persons and electronic devices such as human-computer interfaces. Its development began in the early 1980s. To date research has mainly focused on learning and recognizing actions from video sequences taken by a single visible light camera. There is extensive literature in action recognition in a number of fields, including computer vision, machine learning, pattern recognition, signal processing, etc. [28, 29]. Among the different types of features for recognition, silhouettes and spatio-temporal interest points are most commonly used [30].

Here we enumerate three major challenges to vision based human action recognition. First 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. Second, the number of describable action categories is huge; the same action may have different interpretations under different object and scene contexts. Third, occlusions, cluttered background, cast shadows, varying illumination conditions and viewpoint changes can all alter the way actions are perceived.

The use of range cameras significantly alleviates the challenges presented in the third category, which are the common low-level difficulties that reduce the recognition performance from 2D imagery. Furthermore, a range camera provides the discerning information of actions with depth changes in certain viewpoints. For example, in a frontal view, it would be much more accurate to distinguish person pointing from reaching from depth map sequences than in RGB footage.

In this thesis, we employ a histogram based representation of 3D human posture named HOJ3D. In this representation, 3D space is partitioned into n bins using a modified spherical coordinate system. We manually select 12 informative joints to build a compact representation of human posture. To make our 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 vocabularies. By encoding sequences of depth maps into sequential vocabularies, we recognize actions using HMM classifiers [39]. Our algorithm utilizes depth information only. Experiments show that this algorithm achieves superior results on our challenging dataset and also outperforms Li et al. algorithm [3] on nearly all the testing cases.

1.4 CONTRIBUTIONS

The main contribution of this thesis consists of three parts. First, we put forward a novel algorithm of human detection using depth information. Second, we present an algorithm to recognize human actions using skeletonization result inferred from depth imagery. Third, we propose a view-invariant representation of human poses and prove it is effective at action recognition and the whole system runs at real-time. Finally, we

collected a large 3D dataset of persons performing different kinds of indoor activities with a variety of viewpoints.

The remainder of the thesis is organized as follows: We discuss the related work in the Chapter 2. Chapter 3 describes our human detection algorithm using depth image and also presents our preliminary research on tracking. In Chapter 4 we present our view-invariant human action recognition algorithm. We conclude in Chapter 5 and discuss possible future works.

Chapter 2: Related Work

2.1 HUMAN DETECTION AND POSE ESTIMATION FROM 3D DATA

In recent years, there is a body of research on the problem of human parts detection, pose estimation and tracking from 3D data. Earlier research used stereo cameras to estimate human poses or perform human tracking [20, 21, 22]. In the past few years, a part of the research has focused on the use of time-of-flight range cameras (TOF). Many algorithms have been proposed to address the problem of pose estimation and motion capture from range images [23, 24, 25, 26]. Ganapathi et al. [23] present a filtering algorithm to track human poses using a stream of depth images captured by a TOF camera. Jain et al. [24] present a model based approach for estimating human poses by fusing depth and RGB color data. Recently, there have been several works on human/parts detection using TOF cameras. Plagemann et al. [27] use a novel interest point detector to solve the problem of detection and identifying body parts in depth images. Ikemura et al. [1] proposed a window-based human detection method using relational depth similarity features based on depth information. Recently, there has been research on human parts detection and pose estimation from depth images from Kinect. Shotton et al. [2] propose to extract 3D body joint locations from a depth image using an object recognition scheme.

2.2 REPRESENTATION AND RECOGNITION OF HUMAN ACTIONS

Researchers have explored different compact representations of human actions in the past few decades. Here we mainly divide them into 3 categories:

silhouette/contour/shape based representation, joint/body parts representation and space-time interest points representation.

Silhouette/contour/shape is an effective representation to describe the shape of the body postures. It is usually extracted from the video using background subtraction [63]. Methods proposed in the past for silhouette/contour/shape based action recognition can be divided into two major categories. One is to extract action descriptors from the sequences of silhouettes. Conventional classifiers are frequently used for recognition [31, 32, 33, 34]. In this approach, the action descriptors are supposed to capture both spatial and temporal characteristics of the actions. The other one is to extract features from each silhouette and model the dynamics of the action explicitly [33, 35, 36, 37, 38]. To extract features from the silhouettes/contours, some researchers extract feature directly from the whole silhouettes, e.g. use PCA to extract Eigen images [64, 65, 66]. Many researchers extract extremities of the silhouettes and link the extremities to the centroid to make a skeleton representation [42, 68, 69]. The “star skeleton” proposed by Fujiyoshi and Lipton [41] is widely used. The skeleton is generated by extracting the human silhouettes from the video using background segmentation and finding the gross extremities of the silhouette’s boundary, where extremities should correspond to the limbs and head. The “star skeleton” can also be extended to 3D using multiple cameras [67]. But note such skeleton is not what we mean by the human skeleton which is linking of the body joints. And it may not represent the pose well if not viewed from the favorable view.

Joints/body parts based has also been popular for a few decades. In 1975, Johansson’s experiment shows that humans can recognize activity with extremely compact observers [40]. Johansson demonstrated his statement using a movie of a person walking in a dark room with lights attached to the person’s major joints. Even though only light spots could be observed, there was a strong identification of the 3D motion in

these movies. There are plenty of works focusing on action recognition using joints/body parts [43, 74, 75]. Joints data are usually acquired using markers to the subjects and use multiple cameras to get the 3D positions [70, 71, 72, 73]. The CMU Motion Capture Database is widely used that provide such information. Such representation suffers little of the intra-class variance that plague appearance-based methods. Especially, 3D joint/body parts are view point invariant and appearance invariant, in that the actions vary little from different actors. But in the past, extracting body parts/3D joints accurately is a difficult task, particularly under realistic imaging conditions. As such, low-level appearance features such as spatio-temporal interest points have also been popular recently.

Inspired by natural language processing and information retrieval, space-temporal interest point approaches are also applied to recognize actions as a form of descriptive action unites. In these approaches, actions are represented as a collection of visual words, which is the codebook of spatio-temporal features. Schuldt et al. [44] integrate space-time interest point's representation with SVM classification scheme. Dollar et al. [45] employ histogram of video cuboids for action representation. Wang et al. [45] represent the frames using the motion descriptor computed from optical flow vectors and represent actions as a bag of coded frames.

However, all these features are computed from RGB images and are view dependent. Researchers also explored free viewpoint action recognition algorithms from RGB images. Due to the large variations in motion induced by camera perspective, it is extremely challenging to generalize them to other views even for very simple actions. One way to address the problem is to store templates from several canonical views and interpolate across the stored views [31, 47]. Scalability is a hard problem for this approach. Another way is to map an example from an arbitrary view to a stored model by

applying homography. The model is usually captured using multiple cameras [48]. Weinland et al. [49] model action as a sequence of exemplars which are represented in 3D as visual hulls that have been computed using a system of 5 calibrated cameras. Parameswaran et al. [50] define a view-invariant representation of actions based on the theory of 2D and 3D invariants. They assume that there exists at least one key pose in the sequence in which 5 points are aligned on a plane in the 3D world coordinates. Weinland et al. [51] extend the notion of motion-history [55, 31] to 3D. They combine views from multiple cameras to build a 3D binary occupancy volume. Motion history is computed over these 3D volumes and view-invariant features are extracted by computing circular FFT of the volume.

There are a few works on the recognition of human actions from depth data in the past two years. Li et al. [3] employ an action graph to model the dynamics of the actions and sample a bag of 3D points from the depth map to characterize a set of salient postures that correspond to the nodes in the action graph. However, the sampling scheme is view dependent. Lalal et al. [53] utilize Radon transformation on depth silhouettes to recognize human home activities. The depth images were captured by a ZCAM [54]. This method is also view dependent. Sung et al. [55] extract features from the skeleton data provided by Prime Sense from RGBD data from Kinect and use a supervised learning approach to infer activities from RGB and depth images from Kinect. Considering they extract features from both types of imageries, the result is interesting but at the same time not as good as one would expect.

Chapter 3: Human Detection

3.1 OVERVIEW OF THE METHOD

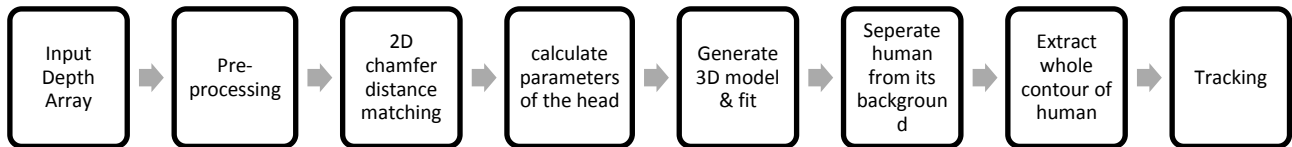


Figure 3.1: Over view of the human detection method

This section provides an overview of the major steps in our method, which is summarized in Figure 3.1.

Given an input depth array, we first reduce the noise and smooth the array for later process. We use a 2-stage head detection process to locate the people. We first explore the boundary information embedded in the depth array to locate the candidate regions. The algorithm used here is 2D Chamfer Distance Matching. It scans across the whole image and gives the possible locations that suggest the appearance of a human's head. We examine each of these regions using a 3D head model, which utilizes the relational depth information of the array for verification. We extract the parameters of the head from the depth array and use the parameter to build a 3D head model. Then we fit the 3D model against all the candidate regions to make a final estimation. We also develop a region growing algorithm to find the entire body of the person and extract the body contour. Also, we give preliminary research on tracking using our detection result.

3.2 2D CHAMFER DISTANCE MATCHING

3.2.1 Preprocessing

As we have mentioned, the quality of the Kinect depth sensing is still inherently noisy. Depth measurements often fluctuate and depth maps contain numerous ‘holes’ where no estimations of range are obtained. In the depth image taken by the Kinect, all the points that the sensor is not able to measure depth are offset to 0 in the output array. To estimate its true depth value, we make the assumption that the space is continuous, and the missing point is more likely to have a similar depth value to its neighbors. With this assumption, we use nearest neighbor interpolation algorithm to fill these pixels and get a depth array that has meaningful values in all the pixels. Then we use median filter with a 4×4 window on the depth array to smooth the data to make up the fluctuation of the depth sensing.

3.2.2 2D Chamfer Distance Matching

The first stage of the method is to use the edge information embedded in the depth array to locate the candidate regions that indicate the appearance of a person. It is a rough scanning approach in that we need to have a rough detection result with a false negative rate as low as possible but may have a high false positive rate. We use 2D Chamfer Distance Matching in this stage for quick processing. Chamfer Distance Matching is a good 2D shape matching algorithm that is invariant to scale. We use canny edge detector to extract edges in the depth array. To reduce the disturbance from the surrounding irregular small objects, we eliminate all the edges whose sizes are smaller than a certain threshold. (Here, the size of the edge is determined by the number of pixels.) Results of Chamfer Distance Matching are shown in Figure 3.2.

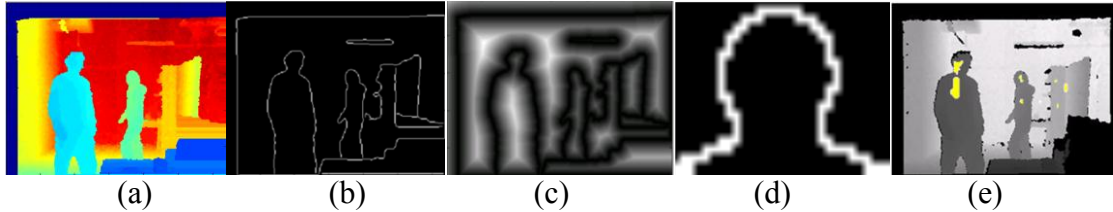


Figure 3.2: Intermediate results of 2D Chamfer Distance Matching. (a) shows the depth array after noise reduction. (b) gives the binary edge image calculated using Canny edge detector and then eliminate small edges. (c) shows the distance map generated from edge image (b). Match the binary head template (d) to (c) gives the head detection result (e). Yellow dots indicate the detected locations.

We use a binary head template shown in Figure 3.2(d) and match the template to the resulted edge image. To increase the efficiency, a distance transform is calculated before the matching process. This results in a distance map of the edge image, where pixels contain the distances to the closest data pixels in the edge image. Matching consists of translating and positioning the template at various locations of the distance map; the matching measure is determined by the pixel values of the distance image which lie under the data pixels of the transformed template. The lower these values are, the better the match between image and template at this location. If the distance value lies below a certain threshold, the target object is considered detected at this place, which means that a head like object is found here. We use the phrase “head like object” here because the object we detected may not be a real head because we used a high threshold here to guarantee a very low false negative rate. Whether this object is actually a head we will decide at the next stage. It is usually the case that the person in the scene is likely to appear at any depth, which means the head size will change according to the depth. To make the algorithm invariant to scale, we generate an image pyramid with the original image at the bottom; each image is subsampled to generate the next image at the higher level. The subsample rate we used here is $3/4$, and the number of the level of the pyramid

depends on the scene. If the scene contains a larger range of depth, a larger number of levels is needed. This template is able to find head of the person in all poses and views. If the person is in a horizontal position or is upside down, it is easily settled by rotating the template and running the same detection process. The result of all the steps in this stage is shown in Figure 3.2.

3.3 3D MODEL FITTING

Now we are going to examine all the candidate regions that selected by 2D Chamfer Distance Matching algorithm.

3.3.1 Compute Head Parameters

To generate the 3D head model, we first estimate the parameter of the head that appears in the detected location. We conduct an experiment and get the regression result for the depth of the head and its height, shown in Figure 3.3.

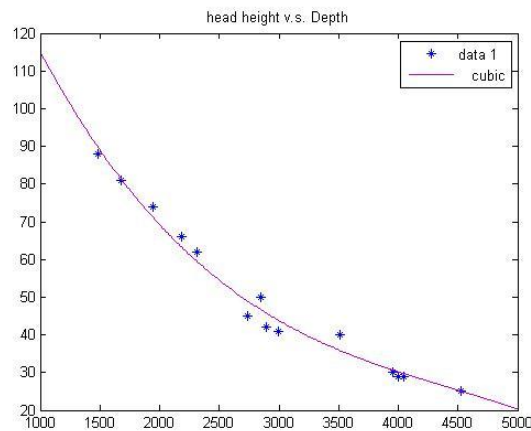


Figure 3.3: Regression result of head height and depth.

The cubic equation we get is:

$$-y = p_1 \cdot x^3 + p_2 \cdot x^2 + p_3 \cdot x + p_4 \quad (3.1)$$

Here,

$$\begin{cases} p_1 = -1.3835 \times 10^{-9} \\ p_2 = 1.8435 \times 10^{-5} \\ p_3 = -0.091403 \\ p_4 = 189.38 \end{cases}$$

From the detection result of 2D Chamfer Distance Matching, we can get the depth of detected location from the original depth array. By equation (3.1), we calculate the standard height of the head in this depth. Then we search for the head within a certain range that is defined by the standard height of the head:

$$R = 1.33h / 2. \quad (3.2)$$

Here, h is the height of the head calculated from equation (3.1), R is the search radius.

Next, we search for the head within a circular region defined by radius R in the edge image. If there is a circular edge in this region that satisfied all the constraints, e.g. size pass a certain threshold, it is decided that a head is detected. The next thing to do is to find the true radius of the head. It happened to be that the distance map we calculated at 2D Chamfer Distance Matching stage can be used to estimate the radius of the head. Recall that the pixels in the distance map contain the distances from this pixel to the closest data pixels in the edge image, considering the head is a circular like shape, the value of the center of the head on the distance map is just an approximation of the radius

of the head. So we can take this directly as our estimation of the true radius of the head R_t .

3.3.2 Generate 3D Model

Considering the calculation complexity of 3D model fitting is comparatively high, we want the model to be view invariant so that we don't have to use several different models or rotate the model and run several times. The model should generalize the characteristics of the head from all views: front, back, side and also higher and lower views when the sensor is placed higher or lower or when the person is higher or lower. To meet these constraints and make it the simplest, we use a hemisphere as the 3D head model. Figure 3.4 illustrates the requirements and shows the head model.

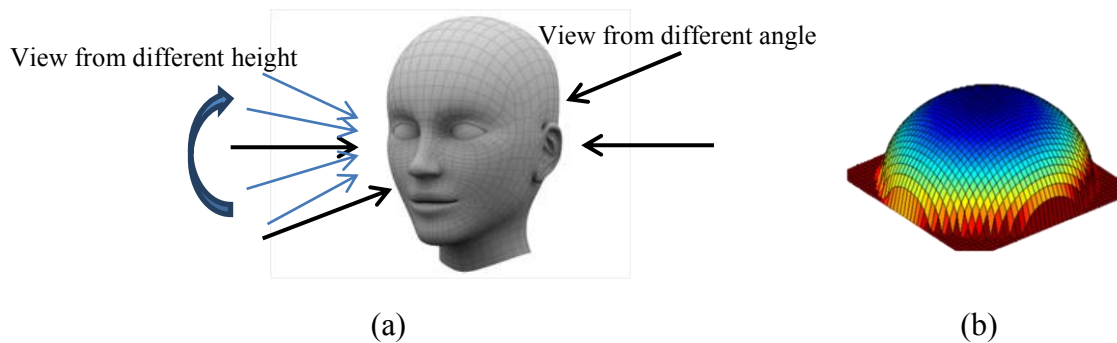


Figure 3.4: 3D head model. (a) illustrates the demands of the head model: the model should be invariant to different views. (b) shows the hemisphere model we used as the 3D head model

3.3.3 Fitting

Next, we fit the model onto the regions detected from previous steps. We extract a circular region CR with radius R_t around the detect center and normalize its depth:

$$depth_n(i, j) = depth(i, j) - \min_{i,j}(depth(i, j)) \quad i, j \in CR \quad (3.3)$$

Here, $depth(i, j)$ is the depth value of pixel (i, j) in the depth array. $depth_n(i, j)$ is the normalized depth value of pixel (i, j) . Then we calculate the square error between the circular region and the 3D model:

$$Er = \sum_{i,j \in CR} |depth_n(i, j) - template(i, j)|^2 \quad (3.4)$$

We use a threshold to decide whether the region is actually a head. Figure 3.5 illustrates some of the steps in this stage, and shows the result of the 3D matching.



Figure 3.5: (a) illustrates the process of estimating the true parameter of the head from the distance map. Input of the 3D model fitting is the output of the 2D Chamfer Distance Matching in Figure 3(e). Output of 3D model fitting is shown in (b). Yellow dots indicate the center of the head detected.

3.4 EXTRACT CONTOURS

We extract the overall contour of the person so that we may track his/her hands and feet and recognize the activity. In an RGB image, despite the person is standing on the ground, it is less a problem to detect the boundary between the feet and the ground

plane using gradient feature. However, in a depth array, the values at the person's feet and the local ground plane are the same. Therefore, it is not feasible to compute humans' whole body contours from a depth array using regular edge detectors. The same applies when the person touches any other object that is partially in the same depth with the person. To resolve this issue, we take advantages of the fact that persons' feet generally appear upright in a depth array regardless of the posture. We use the filter response of

$$F = [1, 1, 1, -1, -1, -1]^T \quad (3.5)$$

to extract the boundary between the persons and the ground.

The filter response after threshold delineates the planar areas that are parallel to the floor. The edges extracted by F filter response together with the original depth array are added together as the input to our region growth algorithm. Figure 3.6 shows an example of the filter response. (The color distributions of in both images are a little different because of we scale the array for display, the corresponding values are the same.)



Figure 3.6: (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 growth algorithm. The ground plane is delineated by the thresholded F filter response. The edges along the feet well separate the persons from the floor.

We develop a region growth algorithm to extract the whole body contours from the processed depth array. It is assumed that the depth values on the surface of a human object are continuous and vary only within a specific range. The algorithm starts with a seed location, which is the centroid of the region detected by 3-D 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 and y in the depth array is defined as:

$$S(x, y) = |depth(x) - depth(y)| \quad (3.6)$$

Here, S is similarity and $depth()$ 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:

$$depth(R) = \frac{1}{N} \sum_{i \in R} (depth(i)) \quad (3.7)$$

The pseudocode of the region growth algorithm is summarized in Table.3.1 The results of the region growth algorithm are shown in Figure 3.7.



Figure 3.7: (a) Result of our region growth algorithm. (b) The extracted whole body contours are superimposed on the depth map.

Start region growth until similarity between the region and neighboring pixels is higher than a threshold
i. Initialize: region = seed ii. (1) Find all neighboring pixels of the region (2) Measure the similarity of the pixels and the region (Eq.7) s_1, s_2, \dots and sort the pixels according to the similarity. (3) If $s_{min} < \text{threshold}$ (3.1). Add the pixel with the highest similarity to the region. (3.2). Calculate the new mean depth of the region. (3.3). Repeat (1)-(3) else algorithm terminate iii. Return the region

Table 3.1: Region growth algorithm

3.5 TRACKING

Finally, we give preliminary results on tracking using depth information based on our detection result. 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. But in depth images we don't have such color information. What we have is the 3D space information of the objects, so that we can measure the movements of the objects in a 3D space. Our tracking algorithm is based on the movements of the objects. We assume that the coordinates and speed of the same objects in neighboring frames change smoothly, i.e. there should not be big jumps in coordinates or speed. First, we find the center of the

detected blob. Then we calculate the 3D coordinates and speed of the persons in each frame. The coordinates are given in the depth array directly; the speed is calculated from the coordinates of neighboring frames. We define a energy score of the changes in space and speed:

$$E = (c - c_0)^2 + (v - v_0)^2 \quad (3.8)$$

Here, E is the energy score, c is the coordinates of the person in the current frame and c_0 is the coordinates of the person in the last frame. v is the speed of the person in the current frame and v_0 is the speed of the person in the last frame.

In the first frame, we label the person in turn according to the detection order. For the subsequent frames, we try all the possible matches of those people and take the one with the smallest energy score to be the solution. Special cases need to be handled when the total number of people in the frame changes, like when there are people get out of the scene or new persons join in.

3.6 EXPERIMENTAL RESULTS

In this section we describe the experiments performed to evaluate our method. We show both qualitative and quantitative results on our datasets and compare our approach with a window-based human detection algorithm [1].

3.6.1 Dataset

We evaluate our method using a sequence of depth arrays taken by the Kinect for XBOX 360 in indoor environment. We took the sequence in our lab with at most two persons presented in the scene. There are tables, chairs, shelves, computers, an overhead

lamp and so on presented in the scene. The people have a variety of poses, and they have interaction with others or the surrounding objects. There are 98 frames in the test set and the frame rate is 0.4s/frame. The size of the depth array is 1200×900 and the resolution is about 10mm.

To better illustrate our image frames, we scale the depth array and plot using color map ‘JET’ as in Figure 3.8. The depth is measured in millimeters and the points that failed to be measured are offset to 0 (which usually happen in the irregular edge of objects or surfaces that do not reflect infrared rays well, e.g. polyporous materials and when the objects are moving fast). 0-value in depth array corresponds to the dark blue in the image in (b).



Figure 3.8: (a) a patch of the depth array (b) the depth array showed using color map JET

3.6.2 Experimental Results

Our detection method performs well on our indoor dataset. Figure 3.9 shows some of the results of our algorithm.



Figure 3.9: Examples of the human detection result.

Figure 3.10 shows the preliminary tracking results based on our detection result. 15 consecutive frames are shown, which includes two people walking past each other, one person gets occluded and appears again. The detection method performs well in most cases. We do not have any FP instances but only a few FN detections. It happened when the person's head is occluded by another person or half of the body is out of the frame, as shown in Figure 3.11.

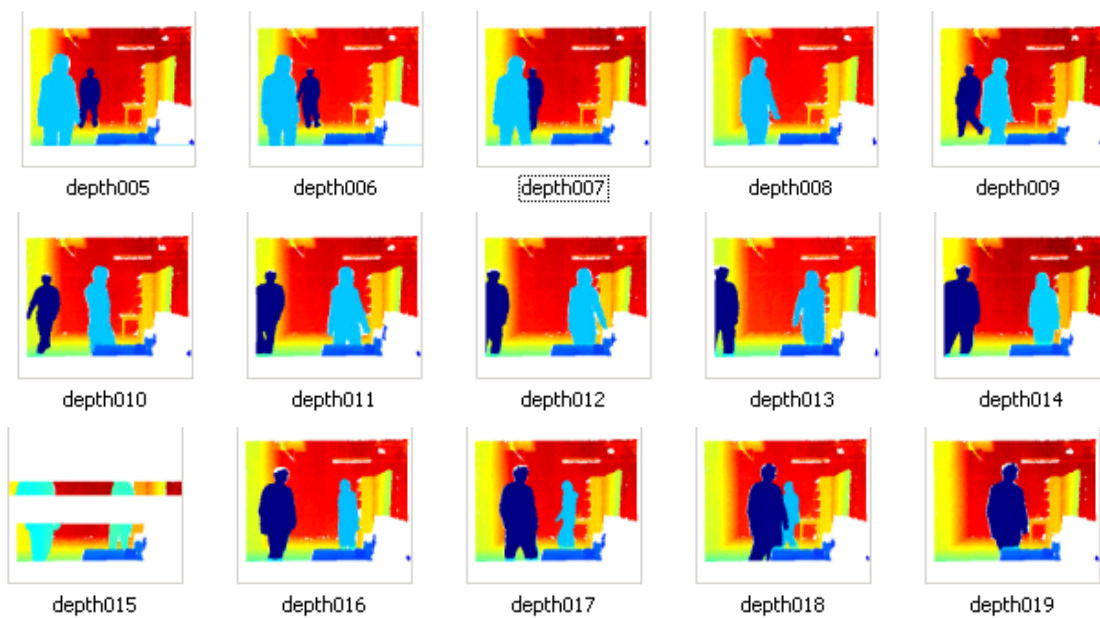


Figure 3.10: Tracking result. 15 consecutive frames are shown.

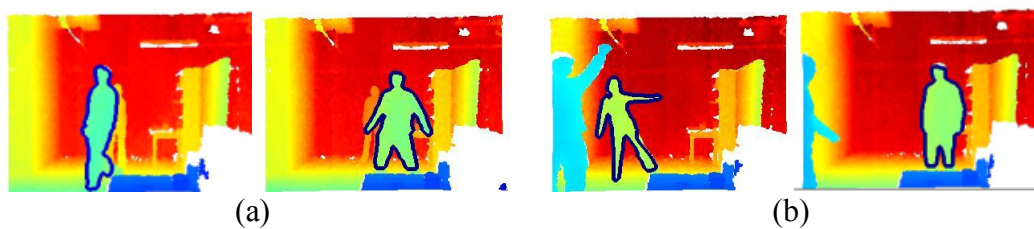


Figure 3.11: Examples of false negative detections. In group (a) the person that got occluded is not detected. In group (b) the person that is half way out of the frame is not detected.

We evaluate our method with different accuracy metrics, as shown in Table 3.2.

The precision, recall and accuracy are defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (3.9)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.10)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.11)$$

TP	TN	FP	FN
169	266	0	7
Precision		Recall	
100%		96.0%	
Accuracy			
98.4%			

Table 3.2: Accuracy of our human detection algorithm.

We compare our algorithm with a recent window-based algorithm which uses the relational depth similarity features for classification [1]. The original method uses TOF data. We perform an additive preprocessing step and implement the scheme on our Kinect data. Table 3.3 shows the comparison of performances of both methods. There are about 0 to 500 windows 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. Note that the unit here is window not frame.

	Precision	Recall	Accuracy
Ours	100%	96.0%	98.4%
Ikemura	90.0%	32.9%	85.8%

Table 3.3: Comparison of our algorithm with Ikemura

From Table 3.3 we can see that our algorithm outperforms Ikemura’s algorithm on this dataset. The main reason is that Ikemura’s 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. The recall of Ikemura’s algorithm is low because it is a window-based method which has a large false negative rate. The high false negative rate actually does not deteriorate his performance because the same person would appear in a lot of scanning windows. The algorithm will produce true positive when the person is well centered in the window, and it will classify the rest of the windows which the persons are not in the center as negative frames. And that is the cause of the high false negative rate. But the person in the image is successfully detected in this case.

To prove the privilege of using depth data and the effectiveness of our algorithm, we also compared our result with human detection algorithm performed on RGB data. We choose the most successful method HOG pedestrian detection algorithm [15] here, and run this algorithm on the RGB images. (Because we didn’t store the corresponding RGB images when we take the dataset, we recaptured the RGB images in the same room and same persons. So even though the RGB images and the depth images are not one to one corresponded, the scene and subjects in the images are the same. So we consider the difficulty of detect the humans are similar.) The result is shown in Figure 3.12. The first row shows examples of typical success cases. Second rows shows that the background clutter causes confusion for the HOG descriptor. 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 field of view (FOV), the pedestrian detector cannot detect the person. Fourth row gives examples when the algorithm totally missed the person even though the person is fully in view. Fifth row shows examples when the algorithm totally messed up the detection.



Figure 3.12: Detection Results using HOG pedestrian detection algorithm [15].

3.7 CONCLUSIONS

We propose a human detection method that utilizes the depth information obtained from the Kinect. The experimental results show that our algorithm can effectively detect the persons in all poses and appearances from the depth array, and it provides an accurate estimation of the whole body contour of the person. In addition, we explore a tracking algorithm based on our detection results. The approach can be applied in multiple tasks such as object segmentation, human detection, tracking, and activity analysis. And the algorithm is generally applicable to depth images acquired by other types of range sensors.

The advantages of our method are briefly described in the followings. First, the method can easily adjust to new datasets, no training is needed. Second, the algorithm uses a two layer detection process with 2D Chamfer Distance Matching in the first layer which largely reduces computational cost. Third, we do not assume person's pose for accurate detection. The limitation is that this algorithm has high dependency on accurate head detection, which implies that if the head is occluded or if the person is wearing a strange shape hat, it probably will not be detected, but this problem can be handled when we extend the head detector to other parts of the body, e.g. combine with hand detector or central body detector.

Chapter 4: View Invariant Action Recognition Using HOJ3D

4.1 OVERVIEW OF THE METHOD

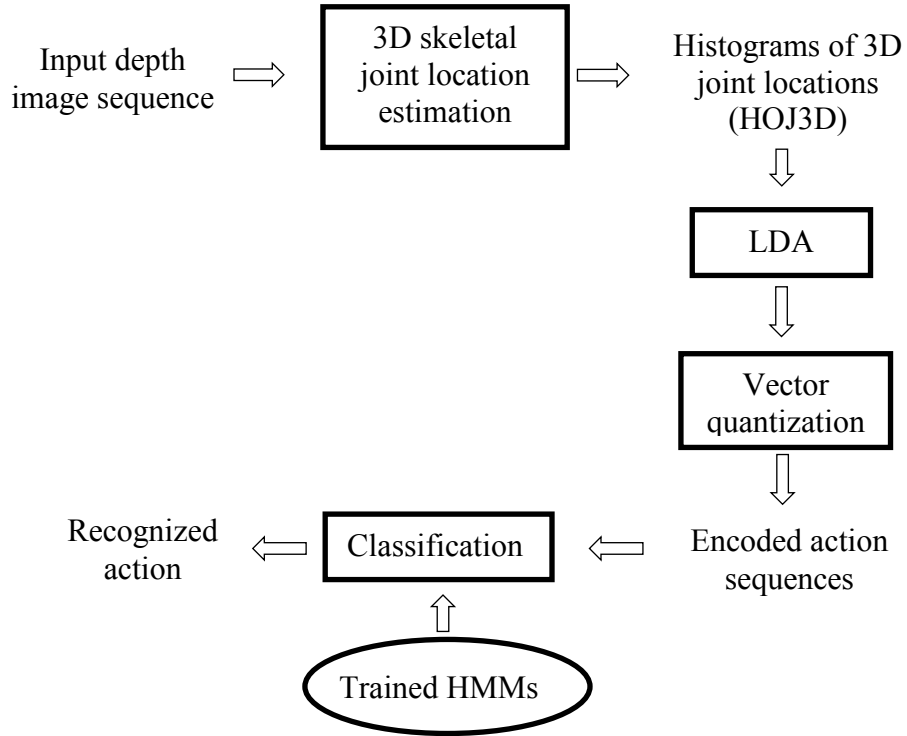


Figure 4.1: Overview of the method.

In this section, we present an action recognition algorithm using a compact representation of postures built from skeleton joint locations extracted from depth images. Taking the skeletal joint locations of one posture, we calculate the histograms of the 3D skeletal joint locations (HOJ3D) and take this as a compact representation of the posture. Then we reduce the dimensions of the HOJ3D features using LDA and cluster the features into k clusters using K-means. In this way, each posture is represented by a

single number which represent the id of the clusters and each action is simplified as a sequence of numbers. Finally, we classify the encoded action sequence using trained HMMs. Taking advantage of Kinect and J. Shotton et al.'s algorithm [2], this method improves on the previous methods in that it achieves excellent recognition rates and is also view invariant and real time.

4.2 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) [56]. Here, we use joint locations to build a compact representation of postures. The launch of Kinect offers a low-cost and real-time solution for the estimation of the 3D locations of objects or persons in the scene. Shotton et al. [2] 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). They compute the confidence-scored 3D position estimation of body joints by employing a local mode-finding approach based on mean shift with a weighted Gaussian kernel. Their gigantic and diverse training set allows the classifier to estimate body parts invariant of pose, body shape, clothing, and so on. Using their algorithm, we acquire the 3D locations of 20 skeletal joints which comprise 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 angle and L/ R foot. Note that

body part segmentation results are not directly available. Figure 4.2 shows an example result of 3D skeletal joints and the corresponding depth map.



Figure 4.2: (a) Depth image. (b) Skeletal joints locations by Shotton et al.

We use these skeletal joint locations to form our representation of postures. 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, we compute our 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. We take the hip center as the center of the reference coordinate system, and define the x-direction according to L/ R hip. The rest 9 joints are voted into the proposed 3D spatial histogram.

4.3 HOJ3D AS POSTURE REPRESENTATION

The estimation of 3D skeleton from RGB imagery is subject to error and significant computational cost. With the use of Kinect, we can acquire the 3D locations of the body parts in real-time with better accuracy. We propose a compact and viewpoint

invariant representation of postures. We use the 3D skeletal joints locations as the representation of postures, and employ a vocabulary of postures to describe the prototypical poses of actions.

4.3.1 Spherical Coordinates of Histogram

Our methodology is designed to be view invariant, i.e., descriptors of the same poses are similar despite being captured from different viewpoints. We achieve this by aligning our spherical coordinates with the reference vectors of the person. (Illustrate in Figure 4.3). 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.

We partition the 3D space into n bins as shown in Figure 4.4 (in our experiment, we take $n=84$). 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. With our spherical coordinate, any 3D joint can be localized at a unique bin.

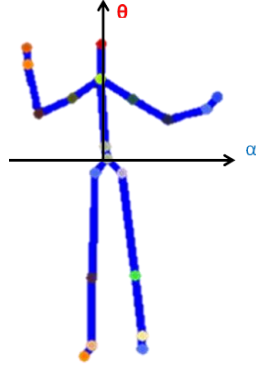


Figure 4.3: Reference coordinates of HOJ3D.

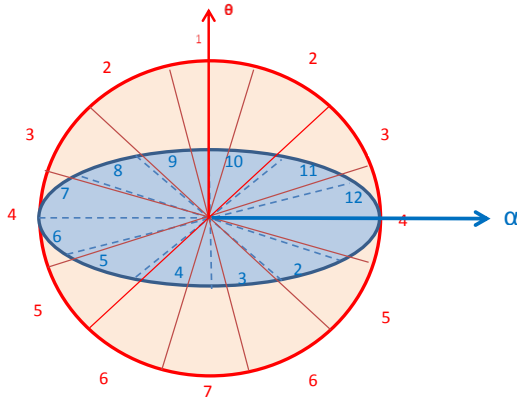


Figure 4.4: Modified spherical coordinate system for joint location binning.

4.3.2 Probabilistic Voting

Our HOJ3D descriptor is computed by casting the 9 joint locations into the corresponding spatial histogram bins. For each joint location, weighted votes are contributed to the geometrically surrounding 3D bins. To make the representation more robust against minor errors of joint locations, we vote the 3D bins using a Gaussian weight function:

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

where $p(X, \mu, \Sigma)$ is Gaussian probability density function with mean vector μ and covariance matrix Σ . For each joint, we only vote over the bin it is in and 8 neighboring bins. We calculate the probabilistic voting on θ and α separately since they are independent (see Fig. 4.5). 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 to bin $[\theta_1, \theta_2]$ is

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

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,

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

Then, the probability voting to bin $\alpha_1 < \alpha < \alpha_2, \theta_1 < \theta < \theta_2$ is:

$$p(\theta_1 < \theta < \theta_2, \alpha_1 < \alpha < \alpha_2; \mu, \Sigma) = \quad (4.4)$$

$$p(\theta_1 < \theta < \theta_2, \mu_\theta, \sigma) \cdot p(\alpha_1 < \alpha < \alpha_2, \mu_\alpha, \sigma)$$

The votes are accumulated over the 9 joints. As a result, a posture is represented by an n -bin histogram. Fig. 4.6 illustrates an instance of the computed histogram.

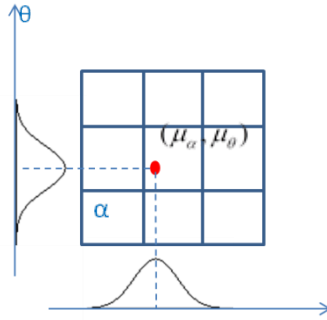


Figure 4.5: Voting using a Gaussian weight function.

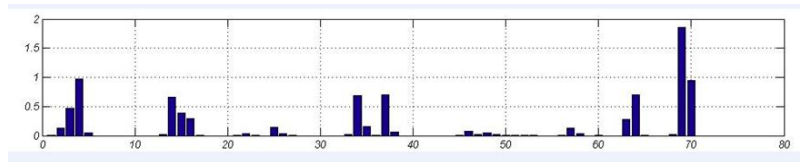


Figure 4.6: Example of the HOJ3D of a posture.

4.3.3 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 dimension to $nClass-1$ dimension.

4.4 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 words. In this way, each action is a time series of the visual words.

4.5 ACTION RECOGNITION USING DISCRETE HMM

We recognize a variety of human actions by the discrete HMM technique similar to what Rabiner did in speech recognition [39]. In discrete HMM, discrete time sequences are treated as the output of a Markov process whose states cannot be directly observed. In Section 4, we have encoded each action sequence as a vector of posture vocabularies, and we input this vector to learn the HMM model and use this model 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,

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

The $N \times M$ output probability matrix B is,

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

And the initial state distribution vector π is

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

We use a HMM to construct a model for each of the actions we want to recognize: the HMM gives a state based representation for each action. After forming the models for each activity, we 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 we classify the action as the one which has the largest posterior probability.

$$\text{decision}=\underset{i=1,2,\dots,M}{\operatorname{argmax}}\{L_i\} \quad (4.8)$$

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.

4.6 EXPERIMENTS

We tested our algorithm on a challenging dataset we collected ourselves. And we also test it on the public MSR Action3D dataset and compare our results with [3].

4.6.1 Data

To test the robustness of the algorithm, we collected a dataset of human indoor actions using Kinect. Kinect hardware has a practical range of about 4 to 11 feet. We evaluate our method on 10 indoor actions. We take the sequence indoors using a single stationary Kinect. The RGB images and depth maps were captured at 30 frames per second (FPS). The resolution of the depth map is 320×240 and resolution of the RGB image is 640×480 . We collected a dataset that contains 10 actions: *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. Each action sample spans about 5-120 frames. Sample RGB images from the dataset are shown in Fig. 4.7. Note that we only use the information from the depth map for action recognition in our algorithm; the RGB sequences are just for illustration.



Figure 4.7: Sample images from videos of the 10 activities in our database. Depth and RGB images are shown. Note only depth images are used in the 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*.

As shown in Table 4.1, 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. Table 4.2 is another example: individuals can toss an object with either their right or left arms or producing different trajectories. Second, the durations of the action clips vary dramatically. Table 4.3 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.









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

Table 4.1: Different views of the actions.





Person No.	Throwing sequence
Person 1	
Person 2	
Person 5	
Person 9	

Table 4.2: The variations of subjects performing the same action.

No.	1	2	3	4	5
Mean	43.60	34.15	25.60	35.50	58.15
Standard variation	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
Standard variation	4.10	4.24	7.72	16.30	20.14

Table 4.3: The mean and standard deviation of the sequence lengths measured by number of frames at 30 fps.

Action	ACC	Action	ACC
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%	Chap hands	100%
Overall: 90.92%			

Table 4.4: Recognition rate of each action type

4.6.2 Experimental Results

We evaluated our algorithm on our 200 sequences dataset using leave one sequence out cross validation (LOOCV). As there is randomness in the initialization of the cluster centroids and the HMM algorithm, we run the experiment 20 times and report the mean performance, shown in Table 4.4. We take the set of clusters to be 125. 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.

We also test our algorithm on the public MSR Action3D 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*. Shown in Figure 4.8. We selected one typical depth image for each action. We divide the actions into 3 subsets the same as in [3], each comprising 8 actions (see table 4.5). We use the same parameter settings as previously. Each test is repeated 20 times, and the average performance is shown in Table 4.6. We compare our algorithm’s performance with Li et al. [3]. We can see that our algorithm achieves considerably higher recognition rates than Li et al. [3] in all the testing setups on AS1 and AS2. On AS3, our recognition rate is slightly lower. As we have noticed in [3] 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 dataset while ours have higher accuracy on the other two dataset. We conjecture the reason to be that the complex actions effects adversely the HMM classification when the number of training samples is small. Note that our algorithm performs better on MSR Action3D dataset than on our own dataset, partially because of the following reasons: 1) the subjects were facing the camera during the activities; 2) the whole body is in view all the times; 3) if the action is performed by a single arm or leg, the subjects were advised to use their right arm or leg.

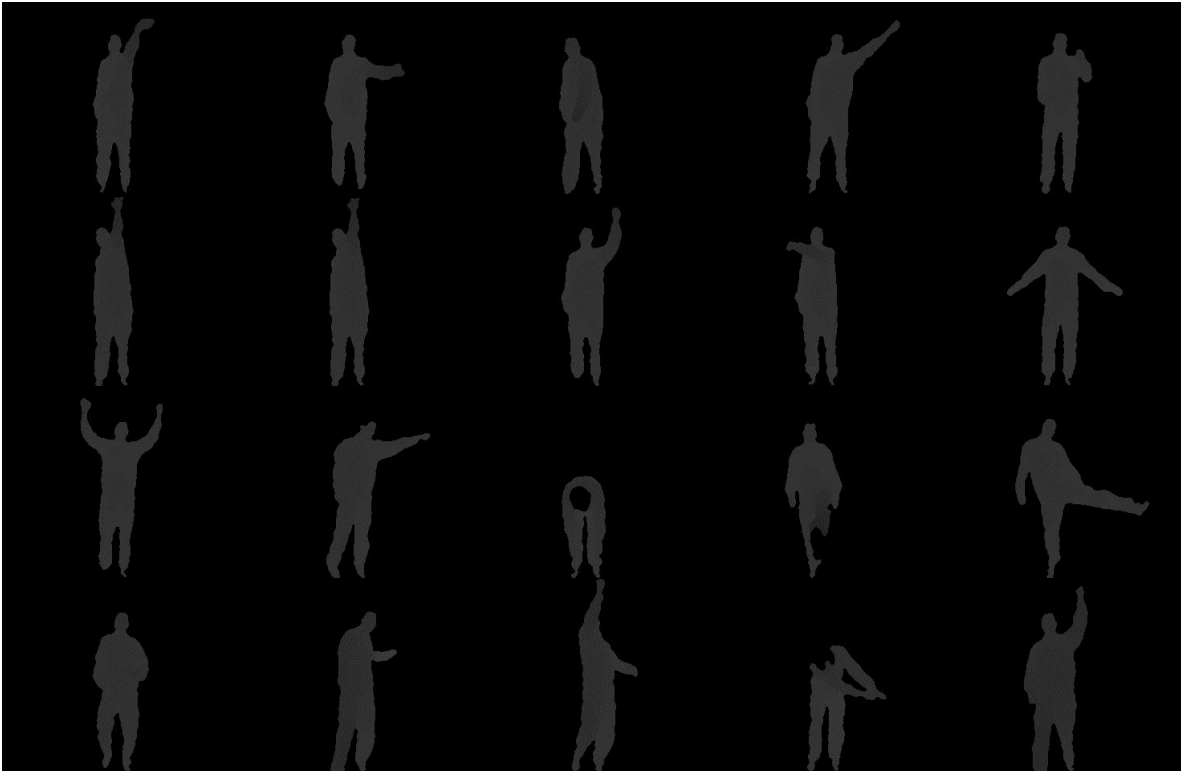


Figure 4.8 Sample depth images from the MSR Action3D dataset. One frame for each of the 20 actions is shown. Action type (from left to right, up to bottom): *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup & throw.*

Action Set 1 (AS1)	Action Set 2 (AS2)	Action Set 3 (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 4.5: The three subsets of actions used in the experiments.

	Test One		Test Two		Cross Subject Test	
	Li et al.	Ours	Li et al.	Ours	Li et al.	Ours
AS1	89.5%	98.47%	93.4%	98.61%	72.9%	87.98%
AS2	89.0%	96.67%	92.9%	97.92%	71.9%	85.48%
AS3	96.3%	93.47%	96.3%	94.93%	79.2%	63.46%
Overall	91.6%	96.20%	94.2%	97.15%	74.7%	78.97%

Table 4.6: Recognition results of our algorithm on the MSR Action3D dataset. We compared our result with Li et al. [3]. 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

4.7 CONCLUSIONS

We have presented a methodology to recognize human action as time series of representative 3D poses. We take as input 3D skeletal joints locations inferred from depth maps as input. We proposed a compact representation of postures named HOJ3D that characterizes human postures as histograms of 3D joint locations within a modified spherical coordinate system. We build posture vocabularies by clustering HOJ3D vectors calculated from a large collection of postures. We train discrete HMMs to classify sequential postures into action types. The major components of our 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 our view invariant representation.

This work also suggests the advantage of using 3D data to recognize human actions and points out a promising direction of performing recognition tasks using depth information. Traditional RGB information can also be combined with the depth data to provide more data and produce algorithms with better recognition rates and robustness.

Chapter 5: Conclusion & Future Work

We have presented our approaches for human detection and action recognition from depth images. The main novelty of this work is the use of depth information instead of traditional RGB images. In the first place, we proposed a simple model based algorithm to detect human from depth images, then we further presented an algorithm to recognize the human actions which is view invariant and runs at real-time.

The algorithms are tested on existing and new datasets collected using Kinects. We have shown that the 3D information in the depth image simplified the traditional human detection and action recognition tasks. And the 3D information is robust to color and illumination changes and background clutter. In addition, the 3D information also makes it easier to realize view-invariance. We compared our performance with existing algorithms using depth images and RGB images and shown that our algorithm has superior result and faster speed.

In the current approach, the action recognition algorithm depends on accurate estimation of the skeletal joint locations. There are mainly two drawbacks of this dependence: first, the accuracy of the algorithm is limited by the accuracy of the skeleton extraction algorithm. Second, there is information loss when extracting the skeleton from the depth image. In the future, we will consider building 3D surface of the person and recognizing his/her action from the reconstructed 3D surface. Depend on the demand, multiple Kinects might be used. We will also investigate on combining RGB information with depth information to provide more information and build more robust algorithms. And it is also an interesting and challenging topic to study human-human interaction and human-objects interaction.

Bibliography

- [1] S. Ikemura, H. Fujiyoshi.: Real-Time Human Detection using Relational Depth Similarity Features. ACCV 2010, *Lecture Notes in Computer Science*, 2011, Volume 6495/2011, 25-38
- [2] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A.x Kipman, and A. Blake, Real-Time Human Pose Recognition in Parts from a Single Depth Image, in CVPR, IEEE, June 2011
- [3] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 9 –14, June 2010. 1, 5
- [4] J. Goles, “Inside the race to hack the Kinect,” *New Scientist*, vol. 208, no. 2789, p.22, December 2010.
- [5] B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. *Depth mapping using projected patterns. Patent Application*, 10 2008. WO 2008/120217 A2.
- [6] M. Breidt, H. Buelthoff, and C. Curio, “Robust semantic analysis by synthesis of 3d facial motion,” in *AFGR*, 2011.
- [7] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011.
- [8] M. Mancas, R.B. Madhkour, D. De Beul, J. Leroy, N. Riche, Y. P. Rybarczyk, F. Zajega, "Kinect: a Saliency-based Social Game", *QPSR of the numediart research program*, volume 4, no. 3, September 2011, pp. 65-70.
- [9] M. Mancas et al. “Toward a Social Attentive Machine”. In: *AAAI Fall Symposium 2011*. Pp.: 65, 67–69.
- [10] M. Ferguson, K. Gero, J. Salles, and J. Weis, Playing Chess with a Human-Scale Mobile Manipulator. In *AAAI*, 2011 .
- [11] L. Xia, C.-C. Chen, and J. K. Aggarwal, "Human Detection Using Depth Information by Kinect", *International Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR (HAU3D)*, Colorado Springs, CO, June 2011.
- [12] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *BMVC*, Aug 2011.
- [13] J. Sung, C. Ponce, B. Selman, Ashutosh Saxena: Human Activity Detection from RGBD Images. *Plan, Activity, and Intent Recognition*, 2011

- [14] H. Zhang and L. E. Parker, 4-Dimensional local spatio-temporal features for human activity recognition. In *Proc. Of IEEE International conference on Intelligent robots and systems*, San Fransisco, CA, 2011.
- [15] N. Dalal and B. Triggs.: Histograms of oriented gradients for human detection. *CVPR*, 1 (2005) 886-893.
- [16] N. Dalal, B. Triggs, C. Schmid.: Human detection using oriented histograms of flow and appearance, in *European Conference on Computer Vision*, Graz, Austria, May 7–13, 2006
- [17] W. Schwartz, A. Kembhavi, D. Harwood, L. Davis, Human detection using partial least squares analysis. In: *ICCV* (2009)
- [18] K. Levi and Y. Weiss. Learning object detection from a small number of examples: the importance of good features. *CVPR* 2(2004) 53-60
- [19] D. G. Lowe.: Object Recognition from Local Scale-Invariant Features. *Proceedings of the International Conference on Computer Vision*. 2 (1999). pp.1150–1157
- [20] T. Darrell, G. Gordon, J. Woodfill, and M. Harville, "Integrated Person Tracking using Stereo, Color, and Pattern Detection," *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, June 1998
- [21] D. Demirdjian, T. Darrell: 3-D Articulated Pose Tracking for Untethered Diectic Reference. *ICMI 2002*: 267-272
- [22] H.D. Yang, S.W. Lee: Reconstruction of 3D human body pose from stereo image sequences based on top-down learning. *Pattern Recognition* 40(11): 3120-3131 (2007)
- [23] V. Ganapathi, C. Plagemann, D. Koller, S. Thrun. Real time motion capture using a single time-of-flight camera. *Proceedings of CVPR 2010*. pp.755~762
- [24] H. Jain and A. Subramanian. Real-time upper-body human pose estimation using a depth camera. In *HP Technical Reports*, HPL-2010-190, 2010
- [25] J. Rodgers, D. Anguelov, H.-C. Pang, and D. Koller. Object pose detection in range scan data. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006
- [26] Y. Zhu, B. Dariush, and K. Fujimura. Controlled human pose estimation from depth image streams. *Proc. CVPR Workshop on TOF Computer Vision*, June 2008
- [27] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Realtime identification and localization of body parts from depth images. In *IEEE Int. Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, USA, 2010

- [28] P. Turaga, R. Chellapa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.
- [29] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. In *ACM Computing Surveys*, 2011.
- [30] W. Li, Z. Zhang, and Z. Liu. Expandable data-driven graphical modeling of human actions based on salient postures. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1499–1510, 2008.
- [31] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.
- [32] H. Meng, N. Pears, and C. Bailey. A human action recognition system for embedded computer vision application. In Proc. *CVPR*, 2007.
- [33] J. W. Davis and A. Tyagi. Minimal-latency human action recognition using reliable-inference. *Image and Vision Computing*, 24(5):455-473, 2006.
- [34] D.-Y. Chen, H.-Y. M. Liao, and S. -W. Shih. Human action recognition using 2-D spatio-temporal templates. In Proc. *ICME*, pages 667-670, 2007.
- [35] V. Kellokumpu, M. Pietikainen, and J. Heikkila. Human activity recognition using sequences of postures. In Proc. *IAPR Conf. Machine Vision Applications*, pages 570-573, 2005.
- [36] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional models for contextual human motion recognition. In Proc. *ICCV*, volume 2, pages 808-815, 2005.
- [37] J. Zhang and S. Gong. Action categorization with modified hidden conditional random field. *Pattern Recognition*, 43: 197-203, 2010.
- [38] W. Li, Z. Zhang, and Z. Liu. Expandable data-driven graphical modeling of human actions based on salient postures. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1499-1510, 2008.
- [39] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of the *IEEE*, 77(2), February, 1989. pp. 257-285.
- [40] G. Johansson: Visual motion perception. *Sci. Am.* 232(6), 76–88 (1975)
- [41] H. Fujiyoshi and A. Lipton. Real-time human motion analysis by image skeletoniation. In *IEEE Workshp on Applications of Computer Vision*, pages 15-21, Princeton, 1998.
- [42] E. Yu and J. K. Aggarwal, "Human Action Recognition with Extremities as Semantic Posture Representation", *International Workshop on Semantic Learning*

- and Applications in Multimedia (SLAM, in conjunction with CVPR)*, Miami, FL, June 2009.
- [43] M. Z. Uddin, N. D. Thang, J.T. Kim and T.S. Kim, Human Activity Recognition Using Body Joint-Angle Features and Hidden Markov Model. *ETRI Journal*, vol.33, no.4, Aug. 2011, pp.569-579.
 - [44] C. Schödl, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, pages III: 32–36, 2004.
 - [45] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.
 - [46] Y. Wang, P. Sabzmeydani, and G. Mori. Semi-latent dirichlet allocation: A hierarchical model for human action recognition. In *Human Motion Workshop, (with ICCV)*, 2007.
 - [47] T. J. Darrell, I. A. Essa, and A. P. Pentland, “Task-specific gesture analysis in real-time using interpolated views,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1236– 1242, 1996.
 - [48] T. F. Syeda-Mahmood, M. Vasilescu, and S. Sethi, “Recognizing action events from multiple viewpoints,” *IEEE Workshop on Detection and Recognition of Events in Video*, pp. 64–72, 2001.
 - [49] D. Weinland, E. Boyer and R. Ronfard. Action recognition from arbitrary views using 3D exemplars, *ICCV* 2007.
 - [50] V. Parameswaran and R. Chellappa, “View invariants for human action recognition,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 613–619, 2003.
 - [51] D. Weinland, R. Ronfard, and E. Boyer, “Free viewpoint action recognition using motion history volumes”, *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 249–257, 2006.
 - [52] J. W. Davis and A. F. Bobick, “The representation and recognition of human movement using temporal templates,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 928–934, 1997.
 - [53] A. Jalal, M. Z. Uddin, J.T. Kim and T.S. Kim, Recognition of Human Home Activities via Depth Silhouettes and R Transformation for Smart Homes, *Indoor and Built Environment*, DOI: 10.1177/1420326X11423163, 1-7, 23 Sep 2011
 - [54] <http://www.3dvsystems.com>
 - [55] J. Sung, C. Ponce, B. Selman and A. Saxena, Human Activity Detection from RGBD Images, In *AAAI workshop on Pattern, Activity and Intent Recognition (PAIR)*, 2011

- [56] V. M. Zatsiorsky. Kinematics of Human Motion. *Human Kinetics Publishers*, 1997.
- [57] B. Gil, A. Mitiche, and J. K. Aggarwal, Experiments in Combining Intensity and Range Edge Maps. In *Computer Vision, Graphics and Image Processing* 21, 395-411, 1983.
- [58] M. J. Magee and J. K. Aggarwal, Using Multisensory Images to Derive the Structure of Three-Dimensional Objects-A Review. In *Computer Vision, Graphics and Image Processing*. 32, 145-157, 1985.
- [59] B. C. Vemuri and A. Mitiche and J. K. Aggarwal, "Curvature-based Representation of Objects from Range Data," *Image and Vision Computing*, Vol. 4, No. 2, pp. 107-114, May 1986.
- [60] M. J. Magee, B.A. Byter, C. H. Chien and J. K. Aggarwal, Experiments in Intensity Guided Range Sensing Recognition of Three-Dimensional Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, No. 6, November, 1985.
- [61] B. Vemuri and A. Mitiche and J. K. Aggarwal, "3-D Object Representation from Range Data Using Intrinsic Surface Properties," Edited by T. Kanade, *Three-Dimensional Machine Vision*, pp. 241-266, Kluwer Academic Publishers, 1986.
- [62] C-C. Chu and J. K. Aggarwal, Image Interpretation Using Multiple Sensing Modalities. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL. 12, No. 8, August, 1992.
- [63] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 747-757, August, 2000.
- [64] T.-S. Kim and Z. Uddin. Silhouette-based Human Activity Recognition Using Independent Component Analysis, Linear Discriminant Analysis and Hidden Markov Model, *New Developments in Biomedical Engineering*, Domenico Campolo (Ed.), ISBN: 978-953-7619-57-2
- [65] M. Ahmad and S. W. Lee, HMM-based human action recognition using multiview image sequences, *18th International Conference on Pattern Recognition*, Vo. 1, pages 263-266, 2006
- [66] F. Niu, and M. Abdel-Mottaleb, View-invariant human activity recognition based on shape and motion features, *IEEE Sixth International Symposium on Multimedia Software Engineering*, 2004, pages 546-556.
- [67] P. Peursum, H. H. Bui, S. Venkatesh, G. West, Technical Report 2004/01, Human action recognition with an incomplete real-time pose skeleton, *Curtin University of Technology*, May 2004

- [68] H. S. Chen, H. T. Chen, Y.W. Chen, and S.Y. Lee, Human action recognition using star skeleton, *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 171-178, 2006
- [69] F. Xiao, P. Hua, L. Jin, and Z.Bin, Human gait recognition based on skeletons, *International Conference on Educational and Information Technology (ICEIT)*, vol. 2, pages V2-83, 2010
- [70] Y. Shen and H. Foroosh, View-invariant action recognition using fundamental ratios, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* , pages 1-6, 2008.
- [71] L. Deng, H. Leung, N. Gu and Y. Yang, Generalized Model-Based Human Motion Recognition with Body Partition Index Maps, *Computer Graphics Forum*, vol. 31, no. 1, pages 202-215. 2012.
- [72] D. Gehrig, T. Schultz, Selecting Relevant Features for Human Motion Recognition, *19th International Conference on Pattern Recognition*, 2008
- [73] F. Lv and R. Nevatia, Recognition and segmentation of 3D human action using HMM and multi-class Adaboost, *ECCV*, pages 359-372, 2006.
- [74] J. Ben-Arie, Z. Wang, P. Pandit and S. Rajaram, Human activity recognition using multidimensional indexing, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pages 1091-1104, 2002
- [75] S. M. Yoon and A. Kuijper, Human action recognition using segmented skeletal features, *ICPR*, 2010.

Vita

Lu Xia was born in China on April 24th, 1988. She received the Bachelor of Science in Control Engineering from Tsinghua University, School of Information Science and Technology, in 2010. During her undergraduate studies, she was awarded with Academic Outstanding Awards in three consecutive years. And she received Bronze Medal in National Physics Competition in 2007. She worked as a research assistant at the Information Lab in the Department of Automation from 2009 to 2010, researching on fingerprint and palm-print recognition. She entered the University of Texas at Austin in fall 2010, in the Department of Electrical and Computer Engineering. She joined the Computer & Vision Research Center and works under supervision of Dr. J. K. Aggarwal from January 2011.

Permanent address (or email): Fenglinlvzhou, Beishatan, Chaoyang District
Beijing, China, 100101.

This thesis was typed by the author.