Markerless motion capture and analysis based on depth images

Bian, Zhenpeng

2015

Bian, Z. (2015). Markerless motion capture and analysis based on depth images. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65865>

<https://doi.org/10.32657/10356/65865>

MARKERLESS MOTION CAPTURE AND ANALYSIS BASED ON DEPTH IMAGES

Bian Zhenpeng

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2015

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....
Date

.....
Bian Zhenpeng

Abstract

The works presented in this thesis focus on depth images based human motion capture in realistic daily scenarios and two novel motion analysis frameworks on fall detection and human-computer interface based on motion capture algorithms. We propose a body part extraction algorithm to improve the robustness, speed and accuracy for human motion capture compared to existing methods. Based on the new motion capture method, we propose two motion analysis frameworks, resulting in better performance than the state-of-the-art methods, especially their robustness and low computational complexity.

The major contributions of this thesis are:

- (i) A fast training algorithm based on a compact feature for markerless motion capture.
- (ii) A robust fall detection framework based on markerless motion capture.
- (iii) A convenient and robust human computer interface for people with disabled hands.

Traditional motion capture approaches obtain the information from the motion of markers, which are mostly used in laboratory environments by professional users. More and more applications require the motion capture techniques, especially the markerless motion capture techniques. This thesis develops markerless motion capture methods using depth images, which can produce a low cost and convenient system for motion capture applications. To capture the motion of the human body, the human body parts are extracted by the proposed Randomized Decision Tree (RDT) algorithm. The training phase affects the robustness and accuracy of the RDT algorithm. The dataset for training of the RDT algorithm should be large enough to cover enough cases. One of the advantages of the depth image framework is that it is easier to build up large realistic training dataset compared with the RGB image framework. Furthermore, the depth images are

independent of colour and lighting illumination. However, one of the challenges of the RDT algorithms is the high computational complexity of training, which is caused by the huge amount of the feature candidates for each training pixel. We propose to reduce the amount of the feature candidates by search algorithms based on a compact feature. This fast training algorithm can dramatically speed up the training phase without loss of accuracy in the test phase.

The first novel motion analysis framework based on motion capture is fall detection. Researches have shown that getting help quickly after a fall can significantly reduce the hospitalization risk and the death risk for elderly people. Based on the markerless motion capture technology, we propose a real-time fall detection framework by analysing the 3D head trajectory. This system is more convenient since it is a non-intrusive approach compared to the wearable sensor based fall detection approaches. There are two major challenges of the fall detection: the daily activities are very variable, and the environmental illumination is complexly. The proposed body part extraction algorithm can track the human body in different human poses, and the proposed system is independent of illumination. Then, the 3D head trajectory is analysed by the Support Vector Machine (SVM) classifier.

The second framework is the human computer interface. The people with disabled hands are very difficult to use personal computer's standard interfaces. Therefore, we propose a human computer interface for the persons with tetraplegia based on markerless motion capture via a single depth camera. The nose position navigates the cursor of computer, and the mouth status triggers the input commands. Compared with many other Assistive Technologies (ATs) for persons with tetraplegia, this system does not need calibration, and enables users to adjust their head postures relative to the cursor freely. The proposed method can detect the nose position and the mouth status in a single depth image, and can avoid the problem of feature drift, which exists in most tracking approaches. During the training phase, the mouth is labeled according to the status of the mouth and the nose is set as a single label. During the test phase, the nose position and the mouth status can be extracted after classifying each pixel. Compared to the landmark-based mouth status detection, the proposed mouth status detection is more efficient and robust. The experimental results show that the performance of the proposed interface is superior to the state-of-the-art ATs.

Acknowledgments

There are many people who assisted me in my past four years' study and research experience.

First, I would like to express my sincere appreciation to my supervisors Associate Professor Lap-Pui Chau and Professor Nadia Magnenat Thalmann, for their encouragement and guidance, which has enriched the content of the thesis. They showed me a clear research path. Under their supervision, I have learned not only the knowledge on motion capture and motion analysis, but also an inspiring and down-to-earth working attitude. In addition, I would also like to thank Dr. CHEW Boon Seng, Miss KIM Ming Hwa, Mr TAN Cheen Hau, Mr CHEN Jie, Mr HOU Junhui and Mr LIANG Hui for many fruitful discussions and suggestions. And I am thankful to other colleagues in my research laboratory for their kind support and assistance. They have provided me with a supportive and enjoyable research environment.

I would also like to acknowledge my beloved parents and wife for their lasting encouragement and support that make my heart strong.

I would like to acknowledge the Ph.D. grant from the Institute for Media Innovation, Nanyang Technological University, Singapore.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	vii
1 Introduction	1
1.1 Motivation and Objective	1
1.2 Contributions	6
1.2.1 Fast Training Algorithm Based on a Compact Feature for Markerless Motion Capture	6
1.2.2 Robust Fall Detection Framework Based on Markerless Motion Capture	6
1.2.3 Convenient and Robust Human Computer Interface for People with Disabled Hands	7
1.3 Organization of the Thesis	8
2 Background	9
2.1 Motion Capture Algorithm	10
2.2 Fall Detection for Elderly People	15
2.2.1 Definition of a Fall	18
2.3 Human Computer Interface for Hand-Disabled People	19
3 Markerless Motion Capture Algorithm	22
3.1 Related Work	22
3.1.1 State-of-the-Art of Markerless Motion Capture Based on the RGB Camera	22

3.1.2	State-of-the-Art of Markerless Motion Capture Based on the Depth Camera	27
3.2	RDT Algorithm	37
3.2.1	The Feature of RDT	38
3.2.2	Aggregating Predictions	42
3.2.3	Training	42
3.3	Fast Training Algorithm Based on a Compact Feature	45
3.3.1	The Compact Feature	45
3.3.2	Fast Training Algorithm	47
3.3.3	Experimental results	51
3.4	Conclusion	53
4	Fall Detection Framework Based on Markerless Motion Capture	54
4.1	Related Work	54
4.1.1	Classification of Fall Detection Techniques	55
4.1.2	Wearable Device Based Techniques	57
4.1.3	Ambient Device Based Techniques	60
4.1.4	Vision Based Techniques	62
4.1.5	Discussion on the Existing Techniques	69
4.2	Overview of the Proposed Approach	69
4.3	Joint Detection	71
4.3.1	The Feature for Joint Detection	71
4.3.2	Pose Correction	72
4.4	Fall Detection Based on SVM	76
4.4.1	Fall Motion Analysis Based on SVM	76
4.4.2	Training Dataset	78
4.4.3	Fall Confirmation	79
4.5	Experimental Results	80
4.5.1	Performance Evaluation Metric	80
4.5.2	Dataset	81
4.5.3	Performance Validation	82
4.5.4	Comparison	84
4.6	Conclusion	90

5	Human Computer Interface Based on Markerless Motion Capture	92
5.1	Related Work	92
5.2	Overview of FM	108
5.3	Detection of Facial Position and Expression	111
5.3.1	Training Data	113
5.3.2	Feature of RDT	113
5.3.3	Decision Method of Nose Position and Mouth Status	117
5.3.4	Speed Up by Pyramid Processing	118
5.4	Mouse Events Based on Facial Position and Expression	119
5.4.1	Mapping Function of Motion	120
5.4.2	Commands Triggered by Mouth Status	121
5.5	Experiments	121
5.5.1	Evaluation of Detection Algorithm	122
5.5.2	Evaluation of Interface Operation	126
5.6	Conclusion	135
6	Conclusion and Future Work	137
6.1	Conclusion	137
6.1.1	Fast Training Algorithm	138
6.1.2	Fall Detection	139
6.1.3	Human Computer Interface	140
6.2	Future Work	142
6.2.1	Fall Risk Estimation	142
6.2.2	GUI System of Human Computer Interface	143
	List of Publications	144
	References	145

List of Figures

1.1	Markers suit in [106].	1
1.2	An example of marker based motion capture system [106].	3
1.3	Depth cameras. (a) Kinect [69]. (b) SoftKinetic [97]. (c) Creative [20]. (d) A close-range depth camera [26].	4
2.1	Applications of human motion capture [40].	10
2.2	A traditional motion capture system based on markers and multiple cam- eras [124].	11
2.3	3D human skeleton model [124].	11
2.4	The background and the subjects with marker suit of marker based motion capture system [40].	12
2.5	The processing machine structure for the motion capture system [23]. . .	13
2.6	Predicted population by age groups all over the world in the year of 2050 [22].	15
2.7	Age-specific rates of fall related deaths, Australia 2004-05 [39].	16
2.8	Physical, mental and social impacts of falls.	17
2.9	Muscle representation for the C3 and C4 spinal cord injury [98].	20
3.1	Overview of a markerless motion capture framework in [21]. The input is a laser-scan mesh model and some multi-view video streams. The method deforms the mesh model as its real-world counterpart in the video streams.	23
3.2	The workflows of tracking steps A and B [21]. Step A is denoted as Arabic numerals. Step B is denoted as Roman numerals.	24
3.3	The framework of the system in [52].	24
3.4	Overview of the GPU accelerated motion capture system in [131].	25

3.5	Separation of the human torso and detection of the base point [59]. (a) The 2D silhouette of the human body. (b) The result of DT. (c) The results of the torso and the base point. (d) The results of the torso and the base point in the color image.	25
3.6	The human skeleton model fitting iteration in [59].	26
3.7	Example of the depth ambiguity [41]. One 2D silhouette can have more than one corresponding 3D human poses.	26
3.8	Overview of the system in [50].	27
3.9	The 3D body model used in [35].	28
3.10	System diagram of the entire pipeline in [132].	29
3.11	Features and head-neck-torso (H-N-T) template used in [132].	29
3.12	Different poses with the matched body prototype overlaid results in [56].	29
3.13	Overview of the full body estimation system in [5].	30
3.14	Left: a representation of a human head. Meddle: the foreground of the left figure. Right: a graphical example of the elliptical mask used in [104].	31
3.15	The bounding box of two hands used in [104].	31
3.16	The 3D articulated body model with vertex skin used in [31].	32
3.17	A sample of the typical tracking situation in [31].	33
3.18	Overview of the motion capture system based on an RDT algorithm in [95].	33
3.19	Randomized decision tree in [95]. Black circlers: split nodes. Red circlers: leaves. A particular input may be taken the blue arrow paths.	34
3.20	Overview of the motion capture system combined the RDT algorithm and the 3D human model in [119].	35
3.21	The 3D human skeleton model in [119].	36
3.22	The calibration human pose of the system in [119].	36
3.23	Overview of the system in [16].	37
3.24	The iteration is used in the system of [16].	37
3.25	Randomized decision tree in [95].	40
3.26	The label map after the classification of RDT, each color stands for the respective part of the human body with the highest probability [95, 119]. (a) A map in [119]. (b) A map in [119].	41

3.27	Predicted joints' positions by mean-shift method of the human body of Figure 3.26.a [95].	43
3.28	An example to show the efficiency of Eq. 3.1 and Eq. 3.10 to describe an object with a certain shape. The normalization operation is included in $z(\bullet)$. The object has only three points. Eq. 3.1 has many more features than Eq. 3.10, but both of them can completely determine the same object. Many features for Eq. 3.1 are redundant.	46
3.29	Randomized decision tree based on a compact feature.	46
3.30	Entropy map and 3D view. (a) Map of entropy. (b) 3D view of the map.	48
3.31	The difference between the surfaces of two adjacent pixels is very small. Red points are the test pixels.	49
3.32	The feature value is changing smoothly. (a) Two adjacent offset pixels (green points). The difference in feature values between these two adjacent offset pixels is very small. (b) A same offset (green points) in two objects. The difference between feature values of these two objects is very small.	50
4.1	Overview of different fall detection methods.	56
4.2	The flow chart of fall detection in [117].	61
4.3	Overview of the proposed fall detection system.	71
4.4	The classification result of each depth pixel, each color stands for the respective part of the human body with the highest probability. Blue: head, red: hip, green: other part.	72
4.5	Joint extraction results.	73
4.6	After pose rotation, the joints of head and hip centre can be detected correctly.	74
4.7	A fall sequence by a person simulating. (a) The sequence of original poses without correction. (b) The sequence of with correction by proposed approach.	75
4.8	An example of re-initialization.	76
4.9	Two patterns of head distance trajectory from a same video sequence. (a) Fall pattern. (b) Non-fall pattern.	78

4.10	Free fall body trajectory fits the head distance trajectory. (a) Without noise. (b) Free fall body with noises.	79
4.11	Non-fall patterns simulated by computer.	79
4.12	Room plan for the evaluation.	83
4.13	The head distance trajectory during the fall of an FN sample of fall motion detection by SVM classifier while the circle indicates the rebound behaviour.	84
4.14	In the experiments, there is an air mattress on the floor to protect the subject.	85
4.15	Two examples of the angle between the orientation (extracted from Kinect SDK) and the floor when the subject walked and fell down ending lying. (a) Two depth image samples from two sequences. (b) Angle waveforms corresponded to the two sequences.	88
5.1	Categories and examples of ATs.	93
5.2	Muscles used for the EMG command source in [120].	94
5.3	The cursor trajectory to a target[120].	95
5.4	The EEG system in [122] controls the computer cursor to point the targets.	96
5.5	The EOG system in [17]. The output signal is right eye movement.	96
5.6	The overview of the EOG system in [17].	97
5.7	The sip-and-puff system in [73].	98
5.8	The mouth pad system used in [103, 102]. (a) The mount pad mounted in the mouth [103]. (b) and (C) two examples of mouth pad devices in [103] and [102].	99
5.9	Eye tracking system used in [130].	100
5.10	Two examples of feedback methods [63, 37, 62].	101
5.11	Overview of the eye system used in [114].	101
5.12	The experiment setting and of the eye system used in [114]. (a) System setting. (b)12 gaze calibration points.	102
5.13	Components of the tongue based human computer interface used in [42].	103
5.14	A subject is using the tongue based human computer interface used in [42].	103
5.15	Two examples of the tongue based interface attaching to the cheek [19].	104

5.16 Overview of the tongue based interface used in [19].	104
5.17 Five examples of gesture pressure maps: “click”, “swipe left”, “swipe right”, “swipe up” and “swipe down” [19].	104
5.18 A head band with inertial sensors for head motion based interface used in [86].	105
5.19 Some templates for tracking [8].	106
5.20 Schematic diagram of the search window [8].	106
5.21 Flow chart of the system used in [71].	107
5.22 Face regions by colour clustering in [71].	108
5.23 The 3D facial articulation model used in [116].	109
5.24 A result of initialization of the 3D facial articulation model [116].	109
5.25 Overview of our proposed human computer interface.	111
5.26 Labelled images for training. (a) Close the mouth. (b) Open the mouth. Green: mouth-close, red: mouth-open, white: nose, blue: head, brown: body.	114
5.27 Structure of our RDT. Each split node n contains a comparison between the feature value f_n and threshold τ_n . Each leaf contains a posterior distribution.	114
5.28 Results of RDT classification. (a)(c) Input depth images shown as mesh after removing pixels with near/far distance. (b)(d) Labels map after RDT classification, each colour stands for the highest probability label, green: mouth-close, red: mouth-open, white: nose, blue: head, brown: body. (a)(b) Close the mouth. (c)(d) Open the mouth.	116
5.29 Pyramid processing when the rough positions of the nose and the mouth are unknown.	118
5.30 Examples from Bosphorus 3D face database. They are shown as mesh. (a) Negative. (b) Positive	122
5.31 Without wearing glasses / facial hairs.	125
5.32 Wearing glasses.	125
5.33 With facial hairs, the hair covers the whole forehead.	125
5.34 Different rotation of the head.	126

5.35	The Graphical User Interface (GUI) screen used for the tapping task in our experiments. (a) 90 possible targets and their order. (b) Targets turn blue one at a time.	128
5.36	Benchmarking ATs in terms of normalized TP with respect to the mouse TP . EyeTracker (eye) [62], SmartNav (head) [36], dTDS (tongue+speech) [42], TDS (tongue) [126], HeadTracker (head) [36], LCS (lip) [44], Head Orientation (head) [120], EMG (neck muscle) [120].	131
5.37	Cursor trajectories of the tapping task with $D=534$ and $W=57$. Red dots are the selected locations. (a) Camera Mouse. (b) FM. (c) Mouse. . . .	132
5.38	TP for different target directions in the tapping task with FM, Camera Mouse and the mouse. Top: original TP . Bottom: Normalized TP by mean TP	133
5.39	Comparison of the interfaces in terms of MT versus ID and their regression models based on Fitts' law.	134
5.40	Assessment results of the questionnaire for FM and Camera Mouse. Higher score is better. Response 7 is the most favourable while response 1 the least favourable. Those comparisons that are significantly different ($p < 0.05$) are denoted with “*” on the horizontal axis.	135

Chapter 1

Introduction

The research in this thesis focuses on algorithms of motion capture and motion analysis. The motion capture method is a powerful tool for human motion analysis. We will present the motivations of this thesis in a holistic and systematic way in this chapter. The main contributions of this thesis will also be concluded. Then, we will describe the organization of the thesis.

1.1 Motivation and Objective

Figure 1.1: Markers suit in [106].

Motion capture of the human body can be extensively applied in many fields, includ-

ing surveillance, human computer interface, medicine, sports, movies, games, robotics, military and so on. Traditional motion capture systems are marker based systems. They require the subject to wear many markers on the body as shown in Figure 1.1. The markers suit looks obtrusive and is not suitable for daily use. Many special and expensive cameras are required to be placed around the subject, as shown in Figure 1.2. This scenario is more suitable for well controlled laboratory environments, but not daily living environments. These systems also need extensive time and manpower for calibration. The markers move together with the human body parts. The motion of markers can be used to calculate the motion of body parts. Thus, the captured target of these systems is the marker position.

The system demonstrates high accuracy of motion capture, and satisfies the high quality requirement for professional applications, such as driving a character in the movie. The main disadvantages of these systems are that they require particular environments, and are expensive, obtrusive and difficult to operate. These inherent disadvantages of the marker based motion capture system limit the number of the users. These systems are usually used by large companies in the movie and video game industry.

The applications of motion capture are becoming increasingly wider and deeper. The increasing requirements inspire researchers to develop novel equipment and algorithms to improve motion capture systems. For applications at home, the environment is more complex than the laboratory and the space is much smaller. The traditional marker based system is not suitable for the applications at home. For daily applications, the motion capture should be convenient to use. The markers suit is not suitable for daily applications.

The inexpensive depth camera provides promising performance for daily applications at home. A depth camera can output 3D information of objects. Some examples of depth cameras are shown in Figure 1.3. Each depth pixel in the depth image includes

Figure 1.2: An example of marker based motion capture system [106].

the depth information, which is the distance between the camera and the object. They use the active infrared LED (light-emitting diode). Although direct sunlight with strong infrared light will affect the quality of depth image, the depth image is independent of lighting illumination and objects' colour. The depth image makes the object detection more robust. Based on depth images, the state-of-the-art motion capture algorithm is based on the Randomized Decision Tree (RDT) algorithm [95]. However, the training of this algorithm has very high computational complexity. The training takes one day on a distributed implementation with a 1000 core cluster. For practical use of this algorithm, it is very important to speed up the training phase. In [95] and [16], the authors employed distributed implementations with high performance hardware.

This thesis presents an efficient algorithm, which can significantly reduce the computational complexity of the training phase while maintaining comparable accuracy in the test phase, compared to existing methods.

Figure 1.3: Depth cameras. (a) Kinect [69]. (b) SoftKinetic [97]. (c) Creative [20]. (d) A close-range depth camera [26].

One important application of motion capture is the surveillance for elderly people. The population of elderly people is increasing rapidly all over the world. The care for elderly people is thus becoming a big problem. One of the biggest risks for elderly people is the fall accident, especially for independently living elderly people. Due to the weak health of elderly people, some reports say that about 30% of elderly people (65+ years old) fall each year. If the person who fell is living alone and medical assistances do not arrive in time, more serious implications could occur following the fall. Therefore, fall

detection is an important issue in the healthcare of elderly people. Fall detection is a kind of action recognition. It is still a challenging issue up to now. Generally speaking, the fall detection system can be classified into two main categories, i.e., vision based methods [4] and non-vision based methods [18]. A very common method based on non-vision for fall detection systems is based on accelerometer devices, which are wearable devices measuring the vertical axis motion. The wearable devices methods have two major disadvantages: (1) intrude on the person's daily life; (2) the sensing devices might not be worn at all the time. The vision based method could address these disadvantages.

For the second contribution of this thesis, we focus on applying the motion capture technology by using computer vision to detect falls.

The human computer interface is another important application of motion capture technology. In modern life, the keyboard and the mouse are two standard interfaces for personal computers. However, people with disabled hands cannot use these standard interfaces. It is very important to design some Assistive Technologies (ATs) for them to use the computers. Over the world, there many hand-disabled people. Some of them, especially the tetraplegia people, cannot live independently. The life quality of these people is very low compared to the normal people. The life-long health-care for the hand-disabled people is huge. If they can use the computers easily, it can enrich their lives dramatically, and allow them to work since many jobs only require the use of the computer. One of the successful AT is the Camera Mouse [8], which is based on motion capture using an RGB camera. However, Camera Mouse requires a guardian to select a tracked point to start the interface.

Therefore, in the third contribution of this thesis, we propose a novel human computer interface based on a markerless human motion capture algorithm to address the convenience, robustness, cost and computational complexity issues.

1.2 Contributions

To address the above objectives, there are three major contributions in this thesis:

1.2.1 Fast Training Algorithm Based on a Compact Feature for Markerless Motion Capture

One challenge of the state-of-the-art markerless motion capture algorithm, i.e., RDT [95], is the high computational complexity in the training phase. In this thesis, we propose to speed up the training phase based on a compact feature. Based on the compact feature, we find that the optimized metric values in the training phase show as a smooth surface. Then, the search algorithm can be used to search for the best candidate of features. The proposed training algorithm can reduce the computational complexity dramatically. The underlying reasons of the smooth surface are analyzed in depth. The analysis shows that the speedup is linear with the reduction of the candidate number. The experimental results show that the fast training algorithm speeds up the training phase by many times without loss of accuracy.

1.2.2 Robust Fall Detection Framework Based on Markerless Motion Capture

To monitor the falls of senior people, a non-intrusive detection framework based on markerless motion capture is proposed in this thesis. The framework utilizes the low cost active infrared depth camera as the sensor. The depth camera can operate in weak light conditions, even in a dark room without light. We use the markerless human motion capture technique to detect the human fall based on depth images. An RDT algorithm is used to extract the human skeleton. The algorithm can extract a highly accurate human skeleton in a few milliseconds. By tracking some key joints' motion, the system can analyse the human motion well. A Support Vector Machine (SVM) classifier is used

to classify the fall motion based on the head joint trajectory. The head joint trajectory includes some very useful information, such as the velocity, the acceleration and the distance between the head and the floor. The SVM classifier can automatically learn the distinguishing features from the trajectory. The experiment results demonstrate that this proposed fall detection method is highly effective.

Compared to the existing methods based on RGB cameras, the proposed method can work in weak lighting, which is one of the main environments that causes falls. The proposed method can protect the subject's privacy better than the RGB cameras methods. The 3D motion information acquired by the proposed method is much more robust than that of the RGB cameras methods.

Compared to the existing methods based on depth cameras, the proposed method can track the joint motion well, rather than the mean position of the whole foreground as in [88]. The joint tracking for fall detection is more robust than that of the existing joint tracking method [85].

1.2.3 Convenient and Robust Human Computer Interface for People with Disabled Hands

For the proposed human computer interface for people with disabled hands, the depth camera is employed as the sensing device. It is independent of illumination and colour changes both from the background and on human face, which is a critical advantage over interfaces based on traditional RGB cameras. The motion of the nose is used to control the movement of the cursor. The mouth status is employed to provide interaction commands. Thanks to the fast training algorithm of RDT, we can also handily use the RDT algorithm framework to capture the interaction signals from the face motions of the hand-disabled users. Both the nose motion and the mouth status are detected by a novel RDT algorithm. In the training dataset, the ground truth labels include the nose location and the mouth status. In the test phase, the proposed approach can detect

the nose location and the mouth status accurately and efficiently. The system combines the advantages of the motion based interfaces and the face expression based interfaces. Compared to the range of the cursor movement, the range of nose motion is very small, measured in pixels in the depth image. We use a nonlinear function to map the nose motion to the cursor motion. This nonlinear function can address the issue of the small nose motion. The experimental results show that the proposed system outperforms the existing ATs.

1.3 Organization of the Thesis

This thesis is organized as below. A comprehensive background of this thesis is presented in Chapter 2. In Chapter 3, the proposed fast training algorithm of RDT based on a compact feature is presented. Chapter 4 introduces the fall detection system by using markerless motion capture technology. Chapter 5 introduces the facial position and expression mouse system by using markerless motion capture technology based on a monocular depth camera. Conclusions are drawn and future works are suggested in Chapter 6.

Chapter 2

Background

Human motion capture technology has been widely applied to surveillance, human computer interface, medicine, sports, movies, games, robotics, military and so on. The marker based motion capture systems have been employed in the industry for many years, such as by movies and video games producers. However, such type of technology is seldom used at home since they occupy large space. More and more applications require convenient, markerless and low cost motion capture systems. A monocular camera based motion capture system is more suitable for the applications at home. Some novel algorithms based on normal RGB cameras have been developed for these applications. However, their performance is not as good as marker based systems. Inspired by the applications' requirements, inexpensive depth cameras have been released in recent years, which can provide 3D information. Although some novel algorithms for depth cameras have shown significant progress for certain special applications during the past few years, there are still some open challenges limiting the applicability of methods that are based on depth cameras. This chapter will introduce the background of the motion capture algorithms. Then the background of two motion analysis frameworks, i.e., fall detection and human computer interface, is introduced.

Figure 2.1: Applications of human motion capture [40].

2.1 Motion Capture Algorithm

Figure 2.1 demonstrates some applications of motion capture techniques. Due to extensive applications of motion capture, many researchers have been dedicated to its technology development for many years. The traditional motion capture systems are based on markers and multiple cameras, as shown in Figure 2.2.

As shown in Figure 2.3, the subject is wearing a special suit. Some optical markers are attached to the key motion parts. The positions of markers are predefined to map into a 3D human skeleton model as shown in Figure 2.3. The human motion is captured via the motion of markers. Each single human joint is required to attach no less than one marker to solve the occlusion problem. Hence, the number of markers is normally much larger than the number of joints.

The movement of the subject is limited in the overlapped area of the multiple cameras'

Figure 2.2: A traditional motion capture system based on markers and multiple cameras [124].

Figure 2.3: 3D human skeleton model [124].

views, as shown in Figure 2.2. Therefore, the place occupied by the entire system is very large. The system is mostly placed in a large laboratory room. To make the object segmentation simple, the background is very clean. As shown in Figure 2.4, the ground has to be clean and its color is monochromatic. The color of markers is high contrast with the color of the ground and the suit, so that they are not confused with each other. Therefore, these motion capture systems require specially controlled laboratory

environments and professional users. They are not suitable for the applications at home.

Figure 2.4: The background and the subjects with marker suit of marker based motion capture system [40].

For robustness and accuracy requirements, each camera of this motion capture system has an array of active LEDs to illuminate the markers. Thus, the camera is special and usually very expensive. The raw data from the multiple cameras array is very large. Furthermore, the 3D information should be calculated from the 2D images provided by the multiple cameras. Thus, the computational load is very heavy. The required computational capacity of machines is very high, which makes the system even more expensive. Many researchers are trying to accelerate the running time when employing limited hardware, such as [106, 23]. In [106], each camera was assigned one CPU to process the calculation. In [23], one computer was assigned for two cameras as shown in Figure 2.5. To calculate the 3D information, the marker based motion capture system requires calibration before use.

To sum up, the disadvantages of the marker based motion capture system are outlined as follows:

- (i) Obtrusive. The subject needs to wear an obtrusive suit with markers.

Figure 2.5: The processing machine structure for the motion capture system [23].

- (ii) Not realistic environment. The system occupies a large and specially prepared place.
- (iii) Not easy to setup. Before use, the system needs special calibration.
- (iv) Expensive. The system needs complex apparatus, e.g., a special marker suit, expensive cameras and high performance computers.

To address the disadvantages of marker based motion capture, some markerless motion capture systems have been developed for the applications at home and other daily scenes. As no marker is used, markerless motion capture systems have to detect the native features on the body parts. However, it is difficult to detect such features accurately and robustly since the features are very complex and have large variations. Current systems can achieve practically useful accuracy and robustness, but only at a cost of very high computational complexity. However, real-time applications strictly require markerless motion capture algorithms to have low computational complexity. This is one of the main limitations of the markerless motion capture techniques for real-time applications.

To fill this research gap, the authors in [95] proposed an accurate and robust algorithm with low computational complexity, i.e., the RDT algorithm, to extract human body

joints from a single depth image. This approach maps the difficult pose estimation problem into a much simpler per-pixel classification problem by designing an intermediate body parts representation. The impact of the variations of pose, body shape, clothing and so on, is addressed by the highly computationally complex training of the RDT classifier. It does not employ the skeleton kinematic and temporal information, and works frame-by-frame across dramatically differing sizes and shapes of the human body, and the learned discriminative approach naturally handles human body cropped by the images frame and self-occlusions. The system runs at 200 frames per second on consumer hardware, at least one order of magnitude faster than the competitive algorithms. This algorithm, recognizing in parts, can generalize better than even the idealized skeleton model based nearest-neighbor whole-body matching until about 150k training images are reached. The authors in [95] pointed out that the results could get even better with large dataset and deeper tree. In practice, the idealized skeleton cannot be accessed as mentioned in [95].

When compared to a realizable system based on nearest neighbor chamfer whole-body matching [33], the accuracy of RDT algorithm is three times better while the speed of the RDT algorithm is also drastically faster (2 fps for the matching method, not real time). There is no other state-of-the-art markerless motion capture technique based on the single depth image that can have the comparable accuracy with such a low computational complexity as the RDT algorithm.

In this thesis, we propose a novel algorithm to speed up the training phase of the RDT algorithm. Based on the fast training algorithm, we can conveniently extend the markerless motion capture technique to two real-time motion analysis frameworks, i.e., fall detection for elderly people and human computer interface for hand-disabled people.

2.2 Fall Detection for Elderly People

The first motion analysis framework is fall detection for elderly people since one of the major risks for senior people is fall accidents, especially for those who are living alone.

The proportion of the senior people is rising quickly in most countries all over the world. In 2010, the proportion of elderly population (older than 60 years) is 11 % of the total population (759 million) all over the world [22]. In 2050, the population of senior people will be 2008 million (approximately 22 % of total population) all over the world [22]. Figure 2.6 shows the predicted population by age groups all over the world in the year of 2050.

Figure 2.6: Predicted population by age groups all over the world in the year of 2050 [22].

Many studies have indicated that fall events for senior persons are one of the most dangerous accidents at home [78]. Approximately 28-35% of elderly people (older than 65 years) fall one or more than one time per year, and this increases to 32-42% in those aged more than 70 years old [81]. Among these fall accidents, about twenty percent lead to injury, such as hip fractures, bruising and brain haemorrhage [7, 57]. About

twenty percent of all the fall events that require senior people to seek medical attention [34]. Higher than 60 percent of injury-related hospitalizations for senior people, including major soft tissue injuries and fractures [7, 57], are attributed to fall accidents.

The fall accident is one of the main reasons of deaths by injury for the elderly people. As in Figure 2.7, the fall related deaths rapidly increase along with the age growth. The most common location of fall events resulting in hospitalization of elderly people (older than 65 years) was at home, which accounts for more than 50 percent [115]. The fall accident is a heavy strain to the whole family, and it is a critical index in the family health care appraisal system [113].

Figure 2.7: Age-specific rates of fall related deaths, Australia 2004-05 [39].

The most common type of fall events resulting in hospitalization of elderly people (older than 65 years) was from falls on same level caused by sudden events, such as tripping , slipping, stumbling and so on, which accounts for nearly 50 percent [115].

The death rate is increased with the years [1]. During 2003-2007, the average rate was 41 per 100,000 populations (about 80,000 falls fatalities) recorded among American

senior people; between 2003 and 2007, it rose from 13,800 to 18,000. Between 2003 and 2007, the annual mortality rate rose from 37/100000 to 45/100000.

Besides normal injuries and even deaths, there are also other negative impacts related to fall accidents, such as mental and social impacts [105, 125]. The mental impacts include fear of fall, depression, loss of confidence, restriction of lifestyle and so on. The social impacts include healthcare expenditure, inability to live at home, indirect costs for the families and so on, as shown in Figure 2.8.

Figure 2.8: Physical, mental and social impacts of falls.

The senior people are suffering from these negative impacts. The families and society are also required to spend expensive costs. The financial burden of fall accidents all over the world is enormous. In the United States of America, European Union, the UK and Australia, the cost of fall-related accident ranges about 0.85 - 1.5 percent of

the total healthcare costs. And the total charges of healthcare related to fall accidents will certainly be enlarged in the future ageing society [38]. For example, according to conservative estimated calculation, the mean cost of all falls is more than \$1000. Only in European Community countries, there are about 100 million senior people. As described above, the fall incident rate of elderly people is 28-35 percent each year. For simplicity, this rate is approximately 30 percent. There are more than thirty million falls per year, thus the societal cost is larger than thirty billion in the European Community countries. To sum up, the fall accident of elderly people is a very important health care issue.

When an elderly person is living alone and has a fall accident, he/she may be lying on the floor for a long time without any help. This scenario mostly will lead to many serious negative problems. Researches show that getting help quickly after a fall can reduce the hospitalization risk by 26 % and the death risk by over 80 % for elderly people [79].

Therefore, a fall accident detection system, which can automatically detect the fall accident and call for help, is very important for elderly people, especially for those living alone.

2.2.1 Definition of a Fall

Some words, i.e., “unplanned”, “involuntary”, “accidental”, “no intention”, “no intention”, “sudden”, “drop”, “hit” or “collide” are used to describe an external point of view, or the description language of the fall. The person who falls can use different words, such as tripping or slipping.

Although it is hard to reach a consensus definition of a “fall”, there are many existing fall definitions in the literature [15, 68]. Investigators have adapted to the relationship of these consensus definitions of a particular target intervention or population. A simple and practical fall definition is especially important for the study on fall research for the elderly people.

For effective research on “fall” detection, it is very important to make a complete definition of the specification on fall. Researchers recognize the need to have a simple and practical fall definition, aiming to clearly determine which activities can be included, and to distinguish between different types of the Activity of Daily Living (ADL) and fall accident.

“An unexpected event in which the participant comes to rest on the ground, floor, or lower level” is a consensus statement definition of a fall [53]. It was proposed by Prevention of Falls Network Europe(ProFaNE). This definition is the same as World Health Organization’s (WHO) one: “A fall is an event which results in a person coming to rest inadvertently on the ground or floor or other lower level.” There are many other very similar definitions about the falls. With regard to these definitions, we can detect the relationship between the person and the floor to detect the “fall”, which is a strong feature of a fall.

2.3 Human Computer Interface for Hand-Disabled People

The second motion analysis framework is a human computer interface for hand-disabled people, who cannot use standard interfaces of computer.

The hand-disable persons, especially those with tetraplegia, cannot use their hands as normal people. The spinal cord injury is one of the major reasons to the persons with tetraplegia. As shown in [107], in the USA, there are more than 10,000 new spinal cord injury cases every year, which is about a new spinal cord injury occurs per 44 minutes; the total number of people with spinal cord injuries in the USA is more than 260,000; the motor vehicle crash is the No.1 cause of spinal cord injury; the average age at injury is 40.7 years old. 56% of spinal cord injury results in tetraplegia, with C5 being the most common injury level [110].

Figure 2.9: Muscle representation for the C3 and C4 spinal cord injury [98].

The high tetraplegia (C1-C4) by spinal cord injuries have extensive paralysis of arm muscles. As shown in Figure 2.9, there are few muscles under voluntary control of the people with C3 or C4 spinal cord injury.

The lifetime costs of people with spinal cord injuries are very high. As shown in Table 2.1, the lifetime costs of the high tetraplegia (C1-C4) are about 2.7 million.

Table 2.1: Lifetime Costs of living with spinal cord injuries (injured at 25) in the United States [107].

Level	Costs
High Tetraplegia (C1-C4)	\$2,711,708
Low Tetraplegia (C5-C8)	\$1,847,631
Paraplegia	\$1,108,930
Functional Motor Incomplete at any level	\$798,391

Nowadays, personal computers play a very important role in modern life. However, persons with tetraplegia, suffering from spinal cord injuries (SCI), cerebral palsy, traumatic brain injury, neurological injury or stroke, are very difficult to use standard interfaces of the personal computers, e.g., the keyboard and the mouse, during their

rehabilitation and everyday life activities. Therefore, it is highly necessary to develop Assistive Technologies (ATs) for persons with tetraplegia.

Chapter 3

Markerless Motion Capture Algorithm

This chapter reviews the state-of-the-art methods for markerless motion capture, and then proposes an approach to address the challenge of training phase of RDT based algorithm, which can speed up the training phase by using a compact feature.

3.1 Related Work

Some markerless motion capture systems have been developed for the applications at home and other daily scenes. These systems can be classified into 2 categories:

- (i) Based on the RGB camera.
- (ii) Based on the depth camera.

The state-of-the-art of these systems is elaborated as below.

3.1.1 State-of-the-Art of Markerless Motion Capture Based on the RGB Camera

Due to the development of the motion capture algorithms, the human motion can be captured even though without markers on the human body. In [21], the markerless

motion capture system employed a laser scanned mesh model and multiple video cameras to capture the human motion. Its framework is shown in Figure 3.1.

Figure 3.1: Overview of a markerless motion capture framework in [21]. The input is a laser-scan mesh model and some multi-view video streams. The method deforms the mesh model as its real-world counterpart in the video streams.

The marker based capture system relies on the kinematic skeleton. In contrast, the scene representation of [21] is a deformable high-quality mesh of a human body. As shown in Figure 3.1, the input includes a static laser-scanned triangle mesh of the captured subject. The multi-view video sequences show that the person moves arbitrarily. The laser scan is aligned to the pose of the person in the first step. In step A, an iterative 3D flow-based deformation approach is employed to extract the motion information of each vertex from the multi-view images. In step B, all vertices are made to follow the movement of the moving marker vertices. Step B implicitly enforces structural integrity of the human mesh model. The workflows of tracking steps A and B are similar, as shown in Figure 3.2.

This approach requires a laser scanner and eight synchronized video cameras. Due to its high computational complexity, it is not suitable for real time applications.

To make the system more compact, in [52], the authors used a 3D articulated model formed by eleven segments imitating main parts of the human body to fit the images from four calibrated and synchronized cameras. The framework is shown in Figure 3.3.

Figure 3.2: The workflows of tracking steps A and B [21]. Step A is denoted as Arabic numerals. Step B is denoted as Roman numerals.

Figure 3.3: The framework of the system in [52].

For real time performance, a GPU accelerated tracking of full body motion was proposed in [131] based on multi-layer search, as shown in Figure 3.4. However, this system is still based on multiple cameras to capture the motion.

To make the system more compact, in [59], the researchers proposed a silhouette-

Figure 3.4: Overview of the GPU accelerated motion capture system in [131].

based method to estimate 2D human pose by using a single video camera. A human skeleton model is pre-defined as the prior information. The silhouette is extracted from an image by the Gaussian Mixture Background Model (GMM).

The Distance Transform (DT) and Principal Component Analysis (PCA) are employed to locate the base point of the human skeleton from the 2D silhouette, see Figure 3.5.

Figure 3.5: Separation of the human torso and detection of the base point [59]. (a) The 2D silhouette of the human body. (b) The result of DT. (c) The results of the torso and the base point. (d) The results of the torso and the base point in the color image.

Then an iterative process adjusts the human skeleton model parameters, see Figure 3.6. This system is simple and can run in real time. However, the output skeleton

information is in 2D in this system. The robustness is limited by the performance of the 2D silhouette. For example, if one arm is inside the body’s silhouette, errors will happen due to uncertainty. This problem can be described as self-occlusion. Another problem is the depth ambiguity (or silhouette ambiguity), as shown in Figure 3.7. Two different 3D poses have the same 2D silhouette when using a monocular camera as shown in Figure 3.7,.

Figure 3.6: The human skeleton model fitting iteration in [59].

Figure 3.7: Example of the depth ambiguity [41]. One 2D silhouette can have more than one corresponding 3D human poses.

One of the major limitations of the traditional RGB camera is that it only outputs

2D images, which lack 3D information. As shown in the following sections, the depth camera can address this problem.

3.1.2 State-of-the-Art of Markerless Motion Capture Based on the Depth Camera

With the rapid development of imaging hardware, the quality of the depth camera is becoming better and better while its price is going lower and lower. Each pixel of the depth image includes the depth information, i.e., the distance between the object and the camera, and can be recognized as a 3D point. It can address the depth ambiguity problem, which is a serious problem of the RGB camera. Some novel approaches have been proposed to capture the 3D human motion based on the depth camera in recent years.

The authors in [50] proposed a tracking system for 3D tracking of human body movements based on a depth camera and an RGB camera. An articulated 3D human body model is used to fit the 3D point cloud by using the iterative closest point (ICP) algorithm. All depth images and RGB images are processed with the same model and fitted in 3D model space. The overview of the system is shown in Figure 3.8. Although this system runs in real time, i.e., 10-14 fps, the resolution of depth image is low (174×144).

Figure 3.8: Overview of the system in [50].

The authors in [35] proposed a combined framework of silhouette and depth information to track the motion of a human. This system also uses a depth camera and a RGB camera. A 3D body model, as shown in Figure 3.9, is used to fit the depth pixels. The configuration of the 3D body model should be prepared manually. The initial pose has to be positioned on the image manually by the user. The system can run in real time in tracking mode.

Figure 3.9: The 3D body model used in [35].

The research in [132] estimates human pose from detected features by using a model-based Cartesian control theoretic approach based on depth images. The overview of this system is shown in Figure 3.10. Some features and a 3D model is used, as shown in Figure 3.11. A probabilistic inference algorithm is employed to detect the features that represent positions of anatomical landmarks. Based on a simple kinematic model with constraints, i.e., joint limit avoidance and self-penetration avoidance, are enforced by the tracking control framework. The system with 17 degrees of the freedom upper body human model can run at 10 fps.

Figure 3.10: System diagram of the entire pipeline in [132].

Figure 3.11: Features and head-neck-torso (H-N-T) template used in [132].

The authors in [56] employed a model-based annealing particle filter (APF) approach to retrieve and track the pose of a single subject in video sequences. GPU is used to accelerate the system with 300 points to run at 2.5 fps. Figure 3.12 shows results with the matched body overlaid.

Figure 3.12: Different poses with the matched body prototype overlaid results in [56].

The authors in [5] proposed a data-driven hybrid strategy that combines local optimization with global retrieval techniques to estimation full body pose based on a single depth image stream. The overview of system is shown in Figure 3.13. The first five geodesic extrema (end effectors, corresponding to head, hands and feet) are extracted as a 15-dimensional feature vector to look up the best matching pose. The pose database is searched by the kd-tree of the database using the recovered geodesic extrema as the query input. This algorithm is very fast, i.e., 60-100 fps. However, there are many practical in this system. The end effectors should be revealed by the geodesic extrema, otherwise the system becomes unstable. The subject should face the camera with rotations around the vertical axis within ± 45 degrees to avoid missing detection of limbs.

Figure 3.13: Overview of the full body estimation system in [5].

In [104], the authors proposed a head and hand tracking algorithm from a depth image stream. An elliptical model of head, as shown in Figure 3.14, is employed to estimate the head position by matching. Hands are detected in a bounding box attached to the estimated head position, as shown in Figure 3.15. This system is only suitable for users who are facing the camera in a standing-up or sitting pose. The tracking mode of this system can not work in real time without down sampling, i.e., with 640×480 resolution. After down sampling, it runs in nine fps with down sampling to 320×240 , and 68 fps with down sampling to 160×120 .

Figure 3.14: Left: a representation of a human head. Middle: the foreground of the left figure. Right: a graphical example of the elliptical mask used in [104].

Figure 3.15: The bounding box of two hands used in [104].

In [84], a real time identification and localization of body parts was proposed based on a single depth image. This approach does not need the 3D human model. The authors claimed that some interest points coincide with salient points of the body, which can be classified as, e.g., head, hand or foot using local shape descriptors. These interest points are based on identification of geodesic extrema on the surface mesh. A boosted

patch classifier was used to classify these interest points. Their experiments show that the interest points are significantly better in detecting body parts in depth images than the sliding-window based detectors. The method is employed to initialize the tracking algorithm in [31].

In [31], the authors proposed a filtering algorithm to track human pose by using a monocular depth camera. As shown in Figure 3.16, a specific 3D articulated body model with vertex skin is used. After initialization, the approach models the system as a Dynamic Bayesian Network (DBN). This approach is a data-driven evidence method (the input data drives the 3D human body model). Figure 3.17 shows an example of the typical tracking situation.

Figure 3.16: The 3D articulated body model with vertex skin used in [31].

This approach can provide highly accurate results at about five frames per second by using GPU acceleration. The resolution of the depth images in that work is down sampled to 128×128 pixels for GPU acceleration. Although the frame rate is not high enough for real-time performance, this method shows that the depth camera provides a promising

Figure 3.17: A sample of the typical tracking situation in [31].

applicability of motion capture applications that require real time performance.

A much faster motion capture algorithm is proposed in [95]. This technique does not need a 3D articulated body model in the test phase. The system overview is shown in Figure 3.18.

Figure 3.18: Overview of the motion capture system based on an RDT algorithm in [95].

In this approach, a much simpler per-pixel classification replaces the difficult pose detection by an intermediate human body parts classification. The Randomized Decision Tree (RDT) algorithm is employed to classify each foreground pixel of the test depth

image. The schematic of RDT is shown in Figure 3.19. To address the variation of poses, body shapes, clothing, etc., a large and highly varied training dataset is used. The properties of depth images, e.g., independence of color and lighting illumination, allow straightforwardly building up large synthesized realistic depth image dataset for the training of the RDT algorithm. In the test phase, the system can run at about 200 fps on the Xbox GPU with high accuracy result of 3D locations of human body joints from a single depth image. This approach can be classified into the modelless category. In contrast, most of the other motion capture approaches are in the model-based category, which requires a human model during the test phase. One major drawback of the RDT approach is the high computational complexity for the training. The training takes one day on a distributed implementation with a 1000 core cluster. This drawback limits its practical applicability since it is too difficult to implement the training of RDT algorithm.

Figure 3.19: Randomized decision tree in [95]. Black circles: split nodes. Red circles: leaves. A particular input may be taken the blue arrow paths.

To combine the advantages of the model-based method and modelless method, in [119], the authors used the RDT algorithm of [95] to initialize and recover (from unsuccessful tracking) a 3D articulated human body model, and then the human body model was registered by a Maximum A Posteriori (MAP) framework. The system overview is shown

in Figure 3.20. The 3D articulated human body model is shown in Figure 3.21. The predefined calibration pose is shown in Figure 3.22. The MAP framework integrates silhouette information, depth data, full-body geometry, occlusion reasoning and temporal pose priors. Although the computational complexity is much higher than that of the RDT algorithm, this system is more accurate when the self-occlusion happens. And the whole system can still run in real time since the frame rate is about 44 fps on a parallel implementation in a GPU. However, this system often cannot provide accurate results when the movement is fast. Trapping into the local minima is another problem of this system.

Figure 3.20: Overview of the motion capture system combined the RDT algorithm and the 3D human model in [119].

To combine the advantages of the RGB camera, the depth camera, the model based method and the modelless method, in [16], the authors proposed a pose estimation approach based on RGB-D cameras. The system flow diagram is shown in Figure 3.23. The RDT algorithm of [95] is employed to classify each pixel over the entire image. A rough skeleton is extracted by a clustering algorithm over the classified pixels. Combined with a kinematical model and the synchronized RGB image, which helps to remove the

Figure 3.21: The 3D human skeleton model in [119].

Figure 3.22: The calibration human pose of the system in [119].

background, the human skeleton is refined. The run-time system flow diagram is shown in Figure 3.24. It also has real time performance since the frame rate is about 17-20 fps on a CPU-GPU combination implementation. One constraint of this system is the high computational complexity of the training phase of RDT, similar to the system of [95]. It takes 7 days to train a tree on a sixteen node cluster, in which each node is an eight-core Xeon X5570 system with 48 GB RAM.

Figure 3.23: Overview of the system in [16].

Figure 3.24: The iteration is used in the system of [16].

3.2 RDT Algorithm

The motion capture system based on the depth camera can offer a convenient and non-invasive system setup. Compared to RGB cameras, there are some significant advantages of the depth images when they are used for the motion capture. The depth images

provide 3D information, rather than measures of RGB color or intensity from traditional cameras. They capture the geometry surface of the object, which is independent of color, texture and lighting illumination. They are usually insensitive to changes in shadows and various other factors in RGB images. The depth images resolve the silhouette ambiguity of human poses, which is one of the serious problems in most monocular RGB camera systems. The properties of depth images allow synthesizing many realistic and generative depth images of persons for training. One disadvantage of the depth image is that it will be affected by similar shaped objects, such as a white plaster model. The algorithm based on a single depth image cannot distinguish a white plaster model and a real human.

Although the model based methods have higher accuracy than modelless based methods (RDT algorithms) under some specific conditions, the model based methods require good initialization/re-initialization, which can be provided by the RDT algorithm. The model based methods are prone to failure when the motion of the subject is fast, as in shown in [119]. The RDT algorithm is much faster than the model based methods, and does not need calibration and initialization. However, one of the major disadvantages of the RDT algorithm is the high computational complexity of the training phase. In [95], to make the training practical, the authors discarded many depth images with similar human poses to reduce the training dataset. In addition, the amount of trained pixels for each depth image was down sampled.

We introduce the existing RDT algorithm in detail as follows.

3.2.1 The Feature of RDT

In [58], the recognized feature is based on the difference of intensities of two pixels taken in the neighbourhood of a key point. This feature was further developed in the depth image by Shotton et al. in [95]. They employed a simple depth comparison feature instead of the intensity comparison feature, resulting in a wonderful success.

Only one or two comparison features are not eligible for discriminating objects, such as discriminating the different parts of the human body. However, an RDT based on these comparison features is sufficient to detect objects. It can handle the noise of the depth image and can even work by using the 2D silhouette [95]. The formulation of comparison feature in [95] can be described as:

$$f((x_0, y_0)|(\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2)) = z((x_0, y_0) + \frac{(\Delta x_1, \Delta y_1)}{z(x_0, y_0)}) - z((x_0, y_0) + \frac{(\Delta x_2, \Delta y_2)}{z(x_0, y_0)}), \quad (\text{Eq. 3.1})$$

where (x_0, y_0) is the test pixel of the depth image, $(\Delta x, \Delta y)$ represents the location offset according to the location (x_0, y_0) of the test pixel, $(\Delta x_1, \Delta y_1)$ and $(\Delta x_2, \Delta y_2)$ are two different offset values, the pixel (x, y) depth value is $z(x, y)$. The location offset is normalized by $1/z(x, y)$, so that the feature can resolve the ambiguity in depth variation. As shown in Eq. 3.1, this feature is 3D translation invariant, and is highly efficient for computing. It does not need pre-processing. Each feature needs only a few arithmetic operations.

The depth information around the test pixel describes the geometric surface around it. Eq. 3.1 is a feature of the geometric surface of the object. If the geometric surfaces of two objects are different, the RDT algorithm with the discriminated feature, i.e., Eq. 3.1, has the potential to discriminate these two objects.

Figure 3.25 shows a randomized decision tree consisting of split and leaf nodes. A randomized decision forest includes several randomized decision trees. In many applications, the task of multi-class classifiers can be implemented with high efficiency and high speed by RDT [95, 58]. In each tree, each split node n has a parameter $p_n = ((\Delta x_{1,n}, \Delta y_{1,n}), (\Delta x_{2,n}, \Delta y_{2,n}), \tau_n)$. $(\Delta x_{1,n}, \Delta y_{1,n})$ and $(\Delta x_{2,n}, \Delta y_{2,n})$ are two different offset values. τ_n is a scalar threshold for comparing with the feature value of the test pixel. The evaluating function of comparison is

$$E((x_0, y_0); p_n) = B(f((x_0, y_0)|(\Delta x_{1,n}, \Delta y_{1,n}), (\Delta x_{2,n}, \Delta y_{2,n})) - \tau_n), \quad (\text{Eq. 3.2})$$

Figure 3.25: Randomized decision tree in [95].

where $B(\cdot)$ is a binary function. When the value of $(f((x_0, y_0)|(\Delta x_{1,n}, \Delta y_{1,n}), (\Delta x_{2,n}, \Delta y_{2,n})) - \tau_n)$ is greater than or equal to 0, there will be $B(f((x_0, y_0)|(\Delta x_{1,n}, \Delta y_{1,n}), (\Delta x_{2,n}, \Delta y_{2,n})) - \tau_n) = 1$; otherwise $B(f((x_0, y_0)|(\Delta x_{1,n}, \Delta y_{1,n}), (\Delta x_{2,n}, \Delta y_{2,n})) - \tau_n) = 0$. When $E((x_0, y_0); p_n)$ is one, the test pixel is split to the left branch child of node n . When $E((x_0, y_0); p_n)$ is zero, the test pixel is split to the right branch child of node n . The operation repeats, and stops when it meets the leaf node l . There is some classification information in the leaf node l , such as the probability of body parts. To classify a pixel (x_0, y_0) , the pixel starts at the root and repeats evaluation of Eq. 3.2, which decides going to left or right with regard to the value of E , until reaching the leaf node.

After classifying each pixel, the joint position can be predicted by body part cluster [95]. In the body part cluster method, the joint position is presented by the mean of the same class pixels.

Figure 3.26 shows a classification result of the human body. Each color in Figure 3.26 stands for the respective part of the human body with the highest probability.

3.26.a:

3.26.b:

Figure 3.26: The label map after the classification of RDT, each color stands for the respective part of the human body with the highest probability [95, 119]. (a) A map in [119]. (b) A map in [119].

3.2.2 Aggregating Predictions

The result of RDT algorithm is the classification of each foreground pixel. Then the joint position is found by a mean-shift method. The mean-shift is based on a Gaussian kernel and a weighting factor. For the mean-shift, the density estimator [95] of each body part c is defined as

$$J_c(L) \propto \sum_{i=1}^N w_{ic} \exp(-\|\frac{L - L_i}{b_c}\|^2), \quad (\text{Eq. 3.3})$$

where c represents the label of a human body part, L represents the 3D location in the 3D real world, w_{ic} represents a pixel weighting, the total amount of test depth pixels in a given test depth image I is N , L_i represents the test pixel (x_i, y_i) 3D location, and b_c represents a bandwidth for body part c . w_{ic} includes two factors: (1) the probability P from the RDT algorithm; (2) the size of the world surface dependent on the depth value z of the test pixel (x_i, y_i) . The formulation of w_{ic} is

$$w_{ic} = P(c|(x_i, y_i), I) \cdot z(x_i, y_i)^2. \quad (\text{Eq. 3.4})$$

After using the mean-shift method, Figure 3.27 shows the joint positions of the human body of Figure 3.26.a. This mean-shift method result is more accurate than the result of the global centre of the same body part since there are some outlying pixels as shown in Figure 3.26.

3.2.3 Training

The accuracy and robustness of the RDT algorithm strongly depends on the feature on each node of RDT. In order to obtain an optimised parameter p_n of each split node n of RDT, the computational complexity was extremely high in [95, 16].

The tree is trained by using the smallest Shannon entropy to split each node. At each node, a weak learner parameter $p((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau) ((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2) \in$

Figure 3.27: Predicted joints' positions by mean-shift method of the human body of Figure 3.26.a [95].

$(\Delta X, \Delta Y)$ are the pixel offset values, $\tau \in T$ is the threshold value in Eq. 3.2) induces a partition of input example set $Q = \{I, (X, Y)\}$ into left and right subsets by Eq. 3.2:

$$Q_L((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau) = \{E((x_0, y_0); p((\Delta x, \Delta y), \tau)) = 1\}, \quad (\text{Eq. 3.5})$$

$$Q_R((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau) = \{E((x_0, y_0); p((\Delta x, \Delta y), \tau)) = 0\}. \quad (\text{Eq. 3.6})$$

For each offset $(\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2)$, compute the τ giving the smallest Shannon entropy:

$$\tau^* = \arg \min_{\tau \in T} S(Q((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau)), \quad (\text{Eq. 3.7})$$

$$S(Q((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau)) = \sum_{sub \in L, R} \frac{|Q_{sub}((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau)|}{Q} H(Q_{sub}((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau)), \quad (\text{Eq. 3.8})$$

where $H(Q)$ is the Shannon entropy (computed on the probability of body part labels) of set Q . $S(Q)$ is the sum of Shannon entropy.

Eq. 3.5, Eq. 3.6 should be tested by the entirety of the set $Q = \{I, (X, Y)\}$ of input examples in a node, and this operation takes a long training time.

The parameter p with the smallest Shannon entropy in all candidate offsets and thresholds is

$$\begin{aligned}
 p((\Delta x_1^*, \Delta y_2^*), (\Delta x_2^*, \Delta y_2^*), \tau^*) &= \arg \min_{(\Delta x, \Delta y) \in (\Delta X, \Delta Y), \tau \in T} S(Q((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau)) \\
 &= \arg \min_{(\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2) \in (\Delta X, \Delta Y)} \left\{ \min_{\tau \in T} S(Q((\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \tau)) \right\}.
 \end{aligned}
 \tag{Eq. 3.9}$$

If the depth of the tree is not too large, the training algorithm recurs for the left and right child nodes with example subsets Q_L and Q_R , respectively, according to $p((\Delta x_1^*, \Delta y_2^*), (\Delta x_2^*, \Delta y_2^*), \tau^*)$.

The amount of the total possible pairs of candidate offsets is very large. There are $(2M + 1)^4$ pairs of candidate offsets when the range of $\Delta x, \Delta y$ is M , i.e., $\Delta x, \Delta y \in [-M, M]$. When the range is increased, more spatial context can be used for decisions of the RDT algorithm. If $M = 100$, the amount of the total possible pairs of candidate offsets is about 1.6×10^9 . [Eq. 3.5](#), [Eq. 3.6](#) should be tested by the entirety of the set $Q = \{I, (X, Y)\}$ of input examples in a node, and this operation takes a long training time. A larger amount of candidate offset pairs requires a longer training time.

Both in [\[95\]](#) and [\[16\]](#), the amount of candidate offset pairs was down sampled to 2000, which is extremely small compared to the amount of the total possible candidate offset pairs. A distributed implementation was employed to make the training practical both in [\[95\]](#) and [\[16\]](#). The training time in [\[95\]](#) was about one day to train three trees with 20 levels from 1 million images (not all the images were used for training) on a 1000 core cluster. The training time in [\[16\]](#) was seven days to train one tree with 16 levels from 80k images on a 16 node cluster. Each node was an 8 cores Xeon X5570 system with 48 GB RAM. Two GTX550ti GPUs were in a single node. From the training in [\[95, 16\]](#), we can note that the computational complexity of RDT algorithm for motion capture is very high. This disadvantage will limit its applicability dramatically. Thus, reducing the training time is very important for practical applications and implementations.

3.3 Fast Training Algorithm Based on a Compact Feature

3.3.1 The Compact Feature

The depth information around the test pixel reflects the geometric surface around it. The geometric surface around the test pixel can be described well enough by the depth differences between the neighbour pixels and the test pixel. Thus, the following feature can be used to describe the same geometric surface as Eq. 3.1 to recognize the test pixel:

$$f((x_0, y_0)|(\Delta x, \Delta y)) = z(x_0, y_0) - z((x_0, y_0) + \frac{(\Delta x, \Delta y)}{z(x_0, y_0)}). \quad (\text{Eq. 3.10})$$

Compared with Eq. 3.1, this feature is with a higher computational efficiency: there is only one division, one addition and one subtraction; it only looks up two depth pixels. The amount of the total possible candidate offsets is much smaller than that of Eq. 3.1. It is $(2M+1)^2$ candidate offsets when the range of $\Delta x, \Delta y$ is M , i.e., $\Delta x, \Delta y \in [-M, M]$. If $M = 100$, the amount of the total possible candidate offsets is about 4×10^4 , which is only one out of 4×10^4 of the amount of Eq. 3.1.

Figure 3.28 is an example that shows the representation of Eq. 3.1 and Eq. 3.10 to describe a certain shape object. There are nine features for Eq. 3.1, i.e., $f_{0-1}, \dots, f_{2-2}$. But there are only two for Eq. 3.10, which can completely determine the curve shape. Compared to Eq. 3.10, there are many redundancies in Eq. 3.1. In fact, any feature for Eq. 3.1 can be represented by Eq. 3.10, such as

$$\begin{aligned} f_{1-2} &= z(D_1) - z(D_2) \\ &= -(z(D_0) - z(D_1)) + (z(D_0) - z(D_2)) \\ &= -f_{0-1} + f_{0-2}. \end{aligned} \quad (\text{Eq. 3.11})$$

Therefore, there is no essential difference in the amount of information carried in Eq. 3.10 and Eq. 3.1. However, Eq. 3.1 has many redundancies compared to Eq. 3.10.

Equation 3.1

$$\begin{aligned}
 f_{0-0} &= z(D_0) - z(D_0) \\
 f_{0-1} &= z(D_0) - z(D_1) \\
 f_{0-2} &= z(D_0) - z(D_2) \\
 f_{1-0} &= z(D_1) - z(D_0) \\
 f_{1-1} &= z(D_1) - z(D_1) \\
 f_{1-2} &= z(D_1) - z(D_2) \\
 f_{2-0} &= z(D_2) - z(D_0) \\
 f_{2-1} &= z(D_2) - z(D_1) \\
 f_{2-2} &= z(D_2) - z(D_2)
 \end{aligned}$$

Equation 3.10

$$\begin{aligned}
 f_{0-1} &= z(D_0) - z(D_1) \\
 f_{0-2} &= z(D_0) - z(D_2)
 \end{aligned}$$

Figure 3.28: An example to show the efficiency of Eq. 3.1 and Eq. 3.10 to describe an object with a certain shape. The normalization operation is included in $z(\bullet)$. The object has only three points. Eq. 3.1 has many more features than Eq. 3.10, but both of them can completely determine the same object. Many features for Eq. 3.1 are redundant.

The redundancies in Eq. 3.1 dramatically increase with the number of neighbour pixels. When there are N neighbour pixels (including the test pixel), the number of instances of Eq. 3.1 is N^2 while the number of instances of Eq. 3.10 is $N - 1 \approx N$. The ratio of the numbers of Eq. 3.1 and Eq. 3.10 is $N^2/N = N$.

Figure 3.29: Randomized decision tree based on a compact feature.

Figure 3.29 shows a randomized decision tree consisting of split and leaf nodes based

on [Eq. 3.10](#). In each tree, each split node n has a parameter $p_n = ((\Delta x_n, \Delta y_n), \tau_n)$. $(\Delta x_n, \Delta y_n)$ is the offset value. τ_n is a scalar threshold for comparing with the feature value of the test pixel. The evaluating function of comparison is

$$E((x_0, y_0); p_n) = B(f((x_0, y_0)|(\Delta x_n, \Delta y_n)) - \tau_n), \quad (\text{Eq. 3.12})$$

where $B(\cdot)$ is a binary function. When the value of $(f((x_0, y_0)|(\Delta x_n, \Delta y_n)) - \tau_n)$ is greater than or equal to 0, there will be $B(f((x_0, y_0)|(\Delta x_n, \Delta y_n)) - \tau_n) = 1$; otherwise $B(f((x_0, y_0)|(\Delta x_n, \Delta y_n)) - \tau_n) = 0$. When $E((x_0, y_0); p_n)$ is one, the test pixel is split to the left branch child of node n . When $E((x_0, y_0); p_n)$ is zero, the test pixel is split to the right branch child of node n . The operation repeats, and stops when it meets the leaf node l . There is some classification information in the leaf node l , such as the probability of body parts. To classify a pixel (x_0, y_0) , the pixel starts at the root and repeats the evaluation of [Eq. 3.12](#), going left or right respecting to the value of E , until reaching the leaf node. The classification of RDT is similar with [\[95\]](#).

After classifying each pixel, the joint position can be predicted by body part cluster as [\[95\]](#).

3.3.2 Fast Training Algorithm

Based on [Eq. 3.10](#) we propose a fast training algorithm which can reduce the number of candidate offsets significantly in the training phase.

At each node, a weak learner parameter $p((\Delta x, \Delta y), \tau)$ ($(\Delta x, \Delta y) \in (\Delta X, \Delta Y)$ is the pixel offset, $\tau \in T$ is the threshold value in [Eq. 3.12](#)) induces a partition of input example set $Q = \{I, (X, Y)\}$ into left and right subsets by [Eq. 3.12](#):

$$Q_L((\Delta x, \Delta y), \tau) = \{E((x_0, y_0); p((\Delta x, \Delta y), \tau)) = 1\}, \quad (\text{Eq. 3.13})$$

$$Q_R((\Delta x, \Delta y), \tau) = \{E((x_0, y_0); p((\Delta x, \Delta y), \tau)) = 0\}. \quad (\text{Eq. 3.14})$$

3.30.a:

3.30.b:

Figure 3.30: Entropy map and 3D view. (a) Map of entropy. (b) 3D view of the map.

For each offset $(\Delta x, \Delta y)$, compute the τ giving the smallest Shannon entropy:

$$\tau^* = \arg \min_{\tau \in T} S(Q((\Delta x, \Delta y), \tau)), \quad (\text{Eq. 3.15})$$

$$S(Q((\Delta x, \Delta y), \tau)) = \sum_{sub \in L, R} \frac{|Q_{sub}((\Delta x, \Delta y), \tau)|}{Q} H(Q_{sub}((\Delta x, \Delta y), \tau)), \quad (\text{Eq. 3.16})$$

where $H(Q)$ is the Shannon entropy (computed on the probability of body part labels) of set Q . $S(Q)$ is the sum of Shannon entropy.

Figure 3.30.a is a map in a node by drawing Shannon entropy (given by $p((\Delta x, \Delta y), \tau^*)$) on the corresponding location $(\Delta x, \Delta y)$. Figure 3.30.b is a mesh view of Figure 3.30.a. From Figure 3.30.b, it can be noted that the Shannon entropy surface is smooth. The smallest Shannon entropy in this surface can be efficiently found by using search algorithms. Thus, just a few offsets need to be trained by Eq. 3.13, Eq. 3.14 for each node. Therefore, using Eq. 3.10 and a suitable search algorithm can dramatically reduce the computational complexity, resulting in faster training or implementation with lower price hardware.

There are two major underlying reasons for the smooth entropy map:

- (i) In the natural world, the 3D surface around the test pixel is continuous, and it changes smoothly when the current test pixel shifts to a nearby one. As shown in Figure 3.31, test pixels are two adjacent pixels. The difference between the surfaces of these two pixels is very small. Therefore, the difference from a test pixel (stand for a 3D surface patch centred at the test pixel) to another one changes smoothly.
- (ii) The capacity of one feature to distinguish human body parts is very weak. As shown in Figure 3.32.a, the difference in feature values between these adjacent offset pixels is very small. In Figure 3.32.b, a same location offset is shown. The difference in feature values between two different surfaces is also very small. Therefore, the feature value changes smoothly.

Figure 3.31: The difference between the surfaces of two adjacent pixels is very small. Red points are the test pixels.

To distinguish human body parts, the capacities of adjacent offset pixels are very similar, due to the fact that the test pixel as well as the feature value changes smoothly. The entropy value reflects the capacity to distinguish human body parts. Thus, the entropy values of adjacent offset pixels are very similar.

The parameter p with the smallest Shannon entropy in all candidate offsets and

3.32.a:

3.32.b:

Figure 3.32: The feature value is changing smoothly. (a) Two adjacent offset pixels (green points). The difference in feature values between these two adjacent offset pixels is very small. (b) A same offset (green points) in two objects. The difference between feature values of these two objects is very small.

thresholds is

$$\begin{aligned}
 p((\Delta x^*, \Delta y^*), \tau^*) &= \arg \min_{(\Delta x, \Delta y) \in (\Delta X, \Delta Y), \tau \in T} S(Q((\Delta x, \Delta y), \tau)) \\
 &= \arg \min_{(\Delta x, \Delta y) \in (\Delta X, \Delta Y)} \{ \min_{\tau \in T} S(Q((\Delta x, \Delta y), \tau)) \}.
 \end{aligned} \tag{Eq. 3.17}$$

If the depth of the tree is not too large, the training algorithm recurs for the left and right child nodes with example subsets $Q_L((\Delta x^*, \Delta y^*), \tau^*)$ and $Q_R((\Delta x^*, \Delta y^*), \tau^*)$, respectively, according to $p((\Delta x^*, \Delta y^*), \tau^*)$.

The computational complexity and speedup of the training phase is not difficult to evaluate. A given trained pixel, at node n , needs to calculate the feature Eq. 3.1 (Eq. 3.10) N_{can} (number of candidates) times, evaluate Eq. 3.2 (Eq. 3.12) $N_{can} \times N_\tau$ (N_τ is the number of different τ values for each candidate) times, and calculate the probability $N_{can} \times N_\tau$ times. The number of levels of the tree is N_{level} . In total, when a pixel goes from the root to the leaf, the computational complexity is

$$\begin{aligned}
 f(\text{one pixel}) &= (N_{can}f(\text{Eq. 3.1}) + N_{can}N_\tau f(\text{Eq. 3.2}) + N_{can}N_\tau f(\text{probability}))N_{level} \\
 &= N_{can}(f(\text{Eq. 3.1}) + N_\tau f(\text{Eq. 3.2}) + N_\tau f(\text{probability}))N_{level}
 \end{aligned} \tag{Eq. 3.18}$$

where, $f(\bullet)$ is the computational complexity of “ $\bullet$ ”. Although $f(\text{Eq. 3.1})$ has a higher computational complexity than $f(\text{Eq. 3.10})$, we assume that they have the same complexity to simplify the analysis, and Eq. 3.18 can also describe computational complexity of the training of feature Eq. 3.10. The computational complexity of a given trained pixel is constant over the number of trained pixels. Therefore, the total computational complexity for training increases linearly with the number of trained pixels, i.e., $O(n)$ for the *O-notation*, and increases linearly with N_{can} , i.e., $O(n)$. When N_{can} is reduced by n times, the total computational complexity is reduced by n times, i.e., $O(n)$. The amount of speedup is constant when the number of trained pixels is increasing, i.e., $O(1)$.

3.3.3 Experimental results

Table 3.1 demonstrates the performance of the algorithms based on [95] and ours. Based on Eq. 3.10 and a search algorithm, it takes less than four hours to train one tree with twenty levels from ten thousand images on a standard PC using one core by Matlab. There are 1000 training example pixels in each image, 33 candidate thresholds per offset and 24 dynamic candidate offsets. It takes 310 hours with 2000 pairs of candidate offset based on Eq. 3.1. 2000 images with ground truth of the head joint are used to test the performance of these two algorithms. The test is implemented on a standard PC using one core by C++. Based on Eq. 3.10, the test time per frame is 2.8 ms. Based on Eq. 3.1, the test time per frame is 5.0 ms, which is 79% more than 2.8 ms. The mean error is measured on the head joint. Their mean errors, i.e., 3.2 cm (Eq. 3.1) and 3.1 cm (Eq. 3.10), are very similar without substantial difference.

Table 3.1: Comparison of algorithms based on [95] and ours.

	Training time per tree (hour)	Test time per frame (ms)	Mean error (cm)
[95]	310	5.0	3.2
Ours	3.9	2.8	3.1

The proposed method can also be used to detect other joints. However, the ground truth of other than the head is more difficult and more subject to deviations. Additionally, other joints rather than head are likely to get occlusion, which is beyond the scope of our current work. Therefore, we only measured the head error.

A more comprehensive comparison is summarized in Table 3.2. Table 3.2 shows the achievements and results of state-of-the-art markerless motion capture techniques in literature. There is no common joints definition among these references. The results in the table are based on their reports. *Depth camera* in Table 3.2 indicates the depth camera model that was used in their experiments. In some references, the used resolutions are resolutions after down sampling the raw images. *With RGB?* indicates whether the system requires the RGB camera. *GPU?* indicates whether the results rely on GPU acceleration to achieve a better frame rate. *Initialization* indicates whether the system requires specific initialization step, such as a specific human posture and manually marking of the joints. If a system requires the user to do some initialized poses, this system is not as convenient as those that do not need the initialization. Some applications cannot allow the pose initialization, such as non-intrusive surveillance. *Model?* indicates whether the system requires an explicit human model in the test phase. *Previous pose?* indicates whether the system requires the pose in the previous frame in order to calculate the pose in current frame, which may face the feature-drift problem. *Accuracy (cm)* is the joints accuracy.

From Table 3.2, we can note that the speed of the proposed algorithm outperforms other algorithms, and the accuracy is better than or comparable with the other algorithms, while it does not need any initialization or previous pose, resulting in a more convenient system than those algorithms that need the initialization and/or previous pose.

Table 3.2: Comparative summary.

	<i>Depth camera</i>	<i>Resolution (pixels)</i>	<i>With RGB?</i>	<i>GPU?</i>	<i>Initialization?</i>	<i>Model?</i>	<i>Previous pose?</i>	<i>Accuracy (cm)</i>	<i>Speed (fps)</i>
[56]	XB3	1280×960	NO	YES	YES	YES	YES	N.A.	2.5
[31]	SR4000	128×128	NO	YES	YES	YES	YES	10-20	4-6
[35]	PMD	64×48	YES	NO	YES	YES	YES	N.A.	5
[132]	SR3000	176×144	NO	NO	YES	YES	YES	3-10	10
[50]	SR4000	176×144	YES	NO	YES	YES	YES	N.A.	10-14
[84]	SR4000	176×144	NO	NO	NO	NO	NO	N.A.	17
[119]	SR4000, Kinect	176×144, 640×480	NO	YES	YES	YES	YES	<10	44
[104]	SR4000, Kinect	160×120	NO	NO	NO	YES	YES	3-6	68
[5]	SR4000, Kinect	176×144, 640×480	NO	YES	NO	YES	YES	<10	60- 100
Ours	Kinect	640×480	NO	NO	NO	NO	NO	>3	357

3.4 Conclusion

The RDT algorithm is very fast and robust in the test phase for human motion capture by utilizing single depth images. However, the computational complexity of the training phase is extremely high. To address this major disadvantage of the RDT algorithm, we proposed a fast training approach. The proposed RDT training algorithm based on the one-offset feature reduces the number of candidate offsets significantly under the same training conditions. The computational complexity of training is linearly reduced with the decreasing of the number of feature candidates. It makes the implementation of RDT algorithm for motion capture much faster. Researchers can thus pay more attention to develop their customized applications.

Chapter 4

Fall Detection Framework Based on Markerless Motion Capture

In this chapter, we develop the markerless motion capture technique to the fall detection for elderly people. We propose a robust fall detection approach by analysing the tracked key joints of the human body by using a monocular depth camera. Compared to the rival methods that rely on RGB images, the proposed scheme is insensitive to the lighting illumination and can be used even in a dark room. In our scheme, a pose-invariant Randomized Decision Tree (RDT)-based algorithm is proposed for the key joint detection, which has low computational complexity during the training and test periods. Then, the Support Vector Machine (SVM) classifier is used to determine whether a fall motion occurs by using the 3D trajectory of the head joint as the input feature vector. The experimental results demonstrate that the proposed fall detection approach is more accurate and robust compared with the state-of-the-art approaches.

4.1 Related Work

In [78, 129, 74], the authors reviewed the approaches used in existing fall detection systems. Nowadays, fall detection approaches could be classified as two main classes: non-vision-based methods and vision-based methods. Most methods of fall detection employ

inertial sensors, such as accelerometers, since they are low costs. However, the methods based on inertial sensors are intrusive. As the vision technologies developed fast during the past few years, the vision-based methods, which are non-intrusive, have become a focal point in the research of the fall detection. They can capture the object's motion and identify the background environment and analyse their relationship, such as the human lying on the floor. The feature of the vision-based systems can be posture [112, 127, 14, 128], shape in-activity/change [29, 92], spatio-temporal [30], 3D head position [89, 93], and 3-D silhouette vertical distribution [4]. To improve the accuracy, some researchers integrated the non-vision-based method and the vision-based method [24]. Most of the fall detection methods based on vision try to execute in real time by using normal computers. The fall motion is very fast, taking a few hundred milliseconds while the image processing requires high computational complexity, which normally takes long time. Thus, most vision-based existing methods cannot capture the specific motion during the fall phase. Even though recent researches on fall detection based on computer vision showed some practical frameworks, the robustness as well as accuracy of vision-based methods still have many open challenges for fall detection systems to develop.

In this section, we will review some existing fall detection methods.

4.1.1 Classification of Fall Detection Techniques

From the first section of this chapter, we know what is a “fall” and a fall detection system is important for elderly people. Here, we will classify the existing fall detection techniques.

From the definitions of fall, there are some common features among falls. But a robust fall detection approach should consider different scenarios and kinds of fall, such as falls from walking or standing and so on. Some fall detection systems can be found in [78, 129, 74].

To have a clear view of all the existing fall accident detection techniques, we classify them into a category tree (Figure 4.1). Based on this category tree, we will analyse the approaches from the root of tree in the following sections.

Figure 4.1: Overview of different fall detection methods.

As shown in Figure 4.1, fall detection systems could be roughly classified as two groups: non-vision based methods and vision based methods. Non-vision based methods include two types: wearable device based methods and ambient device based methods.

Wearable device based fall detection methods could be further classified as accelerometer, posture based and fusion of accelerometer and other wearable devices. Ambient devices based fall detection methods are divided into vibration based methods and audio based methods. The vision based fall detection methods could be classified as posture, shape inactivity/change, spatiotemporal, 3D head position analysis and skeleton analysis. Many existing techniques have a similar framework. We will show some examples in the following subsections.

4.1.2 Wearable Device Based Techniques

The wearable device based techniques are based on wearing some embedded sensors to measure the body motion or body posture. These sensors are always named as smart sensors since they are small and highly sensitive. In the following parts of this section, several different methods based on wearable devices are presented. All of them have a common significant disadvantage that it is an intrusive technique.

4.1.2.1 Accelerometer

Nowadays, the development of science and technology has created devices that can measure human activities by using accelerometers. There are some existing fall detection systems based on accelerometers. They can be classified into several subcategories, but here we integrate them as one category. The accelerometer, worn on the body, can measure the acceleration for the measurement of daily living activity.

In [18], Chen et al. set up a wireless sensor network by utilizing some accelerometers. The on-board devices capture acceleration sequentially to reduce the burden on the network system. These accelerations are used to detect the human motion. When the human fall, these accelerations will have significant changes.

In [65] Mathie et al. classified an activity as a fall when the vertical acceleration, which is measured by a waist-mounted accelerometer, has a large change from standing

pose to lying pose. The system in [118] uses the accelerometer to measure the velocities to detect fall events.

The accelerometer based method in [96] can also measure the inactivity status and can obtain higher accuracy compared with only detecting the dynamics of falls, but the object should wear an array of sensors. This method also shows that the accelerometers can measure comprehensive information, such as relationships among movement duration, intensity and frequency. This detailed information on behaviour is very useful for fall detection.

Wu et al. in [123] developed a system with the combination of inertial sensors and data recording units to detect pre-impact of fall. The inertial force sensor unit includes an acceleration sensor and a tri-axial angular rate sensor. The key variance of the vertical velocity of the inertial coordinate system is detected before the impact of the fall, and it is based on the threshold detection algorithm.

Fang et al. in [27] developed a fall detection system based on the accelerometer embedded in a mobile phone. Since there is an accelerometer in the mobile phone and the mobile phone will be worn on the human body, the mobile phone can measure the person's motion. This system is suitable for sending out the alert message.

4.1.2.2 Posture Based

With regard to the literature review, posture can also be used to distinguish fall and non-fall events. One of the most important posture features is the body orientation. And a gyroscope can be used to measure the body orientation.

In [99], Sorvala et al. proposed a fall detection approach based on posture information. The triaxial gyroscope and the triaxial accelerometer are mounted on the waist and ankle. The human orientation angle would have a large change during a fall. The triaxial accelerometer is used to measure the impact. The post-fall postures are the key feature of this method.

4.1.2.3 Fusion of Accelerometer and Other Wearable Devices

Physical activity and body position changes may lead to different physiological responses, i.e., the information signals of blood pressure and heart rate. This makes it a key factor in the assessment of the dynamic monitoring environmental movement and posture.

In [61], Luo et al. represented acceleration vectors in a 3D space. Some sensors work together to provide a 3D motion model of the human body by using a Gaussian filter. Based on this 3D motion model, the system can analyse the body posture and motion status.

In [47], Kang et al. proposed a wrist-worn fall detector, which integrated health monitoring devices with emergency telemedicine and remote reporting capabilities. The wrist region limits the fidelity of the measured biological signal. This disadvantage can be overcome by a further development of the attitude sensor. Therefore, a two-axis acceleration sensor and a posture sensor have to be used.

In [48], Kantoich et al. proposed a fall detection approach integrated accelerometer device with electrocardiogram (ECG sensor). The ECG device can measure the human heart rate. And the accelerometer device measure the human motion. When a hurt fall happens, the heart rate should change dramatically.

In [51], the authors proposed a fall detection approach integrated accelerometer device with vision system (depth camera). A large signal value will trigger the camera to further analyze the human status since a considerable number of false alarms were reported by using only the accelerometer. If the lying pose is detected, the alarm is triggered. The lying pose is classified by a k-NN classifier. This system requires the subject to wear the accelerometer device, which is an intrusive device and may not be worn always. Though the head is used to improve the human detection, the head detection uses a simple ellipsoid model to fit the 3D point cloud. The head matching does not consider

the anthropometric well, such as the shoulder near the head, which may be trapped in the similar object with the ellipsoid shape.

4.1.2.4 Discussion on Wearable Devices

The wearable device based fall detection method has its own disadvantages as well as advantages. Low cost of wearable device is the main advantage for these systems. Some wearable device methods can operate at home, and also operate outdoor. They can monitor the human in the whole day. However, the most significant disadvantages of these methods are that they intrude in daily life, and the devices may not be worn at all the time. These two disadvantages mean that the methods are not user friendly for elderly people. The power requirement of the wearable device is also a potential problem since the battery has a limited lifetime.

4.1.3 Ambient Device Based Techniques

Ambient devices do not need to be worn on the human body. They are the non-intrusive devices. Vibration sensors and audio sensors are the two popular sensors of ambient devices for fall detection.

4.1.3.1 Vibration Based

The vibrational data can be used to detect some events and their changes. It can be employed in localization, tracking, monitoring and so on.

Alwan et al. in [2] proposed a completely unobtrusive and passive system for fall detection based on floor vibrational sensors. The floor vibration patterns is used to estimate human falls. The principle of the approach is to use the vibration patterns of activities. A special piezoelectric sensor coupled to the floor surface is used. The ground vibration signature of fall is different from normal human activities, such as standing. The vibration patterns are analysed by an alongside circuit.

In [117], Tzeng et al. proposed a fall detection method based on floor pressure and infra-red images. Figure 4.2 is the flow chart of the method in [117]. There is a pressure sensor in the floor. If the floor pressure sensor detects a large floor pressure, which is larger than the previous set threshold, the infra-red camera is activated. The system will further analyse whether it is a fall event or not. This infra-red camera can operate in the room with low light condition and in a dark room. But this infra-red camera is a two dimensional camera, which is similar with the traditional color camera.

Figure 4.2: The flow chart of fall detection in [117].

4.1.3.2 Audio Based

Usually, there is some sound generated when a fall happens. The sound includes the crash noise and the screech from the person. These sounds can also be utilized to detect fall accidents.

Zhuang et al. [133] developed an approach with a far-field microphone to measure the audio signal. The Euclidean distance is used to measure the pairwise difference of audio signals. A support vector machine classifier is employed to detect the fall.

4.1.3.3 Discussion on Ambient Devices

Pressure sensors are used in many approaches based on ambient devices. The principle of the pressure sensor for detection is based on the high pressure of the objects owing to the weight of the object. This is very effective and less intrusive. The audio sensor is another type of ambient devices. The ambient devices measure all events around and in the object by pressure or sounds. Thus, the accuracy of detection is not high. Furthermore, some pressure sensors are not easy to set up. And the acoustics sensor based method has an innate disadvantage that some falls are soundless.

4.1.4 Vision Based Techniques

Vision based techniques are also named camera based techniques. In recent years, cameras are widely used in different situations, e.g., a smart home system, since they have many advantages compared with the systems based on other sensors. Cameras could measure various things without intrusion, which is a very important property in practical use. For the application of real-time fall detection, most of the imaging techniques are too complex. Many researchers try to use some simple image processing algorithm to detect falls. There is a large room for further improvement of the vision based fall detection systems.

4.1.4.1 Posture

The posture information can be used to improve the accuracy of fall detection. The posture method based on vision is similar with the wearable sensor based posture method, but the posture method based on vision can monitor the whole human body posture, which includes more information.

Thome et al. [111] proposed a fall detection system based on orientation information of the subject. The system tracked the angle of the person orientation in real time. They used the HHMM (hierarchical hidden Markov model) as the classifier.

In the research [32], the Kinect depth sensor was mounted on the ceiling to monitor the indoor environments. The system uses some manually predefined anthropometric relationships and features to detect the human subject from all the blobs in the scene. When the person's central point is lower than a threshold, a fall is detected. This system does not measure the motion information since the detection is only based on the status of the person's central point. It cannot distinguish similar voluntary events. The person's central point is not a point fixed to a skeleton joint. The trajectory of this central point cannot reflect the joint's motion well.

4.1.4.2 Shape Inactivity/Change

The body shape is also a common feature applied in fall detection system. The extraction of shapes from images are usually compared with some known shapes to make a further decision. Some shape-based detection methods are presented in the following.

Foroughi [29] proposed a method to detect various shape-based events (including fall event) for elderly people at home. An ellipse shape and projection histograms were employed to detect the human body shape. The temporal changes of the body head position were also used in this approach. A Multilayer Perceptrons (MLPs) neural network was employed as classifier.

Rougier et al. [92] proposed a method to detect fall accidents based on human body shapes. A matched value was employed to stand for the change of shape deformation. If this value became large, it was recognized as the abnormal activity.

In [101], the authors proposed a two-stage fall detection system. The first stage roughly segments the on ground event based on the person's (foreground pixels) vertical state over time. In the second stage, an ensemble of decision trees is employed to compute the confidence that a fall preceded on a ground event. The maximum height of the separated object (human) and the height of the object's centroid are used to normalize the person's vertical state value. The second stage employs five features, i.e.,

- (i) minimum vertical velocity (MVV),
- (ii) maximum vertical acceleration (MVA),
- (iii) mean V_{avg} of smoothed vertical state values,
- (iv) occlusion adjusted change in the number of elements of the discretized (floor) plane to which only points from the object with a height below 38 cm project,
- (v) minimum frame-to-frame vertical velocity (MFFVV).

Compared to the multiple RGB cameras system [4], this system can be more robust to the complex background, ambient lighting and 3D movement of objects. Its setup is easier too. The features are based on some low level image features, i.e., foreground pixels without body parts classification. The authors added some smoothing and normalizing operations to reduce the variation. The experimental results show that these smoothing and normalizing operations can improve accuracy and robustness.

4.1.4.3 Spatiotemporal

The spatiotemporal features of shape modelling offer crucial information of human activities. They can be used to detect different activity events. Some fall detection systems can obtain reasonable accuracy results based on a spatiotemporal feature.

Foroughi et al. [30] proposed an approach of fall detection based on combination of eigenspace technique and ITMI (Integrated Time Motion Image). The motion and the motion time are included in ITMI, which is a spatio-temporal database.

4.1.4.4 3D Head Position Analysis

The head is the most visible part of the body. The head motion information is a strong feature of falls. The head tracking methods are used for analyse of head position to determine the obvious motion in the video sequence. Different states of the model are used to track the motion information of the head.

Rougier et al. [89] developed a fall detection method by using 3D head trajectories. When a fall happens, the head of the subject should move large in a short time. The 3D head trajectory was extracted from a single calibrated camera. A 3D head model was used to match the head in 2D image planes. The 3D head trajectory is employed for detection. But this system cannot extract the 3D information about the background environment, and the detection is based on some predefined simple thresholds.

The authors [77] use a human head model, a circle, to match the input image to detect the head position. The HOG (Histogram of Oriented Gradient) algorithm is also used to avoid variations due to different types of hair style. The speed of the head and the body centroid and their distances to the ground are used to detect falls. The fall detection algorithm works as follows:

- (i) Detect the head position and compute its vertical speed.

- (ii) If the head speed is reliable and the head satisfies the falling conditions, conclude that the detected person is in the falling state. The head satisfies the falling conditions if its vertical speed is higher than V_{min} and the distance from the head to the ground is less than h_{min} ($h_{min} = 0.5m$).
- (iii) Otherwise if the centroid speed is reliable and the centroid satisfies the condition of falling, conclude that the detected person is in the falling state.

However, when the head contour does not appear as the model contour, the head detection algorithm in [77] cannot detect the head position. The HOG algorithm can be used only when both shoulders can be clearly viewed by the camera. The speed threshold of head or body centroid is calculated by assuming that the subject is 1.7 m high. The threshold is sensitive to the subject's height. When the subject falls, the subject is not always falling from the standing posture. The approach lacks of post-fall analysis to reduce the false alarm.

4.1.4.5 Skeleton Analysis

In [49], the authors introduced two methods to detect falls based on the skeleton extraction of Kinect SDK:

- (i) A single frame is used to detect falls by measuring the joints height. When every tracked joint has a distance smaller than predefined threshold, a fall is estimated.
- (ii) Time series data is combined to detect falls. When the average velocity of all 20 joints is less than a predefined threshold, a fall is estimated.

The first method cannot distinguish between true falls and similar activities such as slowly lying down. The second method uses the average velocity of all 20 joints including the foot joints and hand joints. However, some joints' motion is not related to the fall

events. The vertical velocity of feet is very small during falling. The vertical velocity of hands may be very fast during normal activities. The occlusion of some joints out of 20 joints happens easily. The average is changed by the occlusion of some joints while the predefined threshold is fixed. Furthermore, the joints accuracy from the Kinect SDK is low when the subject is not in the upright posture.

The research in [45] fuses the data from the skeleton extraction of Kinect SDK and the bounding ellipse features. A depth camera is used to extract the skeleton, and is mounted on a side view. An overhead RGB camera is used to extract the foreground for the bounding ellipse. The fusion uses the overlap ratio between the foreground segmentation and skeleton re-projection to weight the quality of skeleton extraction. The experiments in this research show that the fusion of these two types of data can outperform the skeleton data or bounding ellipse features alone. However, the skeleton data is from the Kinect SDK, which is designed for entertainment and requires the subject to be in a upright pose. Although the skeleton data can improve the method based on the bounding ellipse features, the bounding ellipse features are sensitive to the foreground extraction. When the ambient light is complex, such as shadow, weak/dark light, the foreground extraction might not work well. The hand interacting with some objects will also impact the foreground extraction as pointed out in [77]. The privacy of the subject suffers from the use of the RGB video, which may be rejected by the elderly people.

4.1.4.6 Discussion on Vision Based Techniques

The fall detection systems based on vision tend to outperform other approaches. They cannot only capture the object's motion, but also analyse the background environment and their relationship, such as the human lying on the floor. On the contrary, the accelerometer based method can only measure the motion of the body part. When a person is lying on the bed or sofa, the accelerometer cannot capture useful information to distinguish between these two different cases.

Recent researches in fall detection based on computer vision show some alternative systems. Most of fall detection systems based on vision try to execute in real time by using low cost hardware. But, the robustness as well as the accuracy of these techniques still have many challenges requiring for novel solutions. From previous description, many methods used some weak fall features, such as the shape bounding box, to detect the falls. The human activities include rich semantic description in vision based analysis. Different levels (low and high) features of activities can be extracted in computer vision processing.

Models are popular in the existing fall detection system based on vision, such as a 3D model of head. Compared with other methods, it is much easier to obtain postures by model dependent method, and it is also more robust. However, most 3D methods are not automatic and generally the manual calibration is required. The 3D information obtained from 2D camera usually takes a high cost in computing. As the output of RGB cameras are 2D images, most of the existing models for fall detection systems are 2D models. These models can recognize various events and physical activities. However, the recognition depends on the object extraction from the 2D information of RGB images. Due to various colours and the complex lighting environment, the object extraction is still a serious problem in RGB image.

For the methods based on the 3D head motion, the abnormal vertical velocity of body is a useful signal for fall detection. The 3D head position is calculated from the 2D image, it limits the capability of tracking and its speed and accuracy are low. Therefore, the robustness of these methods is still a challenging issue.

From the previous analyse of the existing methods, we can note that there is still large room for further improvement of the vision based fall detection methods.

4.1.5 Discussion on the Existing Techniques

The existing fall detection methods are required to improve the robustness and the accuracy. However, they show some worth basic principles to develop new methods for fall detection.

As shown above, the wearable devices can capture the human motion well. However, they intrude the daily life and may not be worn all the times. This unfriendly drawback is a serious trouble for practical application. Since these methods cannot provide some useful background environmental information and the long-time daily activity motions are too complex, it is easy to have a false positive when a fall-like activity happens. The ambient based method overcomes the intrusive disadvantage of wearable device based method. But the ambient devices sense everything (not only the human) around the sensors, the accuracy is usually lower than that of wearable device based method.

Vision based methods can overcome the disadvantages of both ambient based method and wearable device based method. They are non-intrusive methods, and can synchronously identify the useful environment around the object. However, most of image processing are too complex for real time fall detection. The existing vision based methods are usually based on some simple and weak features to detect falls, which tend to lack of flexibility. Most of them were tested in specific laboratory environments. The accuracy as well as robustness is still an open issue for further improvement.

4.2 Overview of the Proposed Approach

The depth cameras have been employed for fall detection systems [88, 85, 10, 11]. One of the popular commercial depth cameras is Kinect [69]. Thanks to the infra-red LED, the depth camera is insensitive to lighting illumination and can still work well in weak lighting environments, even in the middle night without light. When the light condition significantly changes such as switching on or off the lights, it can also work well without

degeneration. As we know, some falls are caused by the weak light condition, and the traditional RGB cameras cannot work well in this condition. So the depth camera based fall detection systems can cover more complex environments, resulting in more practical applications. Instead of the traditional color or intensity information, in the depth image, each pixel value represents the depth information. The depth value is the distance between the object and the camera. The depth information can be used to calibrate each pixel to a real world 3D location point. The depth image can represent the 3D surface of the object. Compared with the traditional intensity or color camera, the depth camera provides several useful advantages in the object recognition. What is more, depth cameras are useful for removing ambiguity in size scale. The object size in the color or intensity image is changed according to the distance between the object and the camera. That introduces ambiguity in size scale since the distance is unknown. In color or intensity images, the shadow greatly reduces the quality of background subtraction. Besides that, depth cameras can resolve silhouette ambiguity of the human body. And they can simplify the tasks of background subtraction and floor detection. That can improve the robustness of the object recognition and can offer some useful information about the relationship between the human and the environment, such as the human hitting the floor. In addition, the realistic depth images of human can be much easier to synthesize. Therefore, a large and reliable training dataset can be built up for markerless motion capture [95].

In this chapter, a robust fall detection system based on human body part tracking by using a monocular depth camera is proposed. To capture the fall motion, an improved Randomized Decision Tree (RDT)-based algorithm is employed to detect the 3D human body joints. By tracking the 3D joint trajectory, a Support Vector Machine (SVM) classifier is proposed for the fall detection. The 3D joint trajectory pattern is the input feature vector of the SVM classifier. Figure 4.3 shows the overview of the proposed fall detection system.

Figure 4.3: Overview of the proposed fall detection system.

The structure of the rest of this chapter is listed in the following sentences. Section 4.3 presents the joint detection approach. Section 4.4 introduces the fall detection methods based on the SVM classifier. Experimental results are presented in Section 4.5. Conclusions are presented in Section 4.6.

4.3 Joint Detection

To capture the 3D human fall motion, a markerless joint detection is applied. The joint detection is based on the proposed RDT algorithm, which is trained by large depth image dataset.

4.3.1 The Feature for Joint Detection

We employ Eq. 3.10 as the feature for joint detection. This feature is with a higher computational efficiency than that of Eq. 3.1 in [95]. That can save running time and leave more time for real-time fall detection application.

After per-pixel classification, the position is presented by the mean of the same class pixels. Since the head and hip, which are used in the proposed fall detection, are the most visible body parts, we can use body part classification to extract them. The result of mean-shift clustering method is more accurate than the result of the global centre of the same body part since there are some outlying pixels as shown in Figure 4.4. Figure 4.4 demonstrates a classification result of the human body. Each color in Figure 4.4 stands for the respective part of the human body with the highest probability.

Figure 4.4: The classification result of each depth pixel, each color stands for the respective part of the human body with the highest probability. Blue: head, red: hip, green: other part.

4.3.2 Pose Correction

In order to track the joint trajectory efficiently, the frame rate of the camera output should be high and the joint extraction should be fast. The frame rate of Kinect is

Figure 4.5: Joint extraction results.

30 frames per second, and the joint detection based on RDT takes a few milliseconds. Therefore, the human joint trajectories can be tracked well.

Based on the RDT algorithm, an open license software has been released, i.e., Kinect for windows SDK [69]. Figure 4.5.a shows that the human joint extraction from the SDK is correct when the person is standing even moving a chair at the same time. However, the SDK joint extraction cannot work well under some scenarios, such as the person lying on a sofa as shown in Figure 4.5.b. The degraded accuracy problem also appears in the moment of falling down since the orientation of the human body changes significantly while falling.

We propose to rotate the depth image in Figure 4.5.b by 90 degree clockwise, and the key joint positions can be extracted accurately as shown in Figure 4.6. We always rotate the torso of the human body to be vertical in the image before the joint extraction algorithm in the proposed vision based fall detection system, as inspired from this observation. Thanks to the high frames throughput of the depth camera, it is fast enough

Figure 4.6: After pose rotation, the joints of head and hip centre can be detected correctly.

to track the human motion during daily activities and falls. The straight line through the hip centre and the head is employed as the orientation of human torso. Even during the fall event, the orientation of human torso changes very little per frame. Thus, the processing of rotating the torso based on the human posture in the last frame can correct the orientation of human torso well enough. Therefore, the joint extraction is pose-invariant.

The head and hip centre positions are detected when the subject walks into the view field of the camera. For the following image frame, these positions will be employed to correct orientation of human torso, and the posture is renewed each image frame. Figure 4.7 shows the image sequences before and after the correction processing. Figure 4.7.a shows the original image sequence of falling down. Figure 4.7.b shows the corrected image sequence of Figure 4.7.a, and the correction processing can always correct the orientation of human torso to be vertical in the depth image frame via the last image.

If the orientation of human torso tracking is lost, the re-initialization is required. The input image after the subject segmentation was rotated by several angles, such as 0° , 120° and 240° , to generate several images, as shown in Figure 4.8. For each pre-rotated depth image, the hip centre and the head are detected to get the orientation

4.7.a:

4.7.b:

Figure 4.7: A fall sequence by a person simulating. (a) The sequence of original poses without correction. (b) The sequence of with correction by proposed approach.

of human torso, which is used to correct the human posture. The joints detection and the posture correction are done some times. Three final human postures are detected after some iterations of the joints detection and the posture correction. The density estimator (Eq. 3.3) of the body part detection is employed to decide which candidate will be selected. A simple way is to select the best candidate by the head joint's density estimator.

To speed up the joint extraction, the rotation of the depth image can be taken place by rotating the offset parameter $(\Delta x_n, \Delta y_n)$ of each split node n of RDT. This rotation of parameters can be done off-line after training. And the rotation angles are fixed, such as $10^\circ, 20^\circ, \dots, 350^\circ$. Thus, the pose correction does not increase the computational

Figure 4.8: An example of re-initialization.

complexity in the test phase.

4.4 Fall Detection Based on SVM

This section will introduce the fall detection based on SVM, which employs the head joint distance trajectory as input feature vector.

4.4.1 Fall Motion Analysis Based on SVM

“Inadvertently coming to rest on the ground, floor or other lower level, excluding intentional change in position to rest in furniture, wall or other objects.” is the definition of fall by World Health Organizations [80]. After the joint extraction, “coming to” “the ground” can be described in a technical word, i.e., some key joints coming to the ground.

This feature can be used for fall detection.

Referring to the above feature, the fall is an activity related to the floor. The floor information should be measured. In [88], they assumed that the floor occupies a sufficiently large part in the image. However, this assumption is not always true such as a small bedroom with a bed. In the proposed system, three selected points of the floor in the depth image define the 3D floor plane. The equation of the 3D floor plane is

$$Ax_w + By_w + Cz_w + D = 0, \quad (\text{Eq. 4.1})$$

$$A^2 + B^2 + C^2 = 1, \quad (\text{Eq. 4.2})$$

where A , B , C and D represent four coefficients, x_w , y_w and z_w represent three real world coordinate variables. Eq. 4.2 is used to normalize A , B and C . The selected three points are employed to solve these two equations.

The joint distance trajectory is defined as the trajectory of the distance between the floor and the joint. This signal includes the fall motion information, such as the acceleration, the joint velocity, the distance between the floor and the human joint and other hidden information. Figure 4.9 shows a fall pattern and a non-fall pattern of the head distance trajectory. The fall motion can be classified by the joint distance trajectory pattern.

A d dimensional feature vector is formed by the distance trajectory in d consecutive frames. d should be large enough to cover all the phases of falling including rapid movement period during the fall, the period before the fall and the period after the fall. The fall detection can be seen as two-class classification problem. SVM is very suitable for two-class classification problem. It can automatically learn the maximum-margin hyperplane for classification. The aforementioned feature vector can be used as the input feature vector of SVM classifier.

4.9.a:

4.9.b:

Figure 4.9: Two patterns of head distance trajectory from a same video sequence. (a) Fall pattern. (b) Non-fall pattern.

4.4.2 Training Dataset

Since the 3D head joint trajectory has been tracked, the head joint motion can be analysed by the physics mechanics principle. During the falling phase, the joint motion can be seen as a free fall body. The free fall body is described as a simple formula

$$h(t) = h_0 + \frac{1}{2}a(t - t_0)^2, \quad (\text{Eq. 4.3})$$

where $h(t)$ is the height at the time t , h_0 is the height at the beginning of fall, a is the acceleration, t is the current time and t_0 is the beginning time. The free fall body can be used to simulate the joint fall motion in order to generate fall patterns by computer. h_0 in Eq. 4.3 refers to the initial height of fall. When the subject is falling, the vertical motion is driven by gravity, rather than the subject's size, shape, height or clothes. Therefore, the vertical velocity of the head hitting the floor is related to the initial height, but is insensitive to the subject's size, shape, height and clothes.

Figure 4.10.a shows a free fall body trajectory fitting the head distance trajectory of a falling person. It can be noted that the free fall body curve fits well. The difference can

4.10.a:

4.10.b:

Figure 4.10: Free fall body trajectory fits the head distance trajectory. (a) Without noise. (b) Free fall body with noises.

Figure 4.11: Non-fall patterns simulated by computer.

be considered as Gaussian white noises. In order to improve the robustness, Gaussian white noises are added into the free fall body curve, as shown in Figure 4.10.b. Some non-fall patterns can also be simulated by computer, as shown in Figure 4.11. Based on the free fall body simulation, a large fall and non-fall patterns dataset can be built up.

4.4.3 Fall Confirmation

The recovery motion analysis after fall motion is required to confirm the fall detection.

There are two recovery metrics: (a) the heights of hip and head are higher than a recovery threshold value $T_{recover1}$ for a certain time; (b) the head height is higher than a high recovery threshold value $T_{recover2}$ for a certain time. If one of these two metrics can be satisfied, it means that the person is recovered. In our experiment, referring to hip breadth size and other sizes of the anthropometry [72], we set $T_{recover1} = 0.5m$ and $T_{recover2} = 0.8m$.

After the fall motion, if the person cannot recover within a certain time, fall detection will be confirmed. Without a fall confirmation, the fall alert will not be triggered. This stage can avoid the false alert.

4.5 Experimental Results

To test the proposed method, some normal activities (like crouching down, standing up, sitting down and walking) and falls, which are simulated by the human, have been tested.

4.5.1 Performance Evaluation Metric

The following parameters suggested by [78] are used to analyse the detection results of the proposed algorithm.

- (1) True positives (TP): the number of fall events detected correctly.
- (2) True negatives (TN): the number of non-fall events detected correctly.
- (3) False positives (FP): the number of non-fall events detected as fall events.
- (4) False negatives (FN): the number of fall events detected as non-fall events.
- (5) Sensitivity (Se): the capacity to detect fall events

$$Se = \frac{TP}{TP + FN}. \quad (\text{Eq. 4.4})$$

(6) Specificity (Sp): the capacity to detect non-fall events

$$Sp = \frac{TN}{TN + FP}. \quad (\text{Eq. 4.5})$$

(7) Accuracy (Ac): the correct classification rate

$$Ac = \frac{TP + TN}{TP + TN + FP + FN}. \quad (\text{Eq. 4.6})$$

(8) Error rate (Er): the incorrect classification rate

$$Er = \frac{FP + FN}{TP + TN + FP + FN}. \quad (\text{Eq. 4.7})$$

A high specificity means that most non-falls are detected as non-fall. A high sensitivity means that most falls are detected. A good method for fall detection should have a high sensitivity and a high specificity. Besides, the accuracy should be high and the error rate should be low.

4.5.2 Dataset

The non-fall and fall activities were simulated as shown in Table 4.1. They are suggested by [78], but with more detailed description. During an impactive fall, the elderly person cannot keep the transient pose kneeling or sitting on the floor. Therefore, the person will lie on the floor after fall as shown in the first row and fifth row of scenarios in Table 4.1. In Table 4.1, “Positive” and “Negative” means fall and non-fall, respectively. In total, there are 20 scenarios. 50% are positive and 50% are negative. Each scenario is simulated several times. Totally, there are 380 samples. There are four subjects, and their heights, ages and weights are: 159-182 cm, 24-31 years old and 48-85 kg. Three are male and one is female. The experiments are in a real bedroom, as shown in Figure 4.12. The camera is mounted 2.3m height on the wall.

Table 4.1: Scenarios for the evaluation of fall detection [78].

Category	Description	Outcome
Backward fall	On the hip, then lying	Positive
	Ending lying	Positive
	Ending in lateral position	Positive
	With recovery	Negative
Forward fall	On the knees, then lying	Positive
	With forward arm protection	Positive
	Ending lying	Positive
	With rotation, ending in the lateral right position	Positive
	With rotation, ending in the lateral to the left position	Positive
Lateral fall to the right	With recovery	Negative
	Ending lying	Positive
Lateral fall to the left	With recovery	Negative
	Ending lying	Positive
Syncope	With recovery	Negative
	Vertical slipping against a wall finishing in sitting position	Negative
Neutral	To sit down on a chair then to stand up	Negative
	To lie down on the bed then to rise up	Negative
	Walk a few meters	Negative
	To bend down, catch something on the floor, then to rise up	Negative
	To cough or sneeze	Negative

The research was approved by the Institutional Review Board of Nanyang Technological University, Singapore.

4.5.3 Performance Validation

To test the performance of SVM classifier method, a large training dataset, which includes about 100k non-fall and fall patterns of head distance trajectory, is generated by computer. After training, the SVM classifier is used to detect falls in the dataset of

Figure 4.12: Room plan for the evaluation.

human simulated scenarios in Table 4.1. The experimental results of fall motion analysis (detection) of SVM classifier is shown in Table 4.2. For the fall motion detection, there are only eight errors, which are FN errors. The head distance trajectory during fall of an FN sample is shown in Figure 4.13. There is an air mattress on the floor to protect the subject, as shown in Figure 4.14. In this fall event, the air mattress bounces the human body rapidly that the head distance trajectory is indicated by a circle in Figure 4.13 when the subject falls backward. The SVM classifier cannot have a correct decision on this head distance trajectory. If there is no air mattress, the dramatic rebound would not happen. In the fall confirmation stage, there is one fall event sample lost tracking the subject. In this sample, when the subject is lying on the floor, the system cannot segment the subject and recognises that there is no subject in the view field. The fall motion in this sample has been detected correctly by the SVM classifier. However, the system fails to confirm this fall event in the fall confirmation stage. It is due to failing in subject segmentation operated by Kinect SDK. In the fall confirmation stage, all the confirmation results are correct, except this lost tracking sample. Without a fall confirmation, the fall alert will not be triggered. Although the fall confirmation stage misses

Table 4.2: The results of fall motion detection based on SVM.

TP	TN	FP	FN	Se(%)	Sp(%)	Ac(%)	Er(%)
182	190	0	8	95.8	100	97.9	2.1

the lost tracking sample, this stage can avoid the false alert effectively. Thus, the system misses nine fall alerts, and there is no false alert.

Figure 4.13: The head distance trajectory during the fall of an FN sample of fall motion detection by SVM classifier while the circle indicates the rebound behaviour.

To demonstrate the fall detection approach based on the SVM classifier, a video demo can be downloaded ¹.

4.5.4 Comparison

To further evaluate the proposed algorithm, we compared it with two state-of-the-art approaches based on a monocular depth camera.

The approach of [88] is based on human silhouette centre height relative to the floor. The human silhouette centre is obtained by the overall of the foreground (with morphological filtering). When the silhouette centre is lower than a threshold, a fall is detected.

¹<http://www.ntu.edu.sg/home/elpchau/FallDetect2013.wmv>

Figure 4.14: In the experiments, there is an air mattress on the floor to protect the subject.

Table 4.3: The results of comparison of different approaches for fall detection.

	TP	TN	FP	FN	Se(%)	Sp(%)	Ac(%)	Er(%)
[88]	188	104	86	2	98.9	54.7	76.8	23.2
[85]	175	151	39	15	92.1	79.5	85.8	14.2
Proposed	181	190	0	9	95.3	100	97.6	2.4

In [88] the floor is detected using a 3D plane equation during the setup of the system. The V-disparity method is used to detect the floor plane equation. the V-disparity image consists of computing the histogram of disparity values for each row of the depth images. Then the Hough transform is employed to extract the straight line corresponding to the ground plane. They assume that the floor represents a sufficiently large part of the scene. However, this assumption is easily violated since there may be a large furniture, such as a bed. The floor detection is done only one time during the setup. Therefore, we use a higher accuracy and more robust method, i.e., selecting the 3D points on the floor of the depth image to detect the 3D plane equation. In [88], a depth background image B is obtained from N_{train} background images. When the difference between the current depth value of a given pixel and the value in the corresponding pixel of B is larger than a predefined threshold, this pixel is detected as a foreground pixel. The 3D silhouette

centre is calculated on all the foreground pixels. Thus, this 3D position is not a human joint position. The motion during falling cannot be a free fall body. This 3D position is rough, and strongly relies on the height estimation result from the foreground extraction. But the foreground extraction cannot be perfect. False alarms are easily triggered by bad foreground extraction. Since the motion feature of the 3D silhouette centre cannot be captured reliably, the authors used the static state of the 3D silhouette centre to detect the fall, i.e., when the silhouette centre is lower than a threshold, a fall is detected.

Its experimental results are shown in row “[88]” of Table 4.3. This algorithm can detect the most of fall events, but with a lot of false positives (FP). It cannot distinguish the fall accident and the non-impact initiative activities well since it does not consider the motion together. When most part of the foreground object is near to the floor, including slowly lying down or sitting down on the floor or bad segmentation, it is detected as fall. The centre location is easily distorted by moving object and incorrect segmentation. To these cases, the two key joints, i.e., head and hip, are still tracked well by the proposed method in our experiment. The proposed joint tracking method has a better robustness.

The approach of [85] makes use of the orientation of the body and the height information of the spine. The orientation of the body is based on the joint positions extracted from Kinect SDK. If the orientation of the body is parallel to the floor and the distance between the floor and the human spine is smaller than a predefined threshold, a fall is detected. The experimental results are shown in row “[85]” of Table 4.3. The main disadvantage of this approach is the unreliable joints extraction. When the subject falls and is lying on the floor, the joint extraction is inaccurate and the orientation of the body obtained from inaccurate joints provides false information. Figure 4.15 shows two examples of the angle between the orientation (extracted from Kinect SDK) and the floor when the subject walked and fell down ending with lying posture. From Figure 4.15, we can see that the orientations extracted from Kinect SDK were wrong when the subject

fell and was lying on the floor. Based on non-robust joint extraction and predefined empirical thresholds, this method's capacity of prediction is limited.

Combined with the fall confirmation, the error rate of the proposed method is 2.4%. From Table 4.3, we can note that the proposed approach is superior to the existing state-of-the-art approaches. Furthermore, compared with [85], which is required to extract at least six joints, less number of joints (only two joints) are required in the proposed algorithm. Integrated with the efficiency feature in Eq. 3.10, the proposed algorithm can have a much lower computational complexity.

Based on the RGB cameras, one of state-of-the-art methods is [4]. In [4], the system uses multiple-camera to reconstruct the 3D-shape of subject, and the fall alarm is triggered when the 3D volume distribution is abnormally near the ground. The speed of this system is less than 1 fps without GPU acceleration, which means a very high computational load for PCs. To extract 3D-shape, the subject should stay inside the overlapping region of camera. Therefore, the covered area is less than that of a single camera and the monitoring region is relatively limited. In [88], the principle of the trigger function is similar to that in [4]. However, a single depth camera is used in [88]. The frame rate in [88] is much higher than that in [4]. If the system uses multiple depth camera, the system can extend the covered region while the multiple RGB cameras reduce the region. Furthermore, the protection of user's privacy is worse when more RGB cameras are used. Thus, the method in [88] advances the state-of-the-art method [4].

The method in [85] can track the 3D body joints avoiding that the actual joint motion is impacted by the average of foreground pixels' change, which allows us to analyze the joints motion better. Meanwhile, the methods in [89, 91, 90] extract the 3D head trajectory from a single camera video. Compared to other 2D camera based methods, this method is real time and uses a low cost single camera. It can provide 3D trajectory of the head. Compared to [85], there are some disadvantages of [89, 91, 90]:

4.15.a:

4.15.b:

Figure 4.15: Two examples of the angle between the orientation (extracted from Kinect SDK) and the floor when the subject walked and fell down ending lying. (a) Two depth image samples from two sequences. (b) Angle waveforms corresponded to the two sequences.

- (i) The head is manually initialized in the first image.
- (ii) The head tracking is more easily affected by the human pose, the human location, similar color objects and the camera view point.
- (iii) The accuracy of the 3D head position is low. The 3D mean error ranges from 17 cm to 51 cm [90].
- (iv) The privacy of the subject suffers from the use of the RGD video, which may be rejected by the elderly people.

Although the methods in [89, 91, 90] advance many other 2D camera based methods, the joint extraction in [85] is more robust and has a higher accuracy. It combines the features of the orientation and position of the subject.

Referring to the RGB camera based research [92], the qualitative comparison with wearable device, 2D vision and 3D vision (multiple RGB cameras) systems is summarized in Table 4.4. The first six items is referred to in [92]. The last four items are new compared to [92].

“Lack of movement” can be detected well by all the systems. ***“Vertical speed”*** cannot be measured well by the 2D vision, which lacks of 3D information. ***“Impact shock”*** requires high accuracy and update rate of the system, which is satisfied by the wearable device and the proposed system; the 2D vision is the worst one; the frame rate of 3D vision based on multiple RGB cameras is not high enough. The wearable device lacks of the environment information and the whole body posture information. Thus, its performance on ***“Lying position”***, ***“Lying on the ground”*** and ***“Body shape change”*** is the worst one. The wearable device intrudes the user since it requires the user to wear the device. Considering the variation of color and lighting, the proposed system is better than the 2D and 3D vision based on RGB cameras. The 3D vision based on RGB cameras is the worst one for the protection of privacy since it uses many RGB cameras.

Compared to the wearable device, the proposed system outperforms the wearable device in four items while in other two items it is not as good as the wearable device. Compared to 2D vision, the proposed system outperforms seven items among ten items. Compared to 3D vision based on RGB cameras, the proposed system is better in four items.

Table 4.4: Qualitative comparison with wearable device, 2D vision and 3D vision (multiple RGB cameras) systems.

	Wearable device	2D vision	3D vision (multiple RGB cameras)	Ours
Lack of movement	++	++	++	++
Vertical speed	++	+	++	++
Impact shock	++	-	+	++
Lying position	+	+	++	++
Lying on the ground	-	+	++	++
Body shape change	-	++	++	++
Intrude user	-	++	++	++
Resist color	++	-	-	++
Resist lighting	++	-	-	+
Protect privacy	++	-	- -	+

4.6 Conclusion

The fall detection system is very important for elderly people. The proposed fall detection approach uses the monocular infra-red based depth camera, so the approach can operate even in the dark condition. The depth camera can measure the human body motion and the relationship between the body and the background environment. The floor plane can be extracted from the depth images. To capture the human motion, an enhanced RDT algorithm is employed to extract the human joints. Existing fall detection method based on joint extraction cannot extract human joints correctly when the subject lies down. The proposed rotation of the person torso orientation increases the joints detection accuracy

for falls detection. After extracting the joints, an SVM classifier is proposed to detect the fall by using the joint trajectory.

Because the proposed motion analysis is based on head tracking and the depth camera can be mounted close to the ceiling, it can avoid most occlusion situations. The capacity of the proposed system has the potential to extend widely in the future since the proposed approach is only used a monocular depth camera. Multiple depth cameras system based on the proposed approach can solve the problem of occlusion. However, the proposed approach cannot detect the fall ending with the posture lying on the furniture, e.g., a high wooden sofa, since the distance between the body and the floor is too high.

Based on the depth image sequence, by extracting and tracking the joints of the human body as well as investigating the joints' behaviour after fall, the proposed approach can detect and confirm the human fall accurately. Experimental results show that the accuracy of the proposed algorithm is improved by 11.8% compared with the most recent state-of-the-art fall detection algorithms.

Chapter 5

Human Computer Interface Based on Markerless Motion Capture

This chapter presents the second motion analysis framework, a human computer interface (namely Facial position and expression Mouse system, FM) for hand-disabled people based on a monocular infrared depth camera. The nose position along with the mouth status (close/open) is detected by the proposed algorithm to navigate and control the cursor as computer user input. The detection algorithm is based on an improved Randomized Decision Tree (RDT) algorithm which is capable of detecting the facial information efficiently and accurately. A more comfortable user experience is achieved by mapping the nose motion to the cursor motion via a non-linear function. The infrared depth camera enables the system to be independent of lighting illumination and colour changes both from the background and on human face, which is a critical advantage over RGB camera based optional interfaces. Extensive experimental results show that the proposed system is superior to existing Assistive Technologies (ATs) in terms of quantitative and qualitative assessments.

5.1 Related Work

Some ATs have been developed to help persons with tetraplegia by using their limited voluntary signals and motions to control computers. Based on different voluntary signals

Figure 5.1: Categories and examples of ATs.

and motions, ATs for people with tetraplegia can be roughly classified into 4 categories [126, 75]: (1) Physiological Signal Based ATs. (2) Voice Command Based ATs. (3) Mechanical Motion Based ATs. (4) Motion Tracking Based ATs. The categories and examples of ATs are shown in Figure 5.1. In the following parts of this subsection, we will review the related work of each group.

5.1.0.1 Physiological Signal Based ATs

Various types of physiological signals are employed to control computers, such as ElectroMyoGram (EMG) [67, 120], ElectroEncephaloGram (EEG) [70, 64] and ElectroOcu-

loGram (EOG) [76], in which the signals are generated from muscles, brain and eyes, respectively. Some of these systems can be used for totally paralysed subjects [64]. One of the major disadvantages of these systems is the difficulty to extract the interaction signal since the signal-noise-rate (SNR) is very low. Other major disadvantages of these systems are low portability and requirement for highly specialized hardware.

In [120], the authors investigated an EMG AT from face and muscles as a human computer interface. The muscles used for the EMG command source are shown in Figure 5.2. The surface EMG signals are collected from the muscles including the frontalis, right platysma, left platysma and left trapezius muscles. The signals from the right and left platysma map to the right and left motion of the computer cursor. The signals from the left trapezius and the frontalis map to the upward and downward motion of the computer cursor. This EMG system provides only four digital commands. Therefore, the motion of the computer cursor is limited on the vertical and horizontal directions. Figure 5.3 shows an example of the cursor trajectory to the target in the experiment of [120].

Figure 5.2: Muscles used for the EMG command source in [120].

Figure 5.3: The cursor trajectory to a target[120].

The EEG system can be used for brain computer interface (BCI) [12, 83, 122], which provide communication and control to the totally paralyzed persons. In [122], the authors proposed a BCI that used scalp-recorded electroencephalographic activity and an adaptive algorithm to provide people with multidimensional point-to-point movement control. Figure 5.4 shows that the system in [122] controls the computer cursor to point the targets.

One example of EOG system is shown in Figure 5.5. Figure 5.6 shows the basic block diagram of the EOG interface system of [17]. It needs to attach some sensors to collect the signals of the eye movements. This system looks obtrusive. The eye movement for command input will affect the normal vision, and the normal vision action will affect the interaction command too.

5.1.0.2 Voice Command Based ATs

Speech recognition and non-verbal vocalization recognition are used to control computers [42, 82]. However, they are unreliable in noisy environments. Moreover, to navigate a

Figure 5.4: The EEG system in [122] controls the computer cursor to point the targets.

Figure 5.5: The EOG system in [17]. The output signal is right eye movement.

Figure 5.6: The overview of the EOG system in [17].

cursor, the method based on voice signals is not as flexible as a motion tracking method [42]. These ATs could have better performance in accompany with other ATs [42].

5.1.0.3 Mechanical Motion Based ATs

With the mechanical motion based ATs, the user controls a computer via the switches or analog devices, such as sip-and-puff [66], mouth stick/pad [54] and lip control system [44]. One of the problems of sip-and-puff and mouth stick/pad interfaces is the hygienic issue. To address the hygienic issue, the authors of [44] proposed a Lip Control System (LCS). However, LCS requires wearing an accessory in front the lip.

The reduced degree of freedom of the sip-and-puff system severely restricts its applicability. Some researchers try to optimize the adaptation of the controller parameters of the sip-and-puff system to control the machines. As in [73], the authors used the sip-and-puff system to control a wheelchair by optimizing the adaptation of the controller

parameters. Its system is shown in Figure 5.7. From Figure 5.7, we can see that a pipe linked to the machine is put into the subject's mouth. Therefore, this system has the hygienic issue. The stick interfaces has the similarly hygienic issue, which requires a stick to keep in the mouth.

Figure 5.7: The sip-and-puff system in [73].

Figure 5.8 shows the mouth pad system. Figure 5.8.a shows the schematic diagram that the mouth pad mounted in the mouth. Figure 5.8.b and 5.8.c show two examples of mouth pad devices. They require a mouth pad placed in the mouth, and the tongue touches the mouth pad to trigger interaction commands.

5.1.0.4 Motion Tracking Based ATs

These ATs track motions of human body parts, such as eye [130, 62] / tongue [126, 42] / head [8, 36] / face [71, 116, 25] trackers. In [130, 62], eye gaze was directly used to select the target. Although the operation is fast, it needs calibration before use and the head of the user should be fixed during use. If the head moves, re-calibration is required. In addition, the methods of eye motions or eye gaze require extra eye motions and interfere with users' normal visual tasks [109]. Figure 5.9 shows the eye tracking system of [130].

5.8.a:

5.8.b:

5.8.c:

Figure 5.8: The mouth pad system used in [103, 102]. (a) The mount pad mounted in the mouth [103]. (b) and (C) two examples of mouth pad devices in [103] and [102].

Its principle of measurement method is the pupil and corneal reflection based on infrared light. This system samples at 30 Hz, and its accuracy is 0.25-1.0 degree visual arc, or about 10-40 pixels under their configuration.

In [62], the authors pointed out that the eye is not capable of moving a fixation point

Figure 5.9: Eye tracking system used in [130].

or cursor along a path smoothly. The eye simply looks at the rough location. This issue causes by the saccade, which is the fast distance-traversing movement. The visual field of the eye's fovea is about 2 degree, which restricts the capability of single-pixel sensing by fixing the eye gaze on a normal monitor. The inherent jitter of eye fixation is further to deteriorate this limitation [55, 121]. In the eye tracking system, the computer cursor usually is not used since the cursor following the eye gaze will make the user uncomfortable or distract the user. In [63] and [37], two visual feedback modes were used to improve the "click" performance, as shown in Figure 5.10. The auditory feedback can be incorporated to improve the "click" performance.

Combined with the gaze motion, the blink action of eyes can be employed to provide the click commands [114]. Figure 5.11 shows the eye system used in [114]. Although the devices of [114] are worn on the subject's face, this system still requires the calibration step, as shown in Figure 5.12.

In [126, 42], the ATs tracked the motion of the tongue. However, the tongue interface in [126, 42] required an accessory embedded into the tongue for long term use, so that it caused the hygienic issue. The major components of the tongue based human computer

5.10.a:

5.10.b:

Figure 5.10: Two examples of feedback methods [63, 37, 62].

5.11.a:

Figure 5.11: Overview of the eye system used in [114].

interface is shown in Figure 5.13. In Figure 5.13, the permanent magnet is embedded into the tongue. We can also note that this system includes the microphone, which is used for speech recognition. Figure 5.14 shows a subject is using the tongue based human computer interface. The tongue based computer interface will be limited by the user oral

5.12.a:

5.12.b:

Figure 5.12: The experiment setting and of the eye system used in [114]. (a) System setting. (b) 12 gaze calibration points.

safety, comfort and risk of interference with mastication/speech [3, 126].

To address the invasive and hygienic issues, i.e., attaching sensors to the tongue, a non-invasive tongue-based interface was proposed in [19]. This interface is attached to the cheek, as shown in Figure 5.15. It includes an array of resistive textile sensors,

Figure 5.13: Components of the tongue based human computer interface used in [42].

Figure 5.14: A subject is using the tongue based human computer interface used in [42].

which measure the pressure drove by the tongue on the inside of the user's cheek. The architecture of the hardware is shown in Figure 5.16. The pressures are mapped to gestures by a k-nearest neighbours algorithm. Figure 5.17 shows some exemplary pressure maps for the five gestures. The detection algorithm has a non-negligible delay since it requires calculating the mean and the variance over 0.75 seconds sliding window.

For persons with tetraplegia who have the capability to move their heads with relative flexibility, head tracking systems are the good optional interfaces. The head of human has multiple degrees of freedom [46]. Nevertheless, some existing ATs require the users

Figure 5.15: Two examples of the tongue based interface attaching to the cheek [19].

Figure 5.16: Overview of the tongue based interface used in [19].

Figure 5.17: Five examples of gesture pressure maps: “click”, “swipe left”, “swipe right”, “swipe up” and “swipe down” [19].

to put on or wear accessories, such as head band, glasses, cap and markers. As shown in Figure 5.18, a head motion based human computer interface requires the user wearing a head band with inertial sensors. And the inertial sensors will help to measure the rotation of the head, and map the rotation movement to the movement of the computer cursor.

Figure 5.18: A head band with inertial sensors for head motion based interface used in [86].

For the non-contact requirement, camera-based methods are optional interfaces to enable users to control computers effectively without wearing any accessory. In [8], based on an RGB video, Camera Mouse tracked a manually selected and automatically updating online template, in which a guardian was required to start the tracking for normal operation. While using Camera Mouse, the subject is required to look at the monitor when the tracking point is on the face. If the subject rotates his head at a large angle to look at other places, it will lead to the loss of tracking and manual re-selection of a tracking feature is needed. The initial template is a square around the clicked feature in the beginning of use. Figure 5.19 shows some examples of selected features. In the next frame of selection, a tracking algorithm searches for this feature and detects the position of this feature. The search area is limited in the neighbourhood of the last position of the feature, as shown in Figure 5.20. A new template around the current position of the feature becomes the updated template, which is employed to search for the position of the feature in the next image frame. The performance of tracking varies with templates, search area and the velocity of the feature motion. If the template is very complex and different from other patches around the feature, the robustness of tracking is better. As shown in Figure 5.19, the "lower lip and chin" template is more complex than the "nose". Thus, the "lower lip and chin" template may be better than the "nose".

The tracking algorithm may detect a similar image patch as a new template, which is not the initial template. This error will pass to the next detection and the error will be accumulated, resulting in drift problem. The user motion and the illumination will change the appearance of the feature, and they also affect the performance of tracking.

Figure 5.19: Some templates for tracking [8].

Figure 5.20: Schematic diagram of the search window [8].

To make the extraction of features compact, in [71], the skin colour was used to detect the users' face and nose, which is vulnerable to complex illumination and similar colour objects. The flow chart of this system is shown in Figure 5.21. For each frame, this system requires to detect the face coloured region by color clustering. False positives are detected as background regions. The large coloured region is detected as user's face. The performance of face region detection will be affected by the objects with similar colour.

Figure 5.22 shows the face regions by colour clustering. Then the two nostrils are selected as tracking facial feature, and are located by a threshold process: the pixels inside the detected face region are divided by a ramped threshold to find two regions matched with the nostril heuristics. The performance of nose detection is affected by the shadows and other dark areas. As shown in Figure 5.22, there are many dark points inside the face region. The average position of the nostrils is used to navigate the cursor. The approach is able to run at 30 fps with 160×120 pixels, or 18 fps with a resolution of 320×240 pixels.

Figure 5.21: Flow chart of the system used in [71].

In [116], the user's face was tracked by a cost function of errors between 2D features in RGB video and a 3D face model. However, the tracking method is also vulnerable to colour and the illumination environment, and requires initialization step. Figure 5.23 shows the 3D facial articulation model. The retrieved (rigid) motion parameters of the 3D model are used to control the cursor movement. The detection of mouth (non-rigid) motion is used to trigger commands. In the first frame of the video, the generic 3D

Figure 5.22: Face regions by colour clustering in [71].

articulation model should be initialized to adapt and deform to fit the special user's 2D facial features. Figure 5.24 shows a result of initialization. The red face mask outline illustrates the facial feature localization result, and the 3D face articulation model is illustrated by the yellow mesh. The tracking algorithm is based on an extended Kalman filter. The displacement of image is detected by optical flow. The displacement of model parameter is computed by LMSE (Least Mean Square Error) fitting. The mouth motion parameters may be destroyed by the global LMSE fitting of the entire facial model. An increase of the LMSE model fitting error and the confidence measure drop of the optical flow will trigger re-initialization of the tracking. When this system only tracks rigid motion, it can run at 22 fps. If it tracks both non-rigid and rigid motions, it can run at 17 fps.

In [25], the authors alleviated the “feature drift” and illumination-dependent problems of [8]. However, that system is not very robust yet, especially when the motion of the user is fast.

5.2 Overview of FM

The proposed human computer interface (namely Facial position and expression Mouse system, FM) employs a monocular commercial depth camera, such as SoftKinetic [97]

Figure 5.23: The 3D facial articulation model used in [116].

Figure 5.24: A result of initialization of the 3D facial articulation model [116].

and Kinect [69], and is based on the facial position and expression. Figure 5.25 shows the flow chart of the proposed approach. Several challenging issues are well addressed, i.e.,

- (i) FM is independent of colour and illumination influences since it is based on an infrared depth camera. The main interference source is the infrared light such as direct sunlight [87]. Moreover, the depth image simplifies the task of facial feature extraction, which makes it more robust than its rivals, i.e., RGB camera based methods.
- (ii) A fast and robust randomized decision tree (RDT) is proposed to automatically detect the position and the expression from a single image, which can avoid the “feature drift” problem that exists in most tracking algorithms.
- (iii) The proposed human computer interface integrates the advantages of facial expression based interfaces and head motion based interfaces to address the problem of small range of head motion for low resolution of camera. An efficient user experience is achieved by a non-linear function mapping the motion of the nose to the motion of the cursor. The mouth status enables the subjects to adjust their poses of the head conveniently, and to efficiently provide commands.

When compared with other interface devices for people with severe mobility impaired access to computers, the strength of the proposed system is its convenience (no guardian, no accessory on body, no calibration, and no initialization) and robustness (insensitive to illumination or colour). It allows persons with tetraplegia to use computers efficiently to enrich their lives.

The structure of the rest of this chapter is organized as the following sentences. Section 5.3 introduces the detection algorithm of facial position and expression. Section 5.4 introduces mouse events based on the facial position and expression. Experimental results are presented in Section 5.5. Conclusion is drawn in Section 5.6.

Figure 5.25: Overview of our proposed human computer interface.

5.3 Detection of Facial Position and Expression

As shown in the flow chart of Figure 5.25, the first step of the interface is to remove the background. If the pixel's depth $z(x, y)$ is larger than a maximum threshold T_{max} or smaller than a minimum threshold T_{min} , this pixel is considered as background for our application. The mask function $M(x, y)$ is

$$M(x, y) = \begin{cases} 1(Foreground) & T_{min} \leq z(x, y) \leq T_{max} \\ 0(Background) & otherwise \end{cases} . \quad (\text{Eq. 5.1})$$

We set $T_{min} = 50$ cm since the minimum reliable depth value of the camera for experiments is 50 cm, and set $T_{max} = 150$ cm by assuming that the distance between the

subject and the computer is smaller than 150 cm. The parameters T_{min} and T_{max} are only used to remove the background, and they do not appear in the RDT algorithm. The RDT algorithm is insensitive to T_{min} and T_{max} . After removing the background, the detection algorithm of the facial position and expression is employed.

The position of the nose is used to navigate the cursor along with the commands triggered by the status of the mouth. It is reasonable to use nose position to control the cursor position. As pointed out in [116], the rigid part of the face is suitable for controlling the cursor position while the non-rigid part of the face is suitable for triggering commands. The eyes, as example, belong to non-rigid part. The unintentional movement of eyes as well as other non-rigid part will impact the cursor control. The nose position is more stable than the eyes. When the user rotates his head, one of the eyes is prone to be blocked by other parts. If the user is wearing glasses, the glasses cover the eyes and the eyes are easily impacted by the opaque part of glasses. As a human computer interface for disable people, the motion of head is small. The nose as the peak protrusion of the face could have a larger motion than the eyes when the user rotates the head.

Both the position and the status are detected from a single depth image based on the RDT framework, and the calibration before use is not needed. The proposed detection approach of the position and the status is robust and has low computational complexity. The motivation of the proposed RDT algorithm is that the nose position and the mouth status can be quickly and accurately detected together in the same framework. In the following four subsections, we present four parts of the detection approach: **(1)** the training data of the proposed RDT-based algorithm, **(2)** the novel feature of the proposed RDT-based algorithm, **(3)** the decision method of nose position and mouth status after RDT classification, and **(4)** the pyramid processing. For more details on RDT, we refer the interested reader to [95, 9].

5.3.1 Training Data

In [95], the RDT algorithm was successfully developed to detect the joint positions by the authors. For training, a fixed predefined label was used to label the same body part in the human model no matter what the human poses. Here, the RDT algorithm is further developed to detect two categories of information together, i.e., expressions and positions. In the training phase of our algorithm, the mouth is labelled corresponding to different statuses of the mouth, as shown in Figure 5.26. In other words, the information of both the position and the expression is embedded into labels. Totally, there are five labels for the RDT algorithm, i.e., nose, mouth-open, mouth-close, head and body. The number of labels is small and the labelling is much easier than [95]. The synthetic depth images simulated by the human model can be used for training as in [95]. In addition, the real depth images can also be labelled for training, which corresponds more closely to the depth images in test phase than the synthetic images. The real personal depth images using for training can improve the accuracy for a specific person with tetraplegia. Figure 5.26 shows two examples of real depth images with labels.

As shown in Figure 5.26, the nose is exactly labelled in a small circle. Nevertheless, the mouth is roughly labelled by a polygon and the labels of the mouth in Figure 5.26.a and 5.26.b are different according to different mouth statuses. The underlying reason is that the position of the nose is employed to navigate the cursor, so that the accuracy of the nose position is important. However, the accuracy of the mouth position is not as important as the nose position. In order to avoid overfitting, the pixels along the nose and the mouth are not labelled, as shown as black band in Figure 5.26.

5.3.2 Feature of RDT

Figure 5.27 shows the structure of our RDT. Each split node n has a parameter $p_n = ((\Delta x_n, \Delta y_n), \tau_n)$. $(\Delta x_n, \Delta y_n)$ is the offset with respect to the location of test pixel (x_0, y_0) .

5.26.a:

5.26.b:

Figure 5.26: Labelled images for training. (a) Close the mouth. (b) Open the mouth. Green: mouth-close, red: mouth-open, white: nose, blue: head, brown: body.

Figure 5.27: Structure of our RDT. Each split node n contains a comparison between the feature value f_n and threshold τ_n . Each leaf contains a posterior distribution.

τ_n is the scalar threshold to be compared with the feature value f_n of the test pixel, and each node n has a different value τ_n learned from the training data. p_n is optimized from training. Each comparison decides that the test pixel goes to the left or right branch child of node n . In the end, the test pixel reaches a leaf, which stores the classification information, i.e., a posterior distribution over the labels. For example, in a given leaf, there are 1000 pixels of label l among total 10000 pixels. Thus the posterior distribution value of label l is $1000/10000 = 0.1$. An efficient feature is very important to the RDT. We propose to use the depth difference between the test pixel and one of its neighbours as the feature of RDT, i.e.,

$$f((x_0, y_0) | (\Delta x, \Delta y)) = z(x_0, y_0) - z((x_0, y_0) + \frac{(\Delta x, \Delta y)}{\bar{z}/R}), \quad (\text{Eq. 5.2})$$

where (x_0, y_0) is the test pixel's location in a given test image, $(\Delta x, \Delta y)$ is the location offset value with respect to (x_0, y_0) , The pixel (x, y) depth value is $z(x, y)$, $\bar{z}$ is a normalization variable, and R is a constant. The offset value is normalized by $\frac{1}{\bar{z}/R}$.

In the training phase, $\bar{z}$ is equal to the depth of the nose in the image. After training, the offset parameter $(\Delta x, \Delta y)$ is normalized by $\frac{1}{\bar{z}/R}$, and stored as some $\text{RDT}_{\bar{z}}$ w.r.t. different $\bar{z}$. Thus, for a given $\text{RDT}_{\bar{z}}$, the feature becomes

$$f((x_0, y_0) | (\Delta x_{\bar{z}}, \Delta y_{\bar{z}})) = z(x_0, y_0) - z((x_0, y_0) + (\Delta x_{\bar{z}}, \Delta y_{\bar{z}})), \quad (\text{Eq. 5.3})$$

where $(\Delta x_{\bar{z}}, \Delta y_{\bar{z}})$ has been normalized offline.

In the test phase, $\bar{z}$ is used as the index to look up the corresponding $\text{RDT}_{\bar{z}}$. Before test, the depth of the nose is unknown. The depth of the test pixel is employed as $\bar{z}$ to look up the corresponding $\text{RDT}_{\bar{z}}$. It is reasonable to use $z(x_0, y_0)$ as $\bar{z}$: **(1)** If the test pixel is on the nose, $z(x_0, y_0)$ is equal to the depth of the nose; **(2)** If the test pixel is on the face, the error between $z(x_0, y_0)$ and the depth of the nose is very small compared to the depth of the nose. When a rough position of the nose for the current frame is

5.28.a:

5.28.b:

5.28.c:

5.28.d:

Figure 5.28: Results of RDT classification. (a)(c) Input depth images shown as mesh after removing pixels with near/far distance. (b)(d) Labels map after RDT classification, each colour stands for the highest probability label, green: mouth-close, red: mouth-open, white: nose, blue: head, brown: body. (a)(b) Close the mouth. (c)(d) Open the mouth.

known, e.g., from the nose position of the previous frame, the rough depth of the nose is employed as $\bar{z}$. For this case, the selection of $\text{RDT}_{\bar{z}}$ takes only once for one test image.

Furthermore, when the size of the image is fixed, the two dimensional offset $(\Delta x_{\bar{z}}, \Delta y_{\bar{z}})$ can be converted to a one dimensional value, resulting in less computational complexity.

5.3.3 Decision Method of Nose Position and Mouth Status

As shown in Figure 5.28, the output of the RDT classifier is a distribution D_{il} over the labels l for each test pixel (x_i, y_i) . After the per-pixel classification, a mean-shift algorithm and a voting algorithm will be used to extract the positions and the mouth status respectively.

5.3.3.1 Position

To extract the positions, we use a mean-shift clustering approach based on the distribution. A weighted kernel is included in this process. For the positions, the density estimator is defined as

$$P_l \propto \sum_{i=1}^M w_{il} k(L - L_i) z(L_i)^2, \quad (\text{Eq. 5.4})$$

where the position of label l is denoted as P_l , the total amount of test pixels of the test depth image is denoted as M , the weighting of the test pixel is denoted as w_{il} , a kernel function is denoted as $k(\bullet)$, e.g., Gaussian kernel, a 3D location is denoted as L , the 3D location of the test pixel (x_i, y_i) is denoted as L_i , and $z(L_i)$ is the depth of L_i . w_{il} is equal to the distribution D_{il} from the classifier of RDT. For the mouth position, the mouth label merges labels of different statuses of the mouth.

5.3.3.2 Mouth Status

We use a voting method to extract the status of the mouth. Its voting metric is

$$S_m^* = \arg \max_{S_m} (w_{S_m} \sum_{i=1}^M w_{iS_m} k(P_m - L_i) z(L_i)^2), \quad (\text{Eq. 5.5})$$

where $S_m \in \{\text{mouth-close, mouth-open}\}$ is the label of the mouth status, w_{S_m} is the weighting of the mouth status (learned from the training data), w_{iS_m} is the weighting of the test pixel (x_i, y_i) , and P_m is the mouth position. The mouth status result S_m^* is determined by the status with larger voting value. Compared with the landmark-based

Figure 5.29: Pyramid processing when the rough positions of the nose and the mouth are unknown.

mouth status detection methods, the proposed method is free from the challenges of (1) facial landmarks extraction, and (2) the mouth status inference based on the extracted landmarks. Thus, it is more efficient than landmark-based methods, e.g., [116].

5.3.4 Speed Up by Pyramid Processing

As a computer interface, it should be implemented with only a little CPU resource, reserving CPU resource for other applications. The detection algorithm mentioned above is performed on all foreground pixels. In practice, the runtime of the proposed detection algorithm can be speeded up by pyramid processing.

The nose and the mouth only occupy small areas in the depth image, even when compared with the foreground. Thus, the nose position and the mouth status can be detected by classifying a small amount of pixels via the RDT algorithm if the rough positions of the nose and the mouth are known. When the rough positions are unknown, the pyramid processing is implemented as shown in Figure 5.29. The rough positions of the nose and the mouth can be quickly detected by down sampling approach. After classifying pixels, the mean-shift algorithm is used to extract the head position, merging labels of head, mouth-close, mouth-open and nose. The head position is the starting point

of the mean-shift for the mouth position, merging labels of mouth-close and mouth-open. Using the mouth position, the voting algorithm is employed to detect the mouth status. And the mouth position is the starting point of the mean-shift for the nose position. The accuracy of the nose position is important. Hence, the finer position of the nose is further detected by classifying neighbourhood pixels around the rough position of the nose without down sampling approach. The number of neighbourhood pixels is much smaller than that of the entire foreground. The speed of the detection operation can be improved several times by this pyramid processing.

The density estimators for the positions and the relationship, e.g., the 3D distance, among the positions of head, mouth and nose are used to measure the confidence level of detection results.

If the previous frame can provide the reliable rough positions of the nose and the mouth for the current frame, only the neighbourhood pixels around these two positions are classified by the RDT algorithm. For the mouth, down sampling approach is used since the size of the mouth is large enough. For the nose, the detection is performed without down sampling approach.

With the pyramid processing, the proposed detection algorithm has very low computational complexity, and the potential for hardware implementation in a camera.

5.4 Mouse Events Based on Facial Position and Expression

The operation function of the proposed interface is presented in this section. The nose position and the mouth status are combined to navigate the computer cursor and trigger interactive commands.

5.4.1 Mapping Function of Motion

In [116], the authors tested three cursor control modes, i.e., joystick mode, direct mode and location mode. Their experimental results showed that the best performance mode is the location mode, in which the behaviour is very similar to the operation of the mouse. The location mode provides users with the freedoms of controlling the cursor speed, the acceleration and the motion insensitive to the directions. Therefore, the location mode based on the 3D position of the nose is chosen as the cursor control mode of FM.

A bottleneck of the interface is the depth image resolution since the interface utilizes the small amplitude motion of the head. If the function mapping the nose motion to the cursor motion is linear, e.g., [116], a small mapping gain will lead to a small cursor motion range, so that the cursor cannot cover the whole screen. However, a large mapping gain will convert a noise to a large motion of the cursor, which will affect the cursor precision and the user experience. Therefore, the precision and range of cursor motion become a problem of trade-off.

To address the problem mentioned above, we propose to use a non-linear mapping function, i.e.,

$$D = Gd^{p_1} \left(1 - \exp \left(- \left(\frac{d}{n} \right)^{p_2} \right) \right), \quad (\text{Eq. 5.6})$$

where D represents the space distance from the current cursor position to the last cursor position, G is the mapping gain, d is the space distance between the current nose position and the last nose position in the xy plane, p_1 and p_2 are powers, and n is a coefficient relative to the noise level. The term $(1 - \exp(-(\frac{d}{n})^{p_2}))$ is used to set $\lfloor D \rfloor = 0$ when d is a small value, which can efficiently eliminate the jitter of the cursor to make the interface more comfortable and more accurate. The total mapping gain increases with the speed of the nose motion. When the cursor is far from its target, the user tends to move his nose faster. The resulting higher mapping gain can help speed up the cursor to

approach the target. Thus, compared to the linear mapping function, the user can move the cursor at a larger range without a larger amplitude of head motion. When the cursor is close to its target, the user can finely adjust the cursor location by moving his nose at low speed. The cursor location is controlled by the nose position and its motion speed, allowing more degree of freedom to control the cursor than the simple linear mapping function.

5.4.2 Commands Triggered by Mouth Status

The close and open statuses of the mouth are used to enable and disable the motion of the cursor, respectively. When the pointing target has a very large distance from the cursor, the subject could open the mouth, so that he can turn his head back to a comfortable position before continuing to move the cursor to the target. Hence, the mouth status enlarges the range of the cursor motion. When the subject opens his mouth for disabling the cursor motion, the mouth status can allow the subject to conveniently adjust his head to a more comfortable posture without moving the cursor. Furthermore, the mouth status can be used to quickly stop the cursor and fix the location of the cursor to avoid overshooting so as to enhance the pointing precision when the cursor is moving on the cursor target.

Commands can be predefined based on the mouth status combined with temporal information. For example, the mouse double click event is provided by open-close-open-close mouth status sequence in a short time, and the single click is provided by open-close-open. All the mouth actions do not interfere with users' normal visual tasks.

5.5 Experiments

In this section, we experimentally evaluate the performance of both the detection method and the interface operation.

5.30.a:

5.30.b:

Figure 5.30: Examples from Bosphorus 3D face database. They are shown as mesh. (a) Negative. (b) Positive .

5.5.1 Evaluation of Detection Algorithm

To verify our extraction algorithm, we did the experiments by using the Bosphorus 3D face database [94]. Twenty challenging poses, i.e., the pose type Lower-Face-Action-Units (LFAU), were selected as training and test samples. The negative label includes nineteen poses and the positive label includes the mouth stretch LFAU_27, as shown in Figure 5.30. Totally, there were 105 subjects, 1876 samples of negative and 105 samples of positive.

The 10-fold cross-validation were used in the experiments. The database was randomly divided into ten subsets of approximately equal size. Each subject appeared in only one subset. The training and test were done ten times, each time nine subsets for training and the rest one for test. The classification accuracy reported was averaged of

the ten different random splits.

Table 5.1: Comparison of detection algorithms. “*p*” denotes the use of pyramid processing.

	[95]	[9]	Ours	Ours^p
FN of mouth	2	2	2	2
FP of mouth	6	5	5	5
Total errors of mouth	8	7	7	7
Error rate of mouth(%)	0.40	0.35	0.35	0.35
Mean error of nose(mm)	3.2	2.4	1.9	1.9
Time(ms)	5.4	2.9	1.5	0.24
Frame rate(fps)	185	345	667	4167

Table 5.1 shows the accuracy and speed of detection algorithms based on difference features of RDT. In Table 5.1, False Negative (FN) means the number of positive events detected as negative events. False Positive (FP) means the number of negative events detected as positive events. We assume the face of Bosphorus database is 0.5 m from the camera. In Table 5.1, “*p*” denotes the method that uses the pyramid processing to speed up the algorithm. For the interface application, the cursor is navigated in the xy plane. Therefore, the mean error of the nose position is the error in the xy plane. The speed was obtained from a single thread program implemented in C++. The test PC had an Intel Core i7 with 8 GB RAM. The number of pixels per image of Bosphorus data is approximated by the number of foreground pixels from Kinect v2 when the subject is sitting 0.8 m from the camera. Thus, the speed in Table 5.1 is close to the speed of the detection by using Kinect v2.

Comparing the first and second columns of data in Table 5.1, the method of [9] has better results in all terms than the method of [95], except FN. Comparing the second and third columns, the proposed method is $1.9\times$ faster than the method of [9]. The mean error of the nose position is reduced by 21% while the error results of the mouth status

are the same compared with the method of [9]. From the last two columns, we can note that the pyramid processing can dramatically speed up the algorithm, i.e., $6.2\times$ faster, without any loss of accuracy. When only the neighbourhood pixels around the mouth and the nose are classified by the RDT algorithm, the frame rate can be higher than 1×10^4 fps.

Table 5.2 shows the error results of the mouth status compared with Support Vector Machine (SVM) classifiers [13], which are used to detect the mouth status by using the ground truth landmarks of the mouth. In Table 5.2, the best results of SVM classifiers are 12 total errors. The results of the proposed method are 7 total errors. Although the SVM classifiers use ground truth landmarks, the total errors results of the proposed method are better. If the landmarks are detected with error by an algorithm, the results of SVM classifiers should be degenerated.

Table 5.2: Comparison with SVM.

Error of mouth	Proposed	SVM-quadratic	SVM-plynomial	SVM-linear	SVM-rbf
FN	2	6	4	1	35
FP	5	6	12	17	9
Total errors	7	12	16	18	44
Error rate (%)	0.35	0.61	0.81	0.91	2.22

The number of subjects in the dataset is large, and the subjects vary a lot in their genders, races, beard/moustaches and short facial hairs. In the experiments, the test subject does not appear in the training data. The cross-validation is used. Therefore, the test results are insensitive to different people and settings.

To further demonstrate the performance of the proposed facial position/expression extraction algorithm, a video demo could be downloaded ¹. Figure 5.31 5.32 5.33 and 5.34

¹<http://www.ntu.edu.sg/home/elpchau/FM.mp4>

are some samples from the video to show that the algorithm can resist many variations, such as glasses, facial hairs and rotation of the head.

Figure 5.31: Without wearing glasses / facial hairs.

Figure 5.32: Wearing glasses.

Figure 5.33: With facial hairs, the hair covers the whole forehead.

With regard to the glasses, the glass lenses are transparent to the Kinect, and the depth information is still presented the eyes. The glasses frame does not affect the proposed detection method, as shown in Figure 5.32.

With regard to facial hairs, even with the hair covering the whole forehead, the interface still detects the nose and mouth well, as shown in Figure 5.33. If the hairs are

Figure 5.34: Different rotation of the head.

outside of the nose, mouth and cheekbone region, the interface still works well. But if the hairs cover the nose significantly, the system cannot extract the nose position.

With regard to the rotation of the head, the interface is also robust, as shown in Figure 5.34. The interface is designed for the subject to use computer, and the subject should see the computer screen well. As shown in bottom left of Figure 5.34, even when the field of view of both eyes is blocked by the face, the interface can still detect the nose and mouth well.

5.5.2 Evaluation of Interface Operation

The interface operation was estimated according to Fitts' Law [28] and ISO/TS 9241-411 standard [43].

5.5.2.1 Task

The ISO/TS 9241-411 standard establishes uniform guidelines and test procedures to evaluate computer pointing interfaces, in which the multi-directional tapping task is used, as shown in Figure 5.35. The procedure of the tapping task was to point at and select the circular target indicated in blue as shown in Figure 5.35.b. Subjects were instructed to move the cursor to the centre of the target as quickly and precisely as possible, and to select the target as quickly as possible (click the left button using the mouse, dwell on the pointing target on the screen for 500 ms, as suggested in [130, 44], using FM or Camera Mouse). After the tapping task, each subject was interviewed and

given a questionnaire, which was defined in ISO/TS 9241-411 standard to evaluate the comfort in the use of interfaces and was adapted to involve the specific body parts used during the experiments.

The primary parameters of a target are the distance (D) and width (W) of the target. The distance of a target is defined by the distance between the centres of two continually appearing targets, as shown in Figure 5.35.b. In the task, three different distances (D) and three different widths (W , diameters as shown in Figure 5.35.b) are used to generate six D-W pair conditions. Table 5.3 shows the design values, which include all the conditions for multi-directional tasks of [126] and [44]. Each D-W condition randomly appeared during experiments.

Table 5.3: Design values of tapping task.

D (pixels)	100	305	305	534	534	534
W (pixels)	57	76	57	76	57	30
ID (bits/s)	1.46	2.33	2.67	3.00	3.37	4.23
Our	✓	✓	✓	✓	✓	✓
[44]		✓	✓	✓	✓	
[126]			✓	✓	✓	

5.5.2.2 Performance Metrics

According to ISO/TS 9241-411 standard, the metric for comparison is throughput (TP), which includes both the accuracy and speed of user operation performance. We used one additional metric, i.e., Task Completion Time (TCT) as used in [126, 42, 44].

Throughput: TP indicates the amount of information that the subjects deliver to the computer via the human computer interface. Higher TP is better. TP can be formulated as

$$TP = \frac{ID}{MT}, \quad (\text{Eq. 5.7})$$

where MT is the mean movement time over all the trails for the same condition targets, and ID is the index of difficulty of the target. The movement time refers to the time that

5.35.a:

5.35.b:

Figure 5.35: The Graphical User Interface (GUI) screen used for the tapping task in our experiments. (a) 90 possible targets and their order. (b) Targets turn blue one at a time.

the subject spends moving the cursor to point a target from the centre of the last target. The movement time should not count the selecting or dwelling time for pointing target selection and the time of initiation delay before moving the cursor [126, 100]. Shannon's formula is used to define ID as

$$ID = \log_2\left(\frac{D}{W} + 1\right). \quad (\text{Eq. 5.8})$$

ID is measured in bits. ID values used in our experiments are shown in Table 5.3.

The mean TP is calculated as

$$TP = \frac{1}{S} \sum_{i=1}^S \left(\frac{1}{C} \sum_{j=1}^C \frac{ID_{ij}}{MT_{ij}} \right), \quad (\text{Eq. 5.9})$$

where S represents the number of subjects, and C is the number of target conditions.

Task Completion Time: TCT represents the total time used for completing each round. Lower TCT is better. TCT shows the overall users' control over the movements of the cursor. TCT is counting from the appearance of the first target, and ending at the last target disappearance.

5.5.2.3 Interfaces

The openCV was used to develop the GUI of the experiments. The size of the computer LCD monitor was 14.0", and the resolution was 1600×900 pixels. The pointing task window had a size of 610×610 pixels with white background. The subjects were sitting 80 cm from the monitor.

Three interfaces were used, i.e., FM, Camera Mouse and a standard optical mouse. For FM, the commercial depth camera Kinect v2 with 512×424 pixels was used and mounted above the LCD monitor. Camera Mouse is a popular interface for disabilities, and has been downloaded more than 2.5 million times to date [108]. It was used to compare with the proposed interface FM. The input RGB camera of Camera Mouse was a 1280×720 pixels HD webcam. For Camera Mouse, the subject had to fix the face to freeze the cursor on the target for 500 ms to select the target. In contrast, for FM, the subject could open the mouth or fix the nose to freeze the cursor. The standard mouse is a good baseline to validate our experimental methods and data analyses as [126, 44]. For the mouse, the range of performance measures has been well established between 3.7 bits/s and 4.9 bits/s [100, 126, 42, 44]. For each subject, the test order of different interfaces (FM, Camera Mouse and the mouse) was randomized. The subject had to finish all tests of one interface, and then started to test another interface. Before the test, the subjects practiced FM or Camera Mouse for more than one hour, and the mouse for more than ten minutes. During the test, subjects were always working in a good illumination condition with suitable colour background to eliminate the environmental effect for Camera Mouse.

Table 5.4 shows the details of the test trials. There were 90 trials each round. Each interface was tested for twenty rounds per subject. Sixteen able-bodied participants (10 males, 6 females, graduate students and staffs from Nanyang Technological University, age range 23~36 years old) took the tests. They had no prior experience to use head or

face interface, and they use PC for more than 5 h daily. The total number of trials was 86400.

Table 5.4: Test trials in our experiments.

Trials	Rounds	Interfaces	Participants	Total trials
90	20	3	16	86400

5.5.2.4 Results

Table 5.5: Experimental results of operation evaluation.

	Camera Mouse	FM	Mouse
$TP(\text{bits/s})$	1.52 ± 0.12	3.26 ± 0.26	4.39 ± 0.36
$TP_n(\%)$	100	215	289
$TCT(\text{min})$	4.68 ± 0.35	2.88 ± 0.25	1.53 ± 0.12
$TCT_n(\%)$	100	61	33

The operation results shown in Table 5.5. The mouse throughput is 4.39 bits/s, which is in line with many other experiments [100, 126, 42, 44] and indicates that our experiments are adequate in terms of the data analysis, GUI functionality and methodology. TP and TCT of FM are 115% and 39% better than that of Camera Mouse, respectively. In the test of Camera Mouse, there were a few times of loss tracking the selected point due to feature drift, and re-selection of the tracking point was required. For these cases, the unsuccessful rounds were discarded and the subjects were required to do the same rounds again. The proposed method avoids this tracking loss problem. In short, Table 5.5 demonstrates that the proposed method is remarkably better than Camera Mouse. Four major reasons contribute to the improvement:

- (i) Our approach enables subjects to conveniently adjust the head to comfortable poses by themselves. For Camera Mouse, the head pose relative to the cursor is adjusted only at the beginning of operation.

Figure 5.36: Benchmarking ATs in terms of normalized TP with respect to the mouse TP . EyeTracker (eye) [62], SmartNav (head) [36], dTDS (tongue+speech) [42], TDS (tongue) [126], HeadTracker (head) [36], LCS (lip) [44], Head Orientation (head) [120], EMG (neck muscle) [120].

- (ii) The proposed mapping function speeds up the cursor motion when the target is far from the cursor, and compresses the noise when the target is close. For Camera Mouse, the mapping function is linear.
- (iii) The proposed detection method can detect the expression and the position from a single image while resisting the Camera Mouse’s “feature drift” problem in feature tracking phase, which is a common problem of tracking algorithms. For Camera Mouse, when the feature drifts, the subjects feel uncomfortable.
- (iv) The proposed mouth status allows the subjects to stop the cursor quickly. Camera Mouse needs a longer time to stop the cursor when the cursor is moving fast.

To better understand the performance of FM, Figure 5.36 benchmarks FM against some other ATs in terms of their normalized TP with respect to the mouse TP . In Figure 5.36, EyeTracker [62] needs to fix the head of user, and SmartNav [36] uses

5.37.a:

5.37.b:

5.37.c:

Figure 5.37: Cursor trajectories of the tapping task with $D=534$ and $W=57$. Red dots are the selected locations. (a) Camera Mouse. (b) FM. (c) Mouse.

an infrared camera to track a reflector placed on the head of user. FM has the best performance in all of the ATs.

Figure 5.37 shows some cursor trajectories of three interfaces in the tapping task. The critical differences among the three trajectories are the precision while pointing to the selected locations. All three methods are suffering from overshooting problem. However, the proposed FM method has similar performance as the mouse but outperforms Camera Mouse. The reason is that FM and the mouse can stop the cursor in time. For FM, the

Figure 5.38: TP for different target directions in the tapping task with FM, Camera Mouse and the mouse. Top: original TP . Bottom: Normalized TP by mean TP .

subject can open his mouth to stop the cursor. Moreover, the term $(1 - \exp(-(\frac{d}{n})^{p_2}))$ of mapping function can effectively eliminate the jitter of the cursor. In contrast, Camera Mouse cannot stop the cursor in time, and it is easy for the cursor to overshoot the target. To avoid overshooting, the speed has to be reduced, resulting in lower TP .

Figure 5.38 shows TP for different target directions in the tapping task with FM, Camera Mouse and the mouse. The top figure shows the original various performance by the target direction, and the bottom figure shows the normalized performance with

Figure 5.39: Comparison of the interfaces in terms of MT versus ID and their regression models based on Fitts' law.

respect to the mean performance. FM has approximately equal TP for different target directions, similar to the mouse and slightly better than Camera Mouse, which illustrates the performance of FM is independent of target direction.

Figure 5.39 shows the relationship between MT and ID . The more linear the relationship is, the better the performance of the interface fits the Fitts' law. From Figure 5.39, we can see that the coefficient of determination R^2 of FM is the same as that of the mouse, i.e., $R^2 = 0.99$, which is higher than that of Camera Mouse, i.e., $R^2 = 0.96$. The intercepts for all three interfaces are inside the normal range $-0.20 \sim 0.40$ s [100].

Figure 5.40 shows the qualitative result of the questionnaire for FM and Camera Mouse. The questions are rated from 1 to 7, where 1 and 7 are the least and most favourable responses respectively. One-way analysis of variance (ANOVA) with pairwise comparison is used to analyse the differences between the results of FM and Camera Mouse. Those comparisons that are significantly different ($p < 0.05$) are denoted with “*” on the horizontal axis. Only the “Neck fatigue” and “Accurate pointing” are not significantly different. FM receives higher scores in all terms except the “Mouth fatigue” since Camera Mouse does not require the mouth motion. The subjects prefer FM to Camera Mouse in terms of “Like to use” in Figure 5.40 that the response is 6.1 vs

Figure 5.40: Assessment results of the questionnaire for FM and Camera Mouse. Higher score is better. Response 7 is the most favourable while response 1 the least favourable. Those comparisons that are significantly different ($p < 0.05$) are denoted with “*” on the horizontal axis.

4.9, which is significantly different. In addition, the average qualitative score of all the responses for FM is 5.7, which is significantly different and 14% better than that for Camera Mouse, i.e., 5.0.

5.6 Conclusion

In this chapter, a human computer interface is proposed for persons with tetraplegia based on facial position and expression information from a depth camera. The proposed RDT-based detection algorithm is able to detect the nose position and the mouth status from a single depth image with accuracy and efficiency, without the requirement for calibration or initialization. The interface combines the advantages of facial expression based interfaces and head motion based interfaces. Furthermore, the interface is independent of colour and illumination changes, which can enhance the robustness compared with interfaces based on RGB cameras. Experimental results show that the proposed detection algorithm is faster and more accurate than the state-of-the-art algorithms.

The quantitative and qualitative results show that the proposed system outperforms the state-of-the-art ATs by 9~58% and 14% in terms of TP and the average qualitative score, respectively.

Chapter 6

Conclusion and Future Work

This chapter draws the conclusion of this thesis and presents the opportunities for further research.

6.1 Conclusion

Motion capture systems based on a single depth camera can provide highly accurate and robust compact solutions for extensive applications in modern daily life. The development of both hardware and software of depth cameras has progressed rapidly during the past few years. Nowadays, some depth cameras have very low costs, but their quality is good enough for many applications. Such depth cameras can work in dark environments since the cameras use the active infrared LED and do not interrupt the subjects, for example, when they are sleeping at night. They can well address some problems faced by the traditional cameras, such as the depth ambiguities, silhouette ambiguities, shadow affection, color variation, intention variation and so on. The depth image can provide faithful 3D information for many applications.

However, as a markerless motion capture technique, the algorithms are required to track the native features on the body parts. However, it is difficult to track such features accurately and robustly since the features are very complex and have large variations.

To achieve high practical accuracy and robustness, very high computational complexity is imposed, which is a research gap for real time performance.

The state-of-the-art motion capture algorithm, i.e., the RDT algorithm, has filled this gap. The depth image is not affected by color and lighting illumination changes at home; hence, the large synthesized realistic depth image database for training of the RDT algorithm can be built for customized applications. However, one of the challenges of the RDT algorithm is the extremely high computational complexity of the training phase. To make the RDT algorithm more practical, we proposed a fast training algorithm based on a compact feature.

With the improved RDT algorithm, we proposed two motion analysis frameworks based on the depth camera motion capture. The first motion analysis framework is the fall detection system. It addresses multiple issues, i.e., low cost, high accuracy, robust and real time performance. It can automatically detect the human fall events at home. Compared with existing approaches, the proposed system is more accurate and more robust.

The second proposed system is a human computer interface for people with tetraplegia, who have difficulties to use standard input devices of computers. Thus, it is of great importance to develop assistive technologies to enrich their lives. The proposed human computer interface addresses the convenient, robust, low cost and low computational complexity issues. It uses the head motion and face expression to control the computer without use of hands. It is a promising option for the people with tetraplegia to interact with the computer.

More conclusions of the three major contributions are drawn as follows.

6.1.1 Fast Training Algorithm

The depth image based approach allows the acquisition of a large dataset of realistic depth images for the training of the RDT algorithm. The RDT algorithm is fast, highly

accurate and robust in the test phase. However, it has a very highly computational complexity in the training phase. Based on the compact feature, we find that the metric values in the training node can be drawn as a smooth curved surface. This property allows us to optimize the parameter of each node of the RDT by using search algorithms. Thus, we do not need to train a large amount of the candidate features. The proposed fast training algorithm can reduce the computational complexity remarkably compared to the existing method. The theoretical analysis shows that the speedup is linear to the reduction of the candidate number. In the test phase, the RDT algorithm does not lose accuracy in test results when the fast training algorithm is used. With the fast training algorithm, the RDT algorithm for motion capture based on depth images can be much easier to implement for customized applications.

6.1.2 Fall Detection

We developed the markerless motion capture technique for fall detection for elderly people.

Fall events pose on extremely high health risk for elderly people. Thus, fall detection is an important issue in the health care system. The traditional RGB camera based methods for fall detection are not robust due to depth ambiguities, color variation and illumination variation since the RGB images measure the color and the intensity of light in 2D images. Especially for a single RGB camera, it is hard to provide 3D information for fall detection in realtime. But, the human motion is in 3D. The fall events might happen during the day or night, and the ambient illumination is very difficult to predict. The robustness of the fall detection based on the RGB camera is restricted by the 2D information and the complex daily environment. The depth camera can provide 3D information of objects and is not sensitive towards color and illumination changes. Thus, the depth camera is more robust than the RGB camera for fall detection.

We use the RDT algorithm to capture the human motion for fall detection. The result of the human body part extraction of the existing algorithm is degenerate for non-standing poses. We combine time information to correct the human pose so as to improve the performance of extraction. The difference of human pose in two consecutive frames is very small since the camera output frame rate is high. From the previous human pose, we can predict the human pose in the current frame and correct the pose for human body part extraction. After pose correction, the human body part extraction becomes more robust.

The markerless motion algorithm provides the 3D human motion. Then the fall and non-fall motions are classified by a support vector machine, which is very robust for two-class classification. The input vector of the SVM classifier is the head trajectory. The head trajectory includes different types of information, consisting of velocity, acceleration, the distance between the head and the floor and other hidden information. The SVM classifier can automatically learn the maximum margined separated hyperplane in high dimensional feature space. Compared to a classifier with manually set threshold values, e.g., a threshold of the distance between the 3D silhouette centre of the foreground pixels and the floor, the SVM classifier can be more robust.

The experimental results show that the proposed approach is more accurate and more robust in fall detection than the existing approaches.

6.1.3 Human Computer Interface

The second presented application of motion capture is the human computer interface for hand-disabled people. Many existing assistive technologies require the users to wear some special accessories. We proposed a facial position and expression mouse system by using a single depth camera. Although the color of each organ on the face, e.g., the nose, the mouth and the cheek, is very similar, the surface geometry of each organ is very

different. The depth camera can capture the 3D surface. In the depth images, we assign the nose as a single label and the open and close statuses of the mouth as two different labels to train the RDT. During the test phase, the nose position and the mouth status can be detected efficiently and accurately. This human computer interface combines the advantages of facial expression based interfaces and head motion based interfaces. To be specific, the position of the nose is used to navigate the computer cursor, and the mouth status provides interaction commands conveniently without interference with normal human vision.

One of the challenges for the camera based interface is that the resolution of camera is low. The nose motion is very small when rotating the head. To address this issue, we use a nonlinear function to map the nose position to the cursor position. When the head motion is fast, the mapping gain is large. When the head motion is slow, the mapping gain is small. It can allow the user to move the cursor farther by faster head motion and to point the cursor target with higher precision by slower head motion. The mouth status can enable or disable the movement of the cursor, which allows the user to conveniently adjust the relationship between the head and the computer cursor in order to achieve a more comfortable pose during use.

In addition, the proposed detection algorithm of the interface has very low computational complexity. The detection algorithm could easily be embedded into the camera and achieve widespread use among disabled people.

The results of experiments, which are done according to ISO standard, show that the proposed human computer interface achieves high performance in terms of both quantitative and qualitative assessments.

6.2 Future Work

Although this thesis has addressed some challenging issues, there are still some open problems on these topics. The future work for fall detection and the human computer interface includes the fall risk estimation and the GUI system of human computer interface, which are explained in the following text.

6.2.1 Fall Risk Estimation

Fall detection is a posterior processing after a fall event. To further improve the quality of life for elderly people, the estimation of fall risk is another good solution. We can measure the health status of the elderly people. If the subject is estimated to be in high fall risk, we can make plans to prevent the fall accident. In [60], the authors compared three methods to estimate the fall risk by some predefined non-daily actions, which are proposed to be used in clinical practice. In [6], the authors proposed a sit-to-stand measurement for in-home monitoring. They used the 3D voxels of the human body to measure sit-to-stand time for fall risk estimation. And the 3D voxels were created from two orthogonal RGB cameras. However, the voxel analysis relies on the 3D voxel model of the whole body, and cannot overcome the occlusion problem of human body parts, e.g., the back of the chair blocking the human hip and waist. If one of these two cameras misses some body parts of the human, it is difficult to build up a 3D voxel model.

Based on the skeleton extraction by using depth camera, it has a great chance to achieve better results. The 3D information of the depth image does not rely on any other camera. To estimate the fall risk, we can detect the head, which is the most visible part of the body. By analysing the 3D head trajectory, we can measure the sit-to-stand time for the fall risk estimation.

6.2.2 GUI System of Human Computer Interface

The conventional Graphical User Interface (GUI) is designed based on the standard input devices. For example, the cursor is designed to be used with the mouse. To further improve the performance of our proposed human computer interface, a special GUI for our interface is a potential solution. Here are some concepts of the future GUI:

- (i) The cursor active area. The normal cursor for the mouse can have very high precision of pointing for general purpose use. For some disabled persons, a very high precision may be unnecessary. For this case, a large cursor active area is useful: if the cursor is close to a target button but not over it, the single click command can also activate the button. By designing a better cursor active area function, we can improve the performance of the interface in terms of both efficiency and accuracy.
- (ii) A special feedback GUI. To help the user to know the cursor status, it is useful to design a special feedback GUI, such as the closest button changing to different color. Audio signals can be incorporated into the feedback GUI.
- (iii) Extension of the interaction commands. Within a short period of time, the mouth status sequence cannot provide many commands. To extend existing commands, new command methods should be designed, which is similar to use a right click button in a mouse.

List of Publications

- [1] **Z.P. Bian**, J. Hou, L.P. Chau, and N. Magnenat-Thalmann, “Fall Detection Based on Body Part Tracking Using a Depth Camera,” *IEEE Journal of Biomedical and Health Informatics*, 2014. (previously known as IEEE Transactions on Information Technology in Biomedicine, TITB)
- [2] **Z.P. Bian**, J. Hou, L.P. Chau, and N. Magnenat-Thalmann, “Facial Position and Expression Based Human Computer Interface for Persons with Tetraplegia,” *IEEE Journal of Biomedical and Health Informatics*, 2015. (previously known as IEEE Transactions on Information Technology in Biomedicine, TITB)
- [3] **Z.P. Bian**, J. Hou, L.P. Chau, and N. Magnenat-Thalmann, “Human Computer Interface for Quadriplegic People Based on Face Position/gesture Detection,” *the 22nd ACM International Conference on Multimedia*, 2014.
- [4] **Z.P. Bian**, L.P. Chau, and N. Magnenat-Thalmann, “Fall detection based on skeleton extraction,” in *Proceedings of the 11th ACM SIG-GRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, 2012.
- [5] **Z.P. Bian**, L.P. Chau, and N. Magnenat-Thalmann, “A depth video approach for fall detection based on human joints height and falling velocity,” in *International Conference on Computer Animation and Social Agents*, 2012.
- [6] J. Hou, **Z.P. Bian**, L.P. Chau, Y. He, and N. Magnenat-Thalmann, “Restoring Corrupted Motion Capture Data via Jointly Low-Rank Matrix Recovery”, in *Proc. IEEE International Conference on Multimedia & Expo (ICME)*, 2014.

References

- [1] H. Alamgir, S. Muazzam, and M. Nasrullah. Unintentional falls mortality among elderly in the united states: Time for action. *Injury*, 43(12):2065 – 2071, 2012.
- [2] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder. A smart and passive floor-vibration based fall detector for elderly. In *IEEE International Conference on Information and Communication Technologies (ICITA)*, pages 1003–1007, 2006.
- [3] E. Angel. The piercing bible: The definitive guide to safe body piercing. Crossing Press, 2009.
- [4] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier. Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution. *IEEE Transactions on Information Technology in Biomedicine*, 15(2):290–300, 2011.
- [5] A. Baak, M. Muller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1092–1099, Nov 2011.
- [6] T. Banerjee, M. Skubic, J. Keller, and C. Abbott. Sit-to-stand measurement for in-home monitoring using voxel analysis. *IEEE Journal of Biomedical and Health Informatics*, 18(4):1502–1509, July 2014.
- [7] A. Bergland and T. B. Wyller. Risk factors for serious fall related injury in elderly women living at home. *Injury Prevention*, 10(5):308–313, 2004.

- [8] M. Betke, J. Gips, and P. Fleming. The Camera Mouse: visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(1):1–10, March 2002.
- [9] Z. Bian, J. Hou, L. Chau, and N. Magnenat-Thalmann. Fall detection based on body part tracking using a depth camera. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2014.
- [10] Z. P. Bian, L. P. Chau, and N. Magnenat-Thalmann. A depth video approach for fall detection based on human joints height and falling velocity. In *International Conference on Computer Animation and Social Agents*, May 2012.
- [11] Z.-P. Bian, L.-P. Chau, and N. Magnenat-Thalmann. Fall detection based on skeleton extraction. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 91–94, New York, NY, USA, 2012. ACM.
- [12] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor. A spelling device for the paralysed. *Nature*, 398(6725):297–298, 1999.
- [13] C. M. Bishop. In *Pattern Recognition and Machine Learning*. Springer, 2006.
- [14] D. Brulin, Y. Benezeth, and E. Courtial. Posture recognition based on fuzzy logic for home monitoring of the elderly. *IEEE Transactions on Information Technology in Biomedicine*, 16(5):974–982, 2012.
- [15] D. M. Buchner, M. E. Cress, B. J. de Lateur, P. C. Esselman, A. J. Margherita, P. R., and E. H. Wagner. The effect of strength and endurance training on gaitbalance, fall risk, and health services use in community-living older adults. *The Journal of Gerontology, Series A, Biological sciences and medical sciences*, 52(4):M218–M224, 1997.
- [16] K. Buys, C. Cagniard, A. Baksheev, T. D. Laet, J. D. Schutter, and C. Pantofaru. An adaptable system for RGB-D based human body detection and pose estimation. *Journal of Visual Communication and Image Representation*, 25(1):39 – 52, 2014.

- [17] B. Champaty, J. Jose, K. Pal, and A. Thirugnanam. Development of eog based human machine interface control system for motorized wheelchair. In *2014 Annual International Conference on Emerging Research Areas: Magnetism, Machines and Drives (AICERA/iCMMD)*, pages 1–7, July 2014.
- [18] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy. Wearable sensors for reliable fall detection. In *27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005*, pages 3551–3554. IEEE, 2005.
- [19] J. Cheng, A. Okoso, K. Kunze, N. Henze, A. Schmidt, P. Lukowicz, and K. Kise. On the tip of my tongue: A non-invasive pressure-based tongue interface. In *Proceedings of the 5th Augmented Human International Conference, AH '14*, pages 12:1–12:4, New York, NY, USA, 2014. ACM.
- [20] Creative. <http://sg.creative.com/p/web-cameras/creative-senz3d>.
- [21] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, pages 1–8, June 2007.
- [22] U. DESA. World population prospects: the 2008 revision. 2009.
- [23] V. Devecseri, A. Doka, J. Molnar, and P. Tamas. An ethological motion capture system. In *2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 487–491, Nov 2011.
- [24] C. Doukas and I. Maglogiannis. Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components. *IEEE Transactions on Information Technology in Biomedicine*, 15(2):277–289, 2011.
- [25] S. Epstein, E. Missimer, and M. Betke. Using kernels for a video-based mouse-replacement interface. *Personal and Ubiquitous Computing*, 18(1):47–60, 2014.

- [26] S. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. B. Kang, and T. Paek. Learning to be a depth camera for close-range human capture and interaction. In *Journal ACM Transactions on Graphics (TOG)*. ACM Association for Computing Machinery, July 2014.
- [27] S.-H. Fang, Y.-C. Liang, and K.-M. Chiu. Developing a mobile phone-based fall detection system on android platform. In *Computing, Communications and Applications Conference (ComComAp), 2012*, pages 143–146, jan. 2012.
- [28] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1954.
- [29] H. Foroughi, B. S. Aski, and H. Pourreza. Intelligent video surveillance for monitoring fall detection of elderly in home environments. In *11th IEEE International Conference on Computer and Information Technology (ICCIT)*, pages 219–224, 2008.
- [30] H. Foroughi, A. Naseri, A. Saberi, and H. S. Yazdi. An eigenspace-based approach for human fall detection using integrated time motion image and neural network. In *9th International Conference on Signal Processing, 2008. ICSP 2008*, pages 1499–1503. IEEE, 2008.
- [31] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 755–762, June 2010.
- [32] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi. A depth-based fall detection system using a Kinect® sensor. *Sensors*, 14(2):2756–2775, 2014.
- [33] D. M. Gavrila. Pedestrian detection from a moving vehicle. In *Computer Vision ECCV 2000*, pages 37–49. Springer, 2000.
- [34] L. D. Gillespie, M. C. Robertson, W. J. Gillespie, S. E. Lamb, S. Gates, R. G. Cumming, and B. H. Rowe. Interventions for preventing falls in older people living in the community. *The Cochrane Library* 4, 2009.

- [35] D. Grest, V. Krüger, and R. Koch. Single view motion tracking by depth and silhouette information. In *Image Analysis*, pages 719–729. Springer, 2007.
- [36] S. Guness, F. Deravi, K. Sirlantzis, M. Pepper, and M. Sakel. Evaluation of vision-based head-trackers for assistive devices. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4804–4807, Aug 2012.
- [37] H. D. W. Hansen, J. P. and A. S. Johansen. Bringing gaze-based interaction back to basics. In *Proceedings of HCI International 2001*, pages 325–328, 2001.
- [38] S. Heinrich, K. Rapp, U. Rissmann, C. Becker, and H. H. König. Cost of falls in old age: a systematic review. *Osteoporosis International*, 21(6):892–902, 2010.
- [39] G. Henley and J. E. Harrison. Injury deaths, australia 2004-05. *Australian Institute of Health and Welfare Canberra AIHW cat. no. INJCAT 127*, 2009.
- [40] M. Holte, C. Tran, M. Trivedi, and T. Moeslund. Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments. *IEEE Journal of Selected Topics in Signal Processing*, 6(5):538–552, Sept 2012.
- [41] N. Howe. Silhouette lookup for automatic pose tracking. In *Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04*, pages 15–22, June 2004.
- [42] X. Huo, H. Park, J. Kim, and M. Ghovanloo. A dual-mode human computer interface combining speech and tongue motion for people with severe disabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 21(6):979–991, Nov 2013.
- [43] ISO/TS. Ergonomics of human-system interaction - part 411 evaluation methods for the design of physical input devices, ISO/TS 9241-411:2012. *ISO/TS*, May 2012.

- [44] M. Jose and R. Lopes. Human-computer interface controlled by the lip. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2014.
- [45] W. Josemans, G. Englebienne, and B. J. Kröse. Fusion of color and depth camera data for robust fall detection. In *VISAPP (1)*, pages 608–613, 2013.
- [46] E. Kandel, J. Schwartz, and T. Jessell. Principles of neural science. McGraw-Hill, New Yor, 2000.
- [47] J. M. Kang, T. Yoo, and H. C. Kim. A wrist-worn integrated health monitoring instrument with a tele-reporting device for telemedicine and telecare. *Instrumentation and Measurement, IEEE Transactions on*, 55(5):1655–1661, 2006.
- [48] E. Kantoch, J. Jaworek, and P. Augustyniak. Design of a wearable sensor network for home monitoring system. In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 401 –403, sept. 2011.
- [49] C. Kawatsu, J. Li, and C. Chung. Development of a fall detection system with Microsoft Kinect. In *Robot Intelligence Technology and Applications 2012*, pages 623–630. Springer, 2013.
- [50] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1686–1691. IEEE, 2006.
- [51] M. Kpski and B. Kwolek. Person detection and head tracking to detect falls in depth maps. In *Computer Vision and Graphics*, pages 324–331. Springer, 2014.
- [52] T. Krzeszowski, A. Michalczuk, B. Kwolek, A. Switonski, and H. Josinski. Gait recognition based on marker-less 3d motion capture. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 232–237, Aug 2013.
- [53] S. E. Lamb, E. C. Jorstad-Stein, K. Hauer, and B. C. Development of a common outcome data set for fall injury prevention trials: the prevention of falls network europe consensus. *Journal of the American Geriatrics Society*, 53(9):1618–1622, 2005.

- [54] C. Lau and S. O’Leary. Comparison of computer interface devices for persons with severe physical disabilities. *Amer. J. Occupat. Therapy*, 47(11):1022–1030, Nov 1993.
- [55] B. Law, M. S. Atkins, A. E. Kirkpatrick, and A. J. Lomax. Eye gaze patterns differentiate novice and experts in a virtual laparoscopic surgery training environment. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ETRA ’04, pages 41–48, 2004.
- [56] N. H. Lehment, M. Kaiser, D. Arsić, and G. Rigoll. Cue-independent extending inverse kinematics for robust pose estimation in 3d point clouds. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2465–2468. IEEE, 2010.
- [57] S. Lehtola, P. Koistinen, and H. Luukinen. Falls and injurious falls late in home-dwelling life. *Arch Gerontol Geriatr*, 42(2):217–224, 2006.
- [58] V. Lepetit, P. Laguerre, and P. Fua. Randomized trees for real-time keypoint recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, volume 2, pages 775–781. IEEE, 2005.
- [59] M. Li, T. Yang, R. Xi, and Z. Lin. Silhouette-based 2d human pose estimation. In *Fifth International Conference on Image and Graphics, 2009. ICIG ’09*, pages 143–148, Sept 2009.
- [60] P. Loncomilla, C. Tapia, O. Daud, and J. Ruiz-del Solar. A novel methodology for assessing the fall risk using low-cost and off-the-shelf devices. *IEEE Transactions on Human-Machine Systems*, 44(3):406–415, June 2014.
- [61] S. Luo and Q. Hu. A dynamic motion pattern analysis approach to fall detection. In *IEEE International Workshop on Biomedical Circuits and Systems*,, pages 1–5, 2004.
- [62] I. S. MacKenzie. Evaluating eye tracking systems for computer input. In *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies*, pages 205–225. IGI Global, 2012.

- [63] P. Majaranta, I. MacKenzie, A. Aula, and K.-J. Rih. Effects of feedback and dwell time on eye typing speed and accuracy. *Universal Access in the Information Society*, 5(2):199–208, 2006.
- [64] J. Mak and J. Wolpaw. Clinical applications of brain-computer interfaces: Current state and future prospects. *IEEE Reviews in Biomedical Engineering*, 2:187–199, 2009.
- [65] M. J. Mathie, A. C. Coster, N. H. Lovell, and B. G. Celler. Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiological measurement*, 25(2):R1, 2004.
- [66] M. Mazo. An integral system for assisted mobility automated wheelchair. *IEEE Robotics Automation Magazine*, 8(1):46–56, Mar 2001.
- [67] P. McCool, G. Fraser, A. Chan, L. Petropoulakis, and J. Soraghan. Identification of contaminant type in surface electromyography (EMG) signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4):774–783, July 2014.
- [68] K. M. Means, D. E. Rodell, P. S. O’Sullivan, and L. A. Cranford. Rehabilitation of elderly fallers: Pilot study of a low to moderate intensity exercise program. *Archives of Physical Medicine and Rehabilitation*, 77:1030–1036, 1997.
- [69] Microsoft. <http://www.microsoft.com/en-us/kinectforwindows/>.
- [70] V. Mihajlovic, B. Grundlehner, R. Vullers, and J. Penders. Wearable, wireless EEG solutions in daily life applications: What are we missing? *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2014.
- [71] T. Morris and V. Chauhan. Facial feature tracking for cursor control. *Journal of Network and Computer Applications*, 29(1):62 – 80, 2006.
- [72] R. Motmans and E. Ceriez. Anthropometry table. In *Ergonomie RC, Leuven, Belgium*,, 2005.

- [73] I. Mougharbel, R. El-Hajj, H. Ghamlouch, and E. Monacelli. Comparative study on different adaptation approaches concerning a sip and puff controller for a powered wheelchair. In *Science and Information Conference (SAI), 2013*, pages 597–603, Oct 2013.
- [74] M. Mubashir, L. Shao, and L. Seed. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152, 2013.
- [75] J. Music, M. Cecic, and B. M. Testing inertial sensor performance as hands-free humancomputer interface. *WSEAS Trans. Comput.*, 8:715–724, Apr. 2009.
- [76] Y. Nam, B. Koo, A. Cichocki, and S. Choi. GOM-Face: GKP, EOG, and EMG-based multimodal interface with application to humanoid robot control. *IEEE Transactions on Biomedical Engineering*, 61(2):453–462, Feb 2014.
- [77] A. T. Nghiem, E. Auvinet, and J. Meunier. Head detection using kinect camera and its application to fall detection. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pages 164–169. IEEE, 2012.
- [78] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy. Fall detection-principles and methods. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1663–1666. IEEE, 2007.
- [79] N. Noury, P. Rumeau, A. Bourke, G. OLaighin, and J. Lundy. A proposal for the classification and evaluation of fall detectors. *Irbm*, 29(6):340–349, 2008.
- [80] W. H. O. WHO global report on falls prevention in older age. *World Health Organization (WHO) Library Cataloguing-in-Publication Data*, 2007.
- [81] W. H. Organization et al. Good health adds life to years: Global brief for world health day 2012. 2012.
- [82] J.-S. Park, G.-J. Jang, J.-H. Kim, and S.-H. Kim. Acoustic interference cancellation for a voice-driven interface in smart TVs. *IEEE Transactions on Consumer Electronics*, 59(1):244–249, February 2013.

- [83] G. Pfurtscheller, C. Neuper, G. Muller, B. Obermaier, G. Krausz, A. Schlogl, R. Scherer, B. Graimann, C. Keinrath, D. Skliris, M. Wortz, G. Supp, and C. Schrank. Graz-bci: state of the art and clinical applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):1–4, June 2003.
- [84] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3108–3113. IEEE, 2010.
- [85] R. Planinc and M. Kampel. Introducing the use of depth data for fall detection. *Personal Ubiquitous Computing*, 2012.
- [86] R. Raya, E. Rocon, R. Ceres, J. Harlaar, and J. Geytenbeek. Characterizing head motor disorders to create novel interfaces for people with cerebral palsy: Creating an alternative communication channel by head motion. In *2011 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–6, June 2011.
- [87] A. Robledo, C. S., and J. Guivant. Outdoor ride: Data fusion of a 3D Kinect camera installed in a bicycle. In *Proceedings of the 2011 Australasian Conference on Robotics and Automation*, 2011.
- [88] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier. Fall detection from depth map video sequences. In *Proceedings of the 9th international conference on Toward useful services for elderly and people with disabilities: smart homes and health telematics*, ICOST’11, pages 121–128, Berlin, Heidelberg, 2011. Springer-Verlag.
- [89] C. Rougier and J. Meunie. Fall detection using 3D head trajectory extracted from a single camera video sequence. In *First International Work-shop on Video Processing for Security(VP4S)*, 2006.
- [90] C. Rougier and J. Meunier. 3d head trajectory using a single camera. In *Image and Signal Processing*, pages 505–512. Springer, 2010.

- [91] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Monocular 3d head tracking to detect falls of elderly people. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pages 6384–6387. IEEE, 2006.
- [92] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Robust video surveillance for fall detection based on human shape deformation. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(5):611–622, 2011.
- [93] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. 3D head tracking for fall detection using a single calibrated camera. *Image Vision Comput.*, 31(3):246–254, Mar. 2013.
- [94] A. Savran, B. Sankur, and M. Taha Bilge. Comparative evaluation of 3D vs. 2D modality for automatic detection of facial action units. *Pattern Recogn.*, 45(2):767–782, Feb 2012.
- [95] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, June 2011.
- [96] A. Sixsmith and N. Johnson. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Comput*, 25, 2004.
- [97] SoftKinetic. <http://www.softkinetic.com/>.
- [98] S. Solanki and R. Kirsch. Evaluation of volitional control of hand with vertical force assist device for high tetraplegia. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 1339–1341, Aug 2012.
- [99] A. Sorvala, E. Alasaarela, H. Sorvoja, and R. Myllyla. A two-threshold fall detection algorithm for reducing false alarms. In *2012 6th International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–4, march 2012.

- [100] R. W. Soukoreff and I. S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' Law research in HCI. *Int. J. Hum.-Comput. Stud.*, 61(6):751–789, Dec. 2004.
- [101] E. E. Stone and M. Skubic. Fall detection in homes of older adults using the microsoft kinect. *Biomedical and Health Informatics, IEEE Journal of*, 19(1):290–301, 2015.
- [102] J. Struijk, E. Lontis, B. Bentsen, H. Christensen, H. Caltenco, and M. Lund. Fully integrated wireless inductive tongue computer interface for disabled people. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 547–550, Sept 2009.
- [103] L. Struijk. An inductive tongue computer interface for control of computers and assistive devices. *IEEE Transactions on Biomedical Engineering*, 53(12):2594–2597, Dec 2006.
- [104] X. Suau, J. Ruiz-Hidalgo, and J. R. Casas. Real-time head and hand tracking based on 2.5 d data. *Multimedia, IEEE Transactions on*, 14(3):575–585, 2012.
- [105] M. Suzuki, N. Ohyama, K. Yamada, and M. Kanamori. The relationship between fear of falling, activities of daily living and quality of life among elderly individuals. *Nursing & health sciences*, 4(4):155–161, 2002.
- [106] H. Tanie, K. Yamane, and Y. Nakamura. High marker density motion capture by retroreflective mesh suit. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005*, pages 2884–2889, April 2005.
- [107] American Spinal Injury Association. http://www.asia-spinalinjury.org/committees/prevention_facts.php.
- [108] Camera Mouse. <http://www.cameramouse.org/>.
- [109] L. R. Hochberg et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171, July 2006.

- [110] Spinal Cord Injury Facts and Figures at a Glance. <https://www.nscisc.uab.edu>. *National Spinal Cord Injury Statistical*, 2011.
- [111] N. Thome and S. Miguet. A hhmm-based approach for robust fall detection. In *9th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–8, 2006.
- [112] N. Thome, S. Miguet, and S. Ambellouis. A real-time, multiview fall detection system: A LHMM-based approach. *IEEE Trans. Circuits Syst. Video Technol. (TCSVT)*, 18(11):1522–1532, 2008.
- [113] M. E. Tinetti and C. S. Williams. Falls, injuries due to falls, and the risk of admission to a nursing home. *The New England Journal of Medicine*, 337(18):1279–1284, 1997.
- [114] C. Topal, S. Gunal, O. Kocdeviren, A. Dogan, and O. Gerek. A low-computational approach on gaze estimation with eye touch system. *IEEE Transactions on Cybernetics*, 44(2):228–239, Feb 2014.
- [115] M. Trotter, S. Stephenson, J. Simpson, and D. Casey. Fact sheet: Hospitalized falls. 2005.
- [116] J. Tu, H. Tao, and T. Huang. Face as mouse through visual face tracking. *Computer Vision and Image Understanding*, 108(12):35 – 40, 2007. Special Issue on Vision for Human-Computer Interaction.
- [117] H.-W. Tzeng, M.-Y. Chen, and J.-Y. Chen. Design of fall detection system with floor pressure and infrared image. In *International Conference on System Science and Engineering*, 2010.
- [118] C.-C. Wang, C.-Y. Chiang, P.-Y. Lin, Y.-C. Chou, I.-T. Kuo, C.-N. Huang, and C.-T. Chan. Development of a fall detecting system for the elderly residents. In *The 2nd International Conference on Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008*, pages 1359–1362. IEEE, 2008.

- [119] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph.*, 31(6):188:1–188:12, Nov. 2012.
- [120] M. Williams and R. Kirsch. Evaluation of head orientation and neck muscle EMG signals as command inputs to a human-computer interface for individuals with high tetraplegia. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(5):485–496, Oct 2008.
- [121] J. O. Wobbrock, J. Rubinstein, M. W. Sawyer, and A. T. Duchowski. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, pages 11–18, 2008.
- [122] J. R. Wolpaw and D. J. McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. In *Proceeding of the National Academy of Sciences of the United States of America*, pages 17849–17854, May 2004.
- [123] G. Wu and S. Xue. Portable preimpact fall detector with inertial sensors. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(2):178–183, 2008.
- [124] J. Yang, Q. Huang, Z. Peng, L. Zhang, Y. Shi, and X. Zhao. Capturing and analyzing of human motion for designing humanoid motion. In *2005 IEEE International Conference on Information Acquisition*, pages 6 pp.–, June 2005.
- [125] L. Yardley and H. Smith. A prospective study of the relationship between feared consequences of falling and avoidance of activity in community-living older people. *The Gerontologist*, 42(1):17–23, 2002.
- [126] B. Yousefi, X. Huo, E. Veledar, and M. Ghovanloo. Quantitative and comparative assessment of learning in a tongue-operated computer input device. *IEEE Transactions on Information Technology in Biomedicine*, 15(5):747–757, Sept 2011.

- [127] M. Yu, A. Rhuma, S. Naqvi, L. Wang, and J. Chambers. A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment. *IEEE Transactions on Information Technology in Biomedicine*, 16(6):1274–1286, 2012.
- [128] M. Yu, Y. Yu, A. Rhuma, S. Naqvi, L. Wang, and J. Chambers. An online one class support vector machine based person-specific fall detection system for monitoring an elderly individual in a room environment. *IEEE Journal of Biomedical and Health Informatics*, 2013.
- [129] X. Yu. Approaches and principles of fall detection for elderly and patient. In *10th International Conference on e-health Networking, Applications and Services, 2008. HealthCom 2008*, pages 42–47. IEEE, 2008.
- [130] X. Zhang and I. S. MacKenzie. Evaluating eye tracking with ISO 9241 part 9. In *Proceedings of HCI International*, pages 779–788. Springer, 2007.
- [131] Z. Zhang, H. S. Seah, C. K. Quah, and J. Sun. Gpu-accelerated real-time tracking of full-body motion with multi-layer search. *IEEE Transactions on Multimedia*, 15(1):106–119, Jan 2013.
- [132] Y. Zhu, B. Dariush, and K. Fujimura. Controlled human pose estimation from depth image streams. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW’08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.
- [133] X. Zhuang, J. Huang, G. Potamianos, and M. Hasegawa-Johnson. Acoustic fall detection using gaussian mixture models and gmm super-vectors. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 69–72, 2009.