Multimodal Human-Robot Interaction in an Assistive Technology Environment

---

by

Zhi Li

Submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

Intelligent Robotics Research Centre (IRRC)
Department of Electrical and Computer Systems Engineering
Monash University
Clayton, Victoria 3800, Australia

January, 2012

Contents

List of Figures…… vi
List of Tables…….. ix
List of Acronyms…… x
Abstract……... xii
Declaration …… xiv
Acknowledgment……... xv
1 Introduction 1
1.1 Motivation 1
1.2 Goals and Challenges 2
1.3 Contributions 5
1.3.1 Face Detection and Tracking 5
1.3.2 Hand Gesture Recognition 5
1.3.3 Consideration of Individual Differences 5
1.3.4 Multimodal Interaction and Dialogue Manager 6
1.3.5 Recognizing Emotional Gestures 6
1.4 Organisation of the Thesis 6
1.5 Publications 8
2 The Robot System 9
2.1 Review of 3D Imaging Technologies 10
2.1.1 Passive Sensing Methods 11
2.1.2 Active Sensing Methods 12
2.2 Cameras on the Robot 15
2.2.1 FaceLAB 15
2.2.2 PMD Range Camera 15
2.2.3 Kinect 16
2.3 Image Registration and Camera Calibration 18
2.3.1 Mapping between the Depth Image and the Colour Image 18
2.3.2 Calibration of the PMD Camera Set 20
2.3.3 Calibration of the Kinect 21

2.4 Data Processing 22
2.5 Conclusions 23
3 Face Tracking and User Identification 27
3.1 Related Work 27
3.1.1 Face Detection 27
3.1.2 User Identification 30
3.2 Face Detection and Tracking 33
3.2.1 Skin Colour Model 33
3.2.2 Face Detection and Tracking 34
3.2.2.1 Face Detection 34
3.2.2.2 Face Tracking 35
3.3 User Identification 36
3.4 Conclusions and Future Work 38
4 Hand Gesture Recognition 41
4.1 Related Work 41
4.1.1 Hand Tracking 41
4.1.2 Spatial-temporal Hand Motion 42
4.1.3 Static Hand Posture 43
4.2 Hand Tracking 44
4.2.1 Hand Detection by Depth Information 44
4.2.2 3D Particle Filter-based Hand Tracking 46
4.2.2.1 Repulsive Force 49
4.2.2.2 Resampling 50
4.2.2.3 Experiment and results 50
4.3 Dynamic Hand Motion Pattern Recognition 53
4.3.1 Experiments 54
4.4 Static Hand Posture Recognition 56
4.4.1 Hand-Forearm Segmentation 56
4.4.2 Orientation Rectification 59

4.4.3 Distance Transformation and Matching 60
4.4.4 Experiments 62
4.5 Conclusion and Discussion 64
5 Visual Interpretation of Natural Pointing Gestures 67
5.1 Related Work 68
5.2 Pointing Direction Calculation 71
5.2.1 Eye/Face-Hand Line 72
5.2.2 Forearm Direction 73
5.3 Pointing Gestures Occurrence 76
5.4 Object Selection 77
5.5 Experiments 80
5.6 Discussion and Conclusion 84
6 Multimodal Interaction: Using Gestures, Speech and Gaze 89
6.1 Related Work 90
6.2 Multimodal Interaction in a Probabilistic Framework 93
6.2.1 Speech Recognition and Word Mapping 95
6.2.2 Attention Direction Estimation 97
6.2.2.1 Related Work 97
6.2.2.2 FaceLAB 98
6.2.3 Alignment 102
6.2.4 Information Fusion at Decision Level 104
6.2.4.1 Task-orientated command table 104
6.2.4.2 Probabilistic method for multimodal information fusion 105
6.2.5 Dialogue Manager 108
6.3 Experiments 111
6.3.1 Experiments design 112
6.3.2 Results 113
6.3.2.1 Speech only 113
6.3.2.2 Interaction in a Natural Way 114

6.3.2.3 Preferred Deliberate Multimodal Interaction 115
6.4 Collaborative Work 116
6.4.1 HRI for Robot Navigation 117
6.4.2 HRI for Object Manipulation 118
6.5 Discussion and Conclusion 120
7 Recognizing Emotional Gestures 123
7.1 Related Work 124
7.2 Emotional Gesture Recognition 129
7.2.1 Feature Vectors 131
7.2.1.1 UCLIC Emotional Body Posture and Motion Database 131
7.2.1.2 Joints Positions and Hand Gestures 133
7.2.1.3 Dimension Reduction 135
7.2.2 Classification Approach 136
7.2.2.1 Support Vector Machines 136
7.2.2.2 Nonlinear Optimization 138
7.2.2.3 Cross-Validation 140
7.3 Experiments 140
7.3.1 Using UCLIC Database 140
7.3.2 Using Joints Data and Hand Gestures 141
7.4 Conclusion and Future Work 142
8 Conclusions and Future Work 145
8.1 Conclusions 146
8.2 Limitations and Future Extensions 148
Appendix A: PMD Camera 151
Specification of the PMD camera 151
Working Principle of the PMD Depth Camera 151
Appendix B: Kinect 154
Specification of the Kinect 154
Working Principle of the IR Depth Camera in the Kinect 154

References 157

List of Figures

Figure 2-1: Our assistive robot with the mobile platform, arm, gripper and cameras. 9
Figure 2-2: FaceLAB – a stereo-vision system for estimating a person's head pose and eye gaze direction 15
Figure 2-3: Combination of the PMD range camera and a RGB web camera.....16
Figure 2-4: Kinect 17
Figure 2-5: In an outdoor experiment, the PMD camera could still detect the pillar when it was exposed to the sunlight. The left picture from the web camera shows the scene and the right image plots the 3D data from the PMD range camera. However, Kinect hardly provided any valid data in this situation. 18
Figure 2-6: An example of mapping between the colour image from the RGB web camera and the depth image from the PMD range camera and the data processing result. 24
Figure 2-7: Mapping result of mapping the colour image and the depth image from the Kinect. It can be seen that, compared to the PMD camera, the Kinect has higher resolution, wider FOV and better quality of the depth data 25
Figure 3-1: An example of the skin colour detection 33
Figure 3-2: The left image shows the face detection result by the Viola-Lienhart algorithm (the blue circles indicate the face positions and sizes) which was implemented in OpenCV. It generated two false positives, which were eliminated by the proposed method as shown in the right image. 35
Figure 4-1: Examples of two hands tracking result with several skin-coloured objects in the background in a cluttered environment. The user rolled up his sleeves, making the tracking task more challenging. 52
Figure 4-2: An example of the hand waving gesture and the state transition of each gesture recognizer. A: Wave hand; B: Draw triangle; C: Come here; D: Pick up; E: Put down 55
Figure 4-3: In the left image, an example of the segmented region contains the hand and part of the forearm. In the right image, an example of the segmented region contains the palm and fingers only 57
Figure 4-4: Two examples of the hand-forearm segmentation 59
Figure 4-5: Hand gestures with different orientations may have different meanings: (a.) thumb up (means 'good'), (b.) thumb down (means 'bad') and (c.) go to right 59
Figure 4-6: An example of the hand shape orientation rectification.....60
Figure 4-7: An example of Distance Transformation of a hand silhouette image 61
Figure 4-8 Hand shape templates 63
Figure 5-1: The person is pointing at a spot on the floor using the Eye/Face-Hand line ...71

Figure 5-2: The person is pointing at a teapot on the table using the forearm direction ... 72
Figure 5-3: An example of the calculated first principal component of a set of 3D points . 74
Figure 5-4: An example that shows the PCA result (the blue line) is not a good estimation of the main trend of the 3D points under the influence of outliers. Iteratively employing PCA under the RANSAC framework, it overcomes the influence of outliers and provides the correct estimation (the red line) 75
Figure 5-5: An example of a pointing gesture in an office environment 79
Figure 5-6: A snapshot of the experiment setup. The potential targets are marked by red circles. They are placed at various locations, at different height and distance levels. Some of the objects are not shown in this picture due to the limited view of the camera. 79
Figure 5-7: Objects locations in the experiments with the PMD camera set 81
Figure 5-8: Objects locations in the experiments with the Kinect sensor 81
Figure 5-9: An example of pointing with a rotated wrist. 86
Figure 5-10: Object arrangement and the viewpoint make a difference to the object selection ability. Both pointing vectors start from point P. 86
Figure 6-1: Flow chart of the multimodal interaction architecture 94
Figure 6-2: Examples of word mapping. 96
Figure 6-4: The corresponding 3D head model of the person in Figure 6-3. 99
Figure 6-6: When two pointing gestures are detected within the time window of the spoken word “that”, we do not simply choose the closest one. Instead, the valid one is chosen using Equation (6.3). 103
Figure 6-7: The flow chart of the dialogue manager. It shows an example of requesting an object, so only action and object classes are included here. An instance with the unique maximum probability related score indicates the valid information for the class is recognized. If the accumulated information suffices any item in the command table, the specific execution is called. The robot may need to confirm its interpretation before the execution 110
Figure 6-8: A snapshot from the panoramic camera. It demonstrates the experimental environment. The room is virtually divided into cells. The robot needed to find its path to the destination and avoid the obstacles. The image is from [92]. 117
Figure 6-9: Human robot interaction and object manipulation. This experiment demonstrated that the robot correctly interpret the user’s command requesting a specified object. The robot could resolve the ambiguity which arose from the presence of two identical objects. 120
Figure 7-1: Overview of the emotional gesture recognition system. 130

Figure 7-2: Examples of the expressive avatars which are built using the captured 3D data. Pictures are obtained from the UCLIC emotional body posture and motion database [19].....132
Figure 7-3: Examples of emotional gestures captured by the Kinect. Note that the ‘victory’ hand gesture and closed fist are expressive in the ‘happy’ and ‘angry’ cases. 135
Figure 7-4: Bottom-up multi-class SVMs structure for the pairwise scheme. With 7 classes, 6 comparisons need to be performed to obtain a classification result. 138
Figure A-1: Phase shift distance measurement principle. Picture from [132]. 152

List of Tables

Table 4-1: Hand motion recognition results 56
Table 4-2: The confusion matrix of the hand shape recognition results 64
Table 5-1: Overall estimation result of different pointing methods with the PMD Camera set 83
Table 5-2: Overall estimation result of different pointing methods with the Kinect 83
Table 5-3: The ratio of the preferences associated with object locations (using PMD) 83
Table 5-4: The ratio of the preferences associated with object locations (using Kinect) 83
Table 6-1: Examples in the command table 105
Table 7-1: The feature vector components and their descriptions 132
Table 7-2: The confusion matrix of the emotional gesture classification results using UCLIC database 141
Table 7-3: Confusion matrix of the emotional gesture classification results using the motion data captured by the Kinect and the hand gesture recognition system 142
Table A-1: The PMD Camera Specifications [93] 151
Table B-1: The Kinect Specifications [193, 235] 154

List of Acronyms

ASL	American Sign Language
ASR	Automatic Speech Recognition
DOF	Degree of Freedom
DT	Distance Transformation
EM	Expectation-Maximization
FABO	Face and Body Gesture Database
FOV	Field of View
FSM	Finite State Machine
GMM	Gaussian Mixture Model
HMMs	Hidden Markov Models
HRI	Human Robot Interaction
HSV	Hue, Saturation, and Value
IR	Infrared
LDA	Linear Discriminant Analysis
LIDAR	Light Detection and Ranging
MDA	Mixed Discriminant Analysis
MLP	Multi Layer Perceptron
NCC	Normalized Cross-Correlation
PDBNN	Probabilistic Decision-based Neural Network
PDF	Probability Distribution Function
PRS	Probability Related Score
RANSAC	Random Sample Consensus
RBF	Radial Basis Function
RGB	Red Green Blue

ROI	Region of Interest
SAD	Sum of Squared Differences
SDK	Software Development Kit
SFS	Shape from Silhouette
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SNR	Signal-to-Noise Ratio
SSD	Sum of Squared Differences
SVD	Singular Value Decomposition
SVMs	Support Vector Machines
TOF	Time of Flight
UCLIC	University College London Interaction Centre

Abstract

The research work presented in this thesis is motivated by the increasing demand for care for the elderly. A domestic assistive robot has the potential to supplement humans in the provision of assistance for the elderly with simple daily tasks, such as retrieving small objects from various places, switching lights on and off, and opening and closing doors. The proposed assistive robot possesses both transactional intelligence and spatial intelligence. This thesis concentrates on the realization of the transactional intelligence, which enables the robot to naturally and effectively interact with human users. The ultimate goal of this research is to develop a system for the robot to perceive multiple modalities used by humans during face-to-face communication, including speech, eye gaze and gestures, so that the robot is able to understand the user's intention and make appropriate responses.

Some important features in the design and implementation of the system are as follows.

1. Naturalness and effectiveness are the fundamental principles in the design of the interaction interface. Therefore, only cameras are used as non-contact sensing devices.
2. The user is observed only from the robot's view, so that the interaction can take place anywhere rather than be confined to a particular room.
3. The behavioural differences between individuals are emphasized, enabling the robot to give appropriate responses to different users. This is achieved by a user identification method and a profile built for each individual user, which stores several characteristics of a specific user.
4. The proposed hand gesture recognition system recognizes both dynamic motion patterns and static hand postures. The 3D Particle Filter-based hand tracking approach combines information of colour, motion and depth. It robustly tracks the hands even when the person wears a short-sleeved shirt exposing the forearm.
5. Different sources of information conveyed by speech, eye gaze and gestures are aligned and then combined by the proposed multimodal interaction system. The approach takes into account that each sub-system may generate incomplete or erroneous results.

6. Mutual interaction is realised by a dialogue manager. Based on the perceived information, the robot decides either to perform a required task or negotiate with the user when the command is ambiguous or not feasible.
7. The ability of the robot to infer the user's emotional states as a social companion is also attempted.

The technical contributions of this thesis have been validated with a series of experiments in typical indoor environments.

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

_____

Zhi Li

Acknowledgment

First of all, I would like to thank my main supervisor Prof. Ray Jarvis, for all of his brilliant ideas, encouragement, advices, and patiently guided me throughout my PhD candidature. I would also like to give my thanks to my associate supervisor, A/Prof. Andy Russell for his advice and reviewing this thesis. I also appreciate the valuable suggestions from A. Prof. Lindsay Kleeman and Dr. Wai Ho Li during my seminar presentations.

Secondly, thanks must go to my colleagues and friends in the Intelligent Robotics Research Centre (IRRC). Fredy Tungadi, Dennis Lui, Anh Tuan Phan, Sutono Efendi, Om Gupta, Wai Ho Li, Alan Zhang, Ee hui Lim, Nghia Ho, Gustavo Schileyer, Damien Browne, Jay Chakravarty, and the rest. Special thanks to Dennis Lui who gave me many advices in discussions. Great thanks to Sutono Efendi and Om Gupta, who did collaborative work with me.

I also thank to Monash University for giving me a scholarship to support me to pursue a PhD degree. The financial support of the ARC grant for the project of Multi-sensory Fusion and Understanding in Robotic Assistive Technology Environments is also gratefully acknowledged.

Acknowledgement is also given to Nadia Bianchi-Berthouze and Andrea Kleinsmith and AffectME: Affective Multimodal Engagement. A project funded by a Marie Curie International Re-Integration Grant: (MIRG-CT-2006-046434). I would also like to thank Dr. Alex McKnight for proofreading this thesis.

Finally, I want to thank my parents, Qinghui Yao and Aizhu Li, for their unconditional support and understanding. Last but not least, I would like to thank my wife, Minyi Li, for her love and care when working on this thesis.

1

Introduction

1.1 Motivation

The populations of many countries, particularly in the western world, are aging rapidly. For example, as reported in [9], the proportion of the population aged 65 years and over in Australia was 12% in 1999, but is projected to increase to 24%-27% in 2051. This causes increasing demands for care of the elderly by health care agencies, family members and home nurses. This has motivated researchers to develop domestic assistive robots to supplement humans in providing assistance to the elderly or disabled with simple daily tasks, such as retrieving small objects (e.g. cups, fruit and books) from various places, switch on/off the lights, close/open the door and so on. Assistive robots, as an alternative to human carers, have great potentials to improve the quality of life for the elderly.

Some examples of such assistive robots include Nursebot [197] which is designed to be an assistant for nurses and elderly people in the hospital environment, PAM-AID [154] which provides mobility aid for the elderly blind, an Intelligent Robotic Assistive Living System [116], the Karlsruhe humanoid robot [269], rehabilitation robots [122], a task-driven robotic system [207], PeopleBot [4], PR2 [288] and so on.

In fact, friendly assistive robots are widely welcomed not only by elderly people. These robots are also expected to act as humans' social companions. They will effectively and reliably work in a human-robot team, take care of patients, play an important role in the course of interactive education for children and be involved in many other tasks.

1.2 Goals and Challenges

Over the past decades, industrial robots have demonstrated great success in factories in increasing productivity by performing repetitive jobs. In general, industrial robots work in constructed and controlled settings and repetitively follow the same pre-programmed routines with little intelligence. The advantage of industrial robots is that they are able to perform specific tasks with great speed and accuracy. However, they are not able to work in an uncontrolled environment and make their own decisions according to the varying contexts.

For a robot to be truly intelligent, it requires the ability to perceive the scene independently. The quality of the robot's capability to perceive, reason and act determines the scope of applicability of a robotics system [118].

For a domestic assistive robot in particular, two types of intelligence need to be provided: Transactional Intelligence and Spatial Intelligence [116]. Spatial Intelligence is the ability of the robot to navigate itself through spaces efficiently and retrieve requested objects accurately. The robot needs to approach the destination avoiding any obstacle in its path, and a hand/eye coordination system is needed to segment the object from the background and grasp it successfully. Transactional Intelligence is the capability of the robot to interact with people effectively. To achieve natural interaction, robots should be able to perceive all the communication modalities used by humans during face-to-face interaction, such as pen/keyboard input, speech, gesture, eye gaze and so on. The robot should be able to understand the user's intention, and accordingly make decisions, either to perform a required task or to negotiate with the user when the provided information is ambiguous or not feasible due to its physical ability or environmental constraints.

The prototype of our Intelligent Robotic Assistive Living System has been described in [116]. It is an ambitious project which involves the work of four researchers. The robot is designed to implement some simple assistive tasks in the kitchen, office and living room scenarios. For example, a user may ask the robot to go to the kitchen and get a can of soft drink from the refrigerator, pick up a coffee cup which is specified by the user's pointing gesture, turn on/off the TV, pass today's newspaper, send a parcel to someone and so on. Additionally, as a social companion, the ability of a robot to initiate a conversation according to the user's emotional state is also valuable. Spatial intelligence has been

researched and implemented by colleagues, including navigation and path planning in time varying environments [91, 92] and the eye/hand coordination system [60, 62].

This thesis concentrates on transactional intelligence and proposes a multimodal interaction system for the assistive robot to recognize the user's gestures, speech, attention direction and emotional states, and integrates the information from these multiple input channels to understand the user's intention, then determine an appropriate response or negotiate with the user when the command is ambiguous or not feasible.

Each modality in the multimodal interaction system plays an important role; however, this thesis mainly focuses on the recognition of the user's various gestures including hand gestures, pointing, attention direction and emotional gestures, while the speech recognition system is relatively simple.

The importance of gestures has been repeatedly emphasised by numerous researchers. According to Mehrabian [186], during interpersonal interaction, spoken sentences account for only 7% of what a listener comprehends; non-verbal communicative behaviour (i.e. body language and intonation) contributes the remaining 93%. According to Kendon [142] the use of non-verbal communication can be, in theory, just as or more effective than speech. Vinayagamoorthy et al. claimed in [282] that 93% of communication is non-verbal: 55% is expressed through body language and 38% through the tone of voice.

Therefore, gesture recognition has attracted a great deal of attention in the field of Human Robot Interaction (HRI) research and application. The main advantages of utilizing gestural expressions in HRI are as follows:

1. Gestures are easy and natural ways to express geometrical and spatial information to a robot. For example, a person can simply point to the object which he wants the robot to grasp, rather than using complicated descriptions by speech utterances. Gesture has many benefits when there are multiple identical/similar objects in the scene. The robot does not even need to recognize what the object is before picking it up when an accurate pointing direction is given to locate the object.
2. Gestures increase the chance of the robot understanding the user's intention. Information expressed through gestures is especially useful when the speech recognition system is not available or fails, for example, in a noisy workshop or at a distance. If the user shows a stop sign by a hand gesture at the same time as he/she

says the word “stop”, it would be easier for the robot to understand the user’s intention, since it needs to correctly recognize only one of the two expressions. This is particularly important when the user wants the robot to stop in an emergence.

3. Gesture and speech are complementary means of expressing intentions and people prefer to interact multimodally rather than unimodally. A multimodal interaction system can provide the flexibility that allows users to select from or mix different input modes.
4. Gestures are more natural and are highly preferred by humans to convey certain kinds of information. For example, people express their emotional states more often by facial expressions or body gestures than in spoken sentences.

To achieve natural and effective interaction between human users and robots, there are a number of challenges. For the implementation of transactional intelligence in our domestic assistive robot, five main challenges are listed as follows:

1. Naturalness of the interaction is of utmost importance. Since the intended user is not expected to be an expert in robotics, convenience and ease of use are the most fundamental and critical principles in the design of the interaction interface. Hence, data suits/gloves and adhesive markers are not employed. Only cameras are used as non-contact sensing devices.
2. The robot works in a cluttered and dynamic environment. The mobile robot is required to interact with humans and implement tasks in various places such as an office, living room and kitchen. Therefore, it must be able to accommodate time-varying environments.
3. Individuals have different ways of behaving. The same gesture may convey completely different meanings, according to the cultural background and personal experience of the user. The robot needs to take individual differences into account in order to give appropriate responses.
4. The result from each recognition module may be imperfect or erroneous. The robot should be able to combine the information from multiple modalities and tolerate errors to some extent.
5. In order to achieve natural interaction, the system needs to run in real time.

1.3 Contributions

The key contributions of the research work described in this thesis are summarised in the following sections.

1.3.1 Face Detection and Tracking

In a human-centric interactive system, tracking the user's face in real time is always important and often the very first step which allows the subsequent interaction. The method proposed in Chapter 3 [166] extends the well-known face detection algorithm [168, 283] by the combination of colour and depth information. After initialization of the position and size of the face, the CAMSHIFT algorithm is then adopted for fast tracking. Several constraints prevent the face tracking from being lost.

1.3.2 Hand Gesture Recognition

Direct use of the hands as a communicative or manipulative input channel is attractive in HRI because of its simplicity and naturalness. The proposed 3D Particle Filter-based hand tracking approach makes use of three cues including colour, motion and depth to track the hands robustly. Two hands were successfully tracked simultaneously in the experiments, in the presence of some skin-coloured distracting objects in the background, even when the person wore short-sleeved clothes exposing the forearm. After locating the hands' position, the proposed hand gesture system is able to recognize both dynamic motion patterns and static hand postures [165]. The former is modelled by the Finite State Machine (FSM) method; to recognize the latter, three steps including hand segmentation, orientation rectification and matching are implemented in sequence.

1.3.3 Consideration of Individual Differences

The robot is designed to take into account that each individual may behave differently. This enables the robot to respond more appropriately during the interaction. This feature is achieved by a user identification method and a user profile for each registered user. The profile includes several characteristics of that user, such as the body height, a set of self-defined gestures, the pointing method preference, a trained speech profile, a lookup table for mapping between certain pronunciations in specific contexts, temporal relationship between pointing gestures and speech, request frequency of certain

objects, and so on. In particular, the choice of pointing methods has been extensively researched considering the individual preference associated with the location of the target [166]. A generic profile is built for any unrecognized user.

1.3.4 Multimodal Interaction and Dialogue Manager

For a humanoid robot to participate in our daily life and interact with humans in a natural style, it is essential that the robot is able to understand humans' multiple communication modalities. A multimodal interaction interface provides the flexibility that allows users to select from or mix different input modes. The proposed multimodal interaction system [167] combines the information from several sources, including speech, eye gaze and gestures, by a probabilistic method at the decision level. A dialogue manager uses a task-orientated command table to determine whether the fusion result is complete and ready for execution; otherwise, the robot negotiates with the user when the command is ambiguous.

1.3.5 Recognizing Emotional Gestures

Recognition of a person's emotional states through body gestures has been achieved and is presented in Chapter 7. The proposed approach adopts the Support Vector Machines (SVMs) classification method, and optimizes its parameters by the Nelder-Mead nonlinear searching method. The Cross-Validation technique is employed to make the most of the available data and avoid the overfit problem. The approach was successfully tested on a publicly available database and our captured data. The experimental result also showed that hand gestures could provide distinguishing features, as a complement to body joints.

1.4 Organisation of the Thesis

The thesis is organised into eight chapters followed by two appendixes. Chapter 2 describes the 3D imaging devices that have been used on our robot to realise this project. Chapter 3 to Chapter 7 detail the proposed approaches/systems and report the experimental results. Each of these chapters begins with a review of the related work in the topic, followed by a description of the approach as well as the experiments, and ends with a conclusion and possible future work.

Chapter 3, Face Tracking and User Identification, deals with two problems. The first is the detection and tracking of the position of the user's face, which is important in a human-centric robotic system. The second is to identify the user and build/load a user profile for each individual, so that the robot is able to give appropriate responses for different individuals.

Chapter 4, Hand Gesture Recognition, proposes a method for hand gesture recognition. Two approaches with different complexities are presented for hand tracking. The trajectory of the hands is used to model the dynamic motion patterns, and the hands' images are segmented, rectified, resized and then matched with templates to classify the meanings of the static hand postures.

Chapter 5, Visual Interpretation of Natural Pointing Gestures, describes the methods of detecting the occurrence of a pointing gesture and calculating the pointing direction. Two kinds of pointing methods, i.e. the Eye/Face-Hand line and the forearm direction, are considered, because the choice of a pointing method may vary with each individual's preference and the location of the intended target.

Chapter 6: Multimodal Interaction, Using Speech, Gaze and Gestures, first describes the approaches to recognizing a person's speech utterances and estimating his/her attention direction. Then a multimodal interaction system is proposed, which integrates speech, gaze direction and gestures. In addition, the collaborative work towards our final goal of a mobile assistive robot is presented here, combining the transactional intelligence in this thesis with the spatial intelligence systems developed by colleagues.

Chapter 7, Recognizing Emotional States through Body Gestures, presents the preliminary work on recognizing emotional gestures. Body joints and hand gestures are used to construct the feature vectors. The dimension of the feature vectors is first reduced and then fed to a SVMs-based classification method. The experimental results show that hand gestures could add a favourable contribution to recognition.

Chapter 8, Conclusions and Future Work, summarises the thesis and proposes suggestions for future work.

Appendixes: The two appendixes include the specification and working principles of the PMD range camera and the Kinect, which are the depth-sensing devices on our robot.

1.5 Publications

Six papers based on the research work presented in this thesis had been accepted or published to date. In order of publication date, the papers are as follows:

1. R. Jarvis, O. Gupta, S. Effendi and Z. Li, "An intelligent robotic assistive living system", in the second International Conference on Pervasive Technologies Related to Assistive Environments (PETRA), Corfu, Greece, 2009, pp 1-8.
2. Z. Li and R. Jarvis, "A Multi-modal Gesture Recognition System in a Human-Robot Interaction Scenario", in the international workshop on Robotic and Sensors Environments (ROSE), Lecco, Italy, 2009, pp 41-46.
3. Z. Li and R. Jarvis, "Real time Hand Gesture Recognition using a Range Camera", in the Australasian Conference on Robotics and Automation (ACRA), Sydney, Australia, 2009 .
4. Z. Li and R. Jarvis, "Visual Interpretation of Natural Pointing Gestures in 3D Space for Human-Robot Interaction, in the 11th International Conference on Control, Robotics and Vision (ICARCV), Singapore, 2010, pp 2513-2518.
5. I. Zukerman, A. Mani, Z. Li and R. Jarvis, "Speaking, Pointing and Unknown Things -- from Simulations to the Laboratory", in the 7th Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Barcelona, Spain, 2011.
6. Z. Li and R. Jarvis, "Multimodal Interaction System for a Household Assistive Robot", in the Australasian Conference on Robotics and Automation (ACRA), Melbourne, Australia, 2011.

2

The Robot System

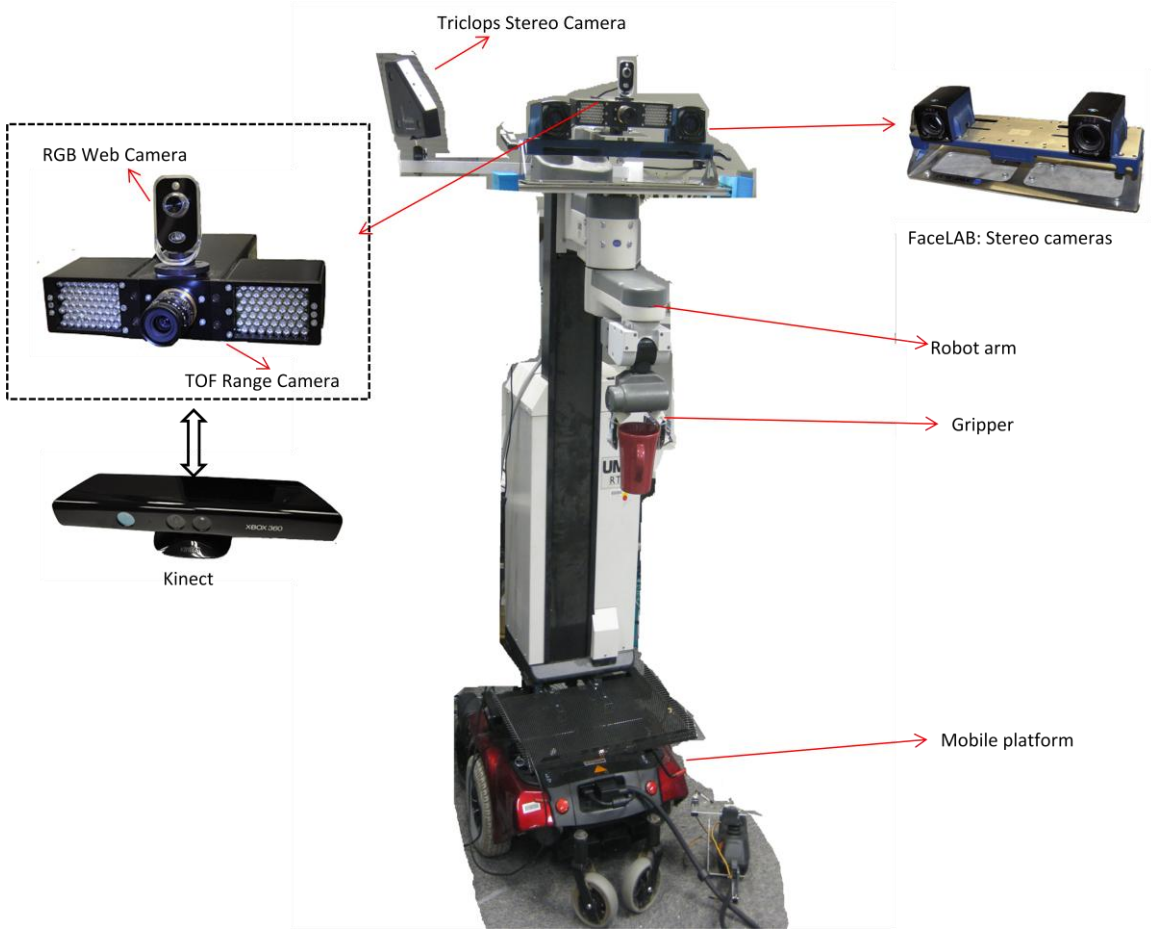


Figure 2-1: Our assistive robot with the mobile platform, arm, gripper and cameras.

The configuration of our domestic assistive robot is shown in Figure 2-1. The robot has a wheelchair base with six wheels as its mobile platform: two central drive wheels and two pairs of castor wheels (one at the front and the other at the back). A joystick controller using two servomotors can interface to an on-board PC through a RS-232 serial port [92].

The robot is equipped with a robot arm manipulator (UMI RTX 100) [150], which has six Degrees of Freedom (DOF) and a two-finger gripper for grasping small objects. A Triclops black and white stereo camera, or alternatively a Bumblebee (colour) camera [234], is mounted on the robot's upper body for object recognition and segmentation tasks in a hand/eye coordination system. These devices are employed for the implementation of spatial intelligence systems, i.e. robot navigation and object manipulation, by the other three colleagues in this assistive robot project.

To achieve the transactional intelligence system of the robot, several cameras and a wireless microphone are employed. A Time-of-Flight (TOF) range camera derives depth data and an ordinary Red-Green-Blue (RGB) web camera compensates for its low-resolution and also provides colour information. These two cameras comprise a camera set, and can be used interchangeably by the Kinect which also consists of a depth sensor and a colour camera. A pair of stereo cameras, called faceLAB, is used to estimate the user's head pose and eye gaze direction.

3D information is important in our system in order to achieve spatial accuracy and robust performance. In this chapter, we first review some popular 3D sensing techniques, and then introduce the features of the above-mentioned cameras on our robot. In Section 2.3, the camera calibration and image registration methods are described. Finally, the data processing method is described. Part of the work in this chapter was published in [164].

2.1 Review of 3D Imaging Technologies

Conventional 2D video cameras have been used as main imaging devices for many decades. However, depth information is lost during the process of perspective projection from the 3D real world to the 2D image planes.

Depth data is critical to being able to retrieve 3D information from image pixels. 3D information is desirable for the computer/robot to perceive the real world. Moreover, for many computer vision tasks, such as image segmentation, object recognition, motion analysis and scene reconstruction, 2D images are often insufficient to provide reliable and robust results. The introduction of depth cameras has significantly simplified these tasks and improved their performance [210, 251, 304].

Depth information can be obtained using either passive or active methods. There are various techniques that are able to produce depth data. In this section, we include only a brief review of the methods which have already been used or have great potential to be applied in the areas of human motion capture, gesture recognition and human-robot/computer interaction. More complete and detailed reviews can be found in [117], [45] and [22].

2.1.1 Passive Sensing Methods

Passive sensing methods use reflected ambient light rather than actively emitting any kind of light, and their performance partially relies on the lighting conditions in the environment. Stable lighting with moderate brightness is usually preferred. Their low cost is one of the important reasons for their popularity, since they mainly utilize digital cameras.

The Shape from Silhouette (SFS) 3D reconstruction technique uses a sequence of images around an object against a well constructed background to reconstruct the 3D structure of the object. To obtain the 3D shape of the target, silhouettes of the object in the foreground are extracted to form a visual hull [161], which is a bounding geometry of the actual 3D shape of the object. This method requires several images of an object from different viewpoints and the object needs to remain static during these captures. For real time applications, multiple calibrated and synchronized cameras are mounted surrounding the object to capture images simultaneously [292]. Two important factors which affect the result of the generated visual hull are the number of the image sequences and the positioning of the cameras. As the number of the images used increase, the computed visual hull becomes finer. Shanmukh and Pujari [255] proposed a method to determine the positioning of the cameras that optimize the reconstruction. However, using the SFS technique, the shape of non-convex objects cannot be fully computed.

Stereo-vision systems employ two or more calibrated cameras with known baselines. With the correspondence pixels in image pairs from the calibrated stereo cameras, it is possible to determine the distance for the given pixels using the stereo triangulation method [98]. Images are normally rectified using the calibration results so that the epipolar lines are aligned before triangulation. In this way, the matching procedure becomes a linear search. The accurate calibration of the intrinsic and extrinsic parameters of each camera is

essential for image rectification and 3D-coordinate triangulation. Stereo-vision systems have been widely used in the computer vision and robotics fields [97, 100, 125]. In addition to their low cost, they have the advantage that they can be deployed in both indoor and outdoor environments.

However, there are several non-trivial problems in stereo-vision systems. Finding two pixels, which represent the same 3D physical point, in the left and right image pair is called the *correspondence* problem. A number of methods, local or global, have been proposed. Local methods, such as Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD), Normalized Cross-Correlation (NCC) [36], can be efficient, but they are sensitive to occlusion and often fail in texture insufficient regions; while global methods, such as Dynamic Programming [14], Graph Cuts [30] and belief propagation [273], can be less sensitive to local ambiguity but they are more computationally expensive. All of the methods require the object surface to have sufficient texture, otherwise, the correspondence cannot be found correctly. Therefore, stereo triangulation methods suffer from inaccuracy caused by mismatching in the texture insufficient regions of the objects. In addition, the arrangement of the stereo cameras has to make sure that the common Field of View (FOV) is large enough to cover the desired scene. 3D reconstruction of sparse features can be fast, while building a dense depth map at every pixel is time-consuming.

2.1.2 Active Sensing Methods

Active 3D sensing devices are also widely employed in research institutes and industry. In contrast to passive methods, these methods normally comprise two components: transmitting and receiving. Different from conventional cameras, they are less sensitive to the ambient illumination conditions. They do not use the texture and colour information of the target, so they can achieve texture- and colour-invariant results, which is desirable for many computer vision applications. Two of the most popular active techniques, i.e. Time-of-Flight and structured light, are briefly described in the following sections.

Time-Of-Flight (TOF): A TOF rangefinder measures the distance to a point on a surface by timing the round-trip time or determining the phase shift between the emitted and received light pulses. Laser, ultrasound and infrared light are the common types of emission.

Because the rangefinder only detects one point at a time, to obtain 3D information of the whole scene, the scanner needs to scan point-by-point by changing the direction of the emitted light. For example, Light Detection and Ranging (LIDAR) systems often use moveable mirrors to steer the laser beam. Alternatively, a TOF camera, with a 2D sensor array, can measure the entire scene at one time.

Compared to stereo triangulation systems which need to search for correspondences in each image pair, TOF systems deliver distance data in a simple and direct way, and require little processing power. A comparison of the PMD camera (which is a type of TOF range camera that will be described in Section 2.2) and stereo-vision for the task of surface reconstruction has been investigated by Beder et al. [12]. Their experiments showed that the PMD range camera system outperformed the stereo system by about one order of magnitude in terms of achievable accuracy for distance measurements.

However, a disadvantage of the TOF camera is that the light signal may suffer from interference and multiple reflection problems. More importantly, because TOF cameras normally employ low strength infrared for safety reasons, the signal may be contaminated by sunlight, which has a broad spectrum. Therefore, these devices are normally not ideal for outdoor environments. Popular TOF cameras on the market include the PMD CamCube3.0 [231], the SwissRanger 4000 [189], the Panasonic D-Imager [222], and the Fotonic C70 [73] and so on.

TOF cameras have been applied in a wide range of applications. Wang et al. [285] extracted 3D landmarks from the TOF range data for simultaneous localization and mapping (SLAM). Soutschek et al. [265] proposed a gesture classification system for scene navigation in medical imaging applications using TOF cameras in order to avoid physical interaction with an input device. Haker et al. [95] combined depth and greyscale images of a TOF camera to recognize simple hand gestures for the application of controlling a slideshow presentation. In [59], a TOF camera was used on a mobile robot to perceive obstacles that were not in the scan plane of a laser range finder. The camera was mounted on a pan-tilt platform to overcome the limited FOV of the sensor. TOF cameras have also been used for real time 3D respiratory motion detection in [226] and automatic plant phenotyping in [149].

Structured Light: A 3D imaging device using the structured light technique consists of two components, a projector that emits light patterns and one or multiple observing camera(s).

Similar to stereo-vision, the corresponding points in the projected patterns and the camera images need to be found, and the distance is then determined by the triangulation method. Therefore, an important issue for such systems concerns the choice of the patterns and the associated coding and decoding strategies.

The structured light technique can be classified, in terms of the method of encoding the light patterns, into three categories: time multiplexing, direct coding and spatial neighbourhood [247]. Each has its own advantages and drawbacks. The first method is easy to implement and is able to achieve high resolution, but it is not suitable for dynamic scenes. The second is sensitive to noise and light variations, and the third method can deal with moving objects, but with lower resolution than the other two [5].

Although many variants of structured light patterns have been proposed, parallel stripes have been most widely used. An exact geometric reconstruction of the surface can be calculated according to the displacement of each stripe. For this purpose, the individual stripe has to be identified. Methods such as tracing or counting stripes can be used. Alternatively, the time multiplexing method projects a sequence of alternating stripe patterns, which results in a Gray code for each pixel revealing the stripe number.

The resolution of the structured light method relies on several factors: the spatial density of the stripes, the optical quality and the wavelength of the emitted light. The resolution can be at micrometer level [21]. However, like other optical methods, extremely reflective or transparent surfaces, multiple reflections and cavities often cause difficulties.

The structured light technique has been successfully used in various applications, including precise shape measurement for production control [21], and face and body shape reconstruction [70].

As each range sensing method has its own advantages and drawbacks, the combination of different methods could achieve more accurate and robust depth maps. For example, active TOF sensors and passive stereo cameras have complementary error characteristics. Rich texture may cause difficulties for the TOF sensors due to the variety of the surface reflectance, while stereo cameras excel on such regions. In contrast, stereo systems perform poorly on textureless surfaces. Zhu et al. [303] explored these complementary characteristics by first generating per-pixel Look-Up-Tables (LUTs) to compensate the depth bias caused by the different reflectance and system noise of the TOF

sensor, and then further improved the accuracy by a fusion method for combining the results from both sensors. The fusion was built on a pixel-wise reliability weighting calculated for each method. However, the proposed method is only useful for the accurate reconstruction of static scenes, since it requires about 20 seconds for the generation of a 400×300 depth map, which is far from real time.

2.2 Cameras on the Robot

2.2.1 FaceLAB

A commercial stereo-vision system called "faceLAB" (version 1.1, developed by the Seeing Machines Company [253], as shown in Figure 2-2) is employed on our robot to estimate the user's head pose and eye gaze direction. Images from the cameras are analysed to work out the characteristics of a person's face. Then the head position, orientation and gaze direction are calculated from these features (details will be provided in Chapter 6). Sparse features are tracked in small searching windows so that the system runs at a high frame rate up to 60 fps.

Figure 2-2: FaceLAB – a stereo-vision system for estimating a person's head pose and eye gaze direction

2.2.2 PMD Range Camera

A TOF range camera (PMD[version]®19k) developed by PMD Technologies GmbH [232] has been employed since the beginning of the author's Ph.D. candidature in 2008. It uses the TOF technology to measure the distance of a 3D point by the phase shift between the received and transmitted infrared light pulses. The 2D-array sensor inside the camera consists of Photonic Mixer Devices (PMDs) which enable the sensing and demodulation of incoherent infrared light signals in one component [242]. Its detailed specifications and working principle are provided in Appendix A.

The PMD camera on our robot provides both depth data and greyscale images. However, its resolution (160×120) is low and the quality of the greyscale image is poor for most image processing purposes. Therefore, we combine the PMD camera with an ordinary web camera, which also compensates for the lack of colour information. A picture of the configuration of the two cameras is shown in Figure 2-3. For the purpose of brevity, in the rest of this thesis, the set of the two cameras is simply referred to as 'the PMD camera set'.

Figure 2-3: Combination of the PMD range camera and a RGB web camera

3D imaging technologies have developed very quickly. The latest PMD product, CamCube 3.0 (released in June 2010), is now available with significant improvements in several aspects. High accuracy even with lower integration time and longer distance is achieved and the frame rate is increased up to 40 fps with the resolution of 200×200 pixels (80 fps with @ 160×120 pixels). Importantly, it is claimed that due to the integrated Suppression of Background Illumination technology, it can be deployed in both indoor and outdoor environments. In addition, the resistance to motion blur allows the detection of fast moving objects [231, 233].

2.2.3 Kinect

More Recently (November 2010), the launch of Kinect by Microsoft has made depth imaging technology available at a consumer price. The Kinect is a depth sensing device originally designed for the Xbox 360 game console. It determines the depth information of the scene using the structured light technology. A set of coded light patterns is emitted by a projector and then observed and decoded by an Infrared (IR) camera [235]. Detailed descriptions of the specification and working principle are provided in Appendix B.

The attractiveness of providing high quality depth data at a real time frame rate and at a low price makes it popular among amateurs and researchers in the field of robotics, interactive computer interfaces and so on. Before Microsoft released its official driver and SDK for the Kinect [192] in June 2011, numerous people had been devoted to reverse engineering efforts who managed to program their own drivers to access the Kinect to extract the depth data and colour images. A number of applications have been demonstrated on the Internet.

In addition to a depth sensor, the Kinect comprises a RGB colour camera, a multi-array microphone and a motorized pivot as shown in Figure 2-4. Its depth sensor has a wide field of view (43 ° vertically by 57 ° horizontally) and its data stream is at a real time frame rate (30 fps).

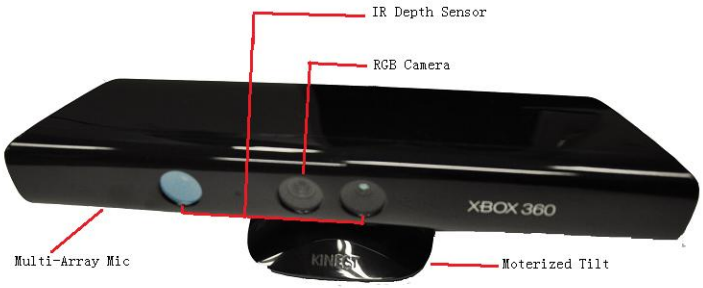


Figure 2-4: Kinect

It is worthwhile to emphasize that the approaches proposed in this thesis are basically independent of the employed sensing devices employed, provided that they provide valid colour and depth data. Most of the experiments with the PMD camera have already been re-conducted with the Kinect following its launch. In fact, the experimental results in Chapter 5 show that both devices produce similar performance in terms of the recognition rate and accuracy, although the Kinect enjoys the advantages of wider FOV, higher resolution and higher frame rate compared to the PMD camera.

Although our current application concerns mostly the indoor environment, it is also worth mentioning that the PMD camera generates a better quality of data compared to the Kinect when exposed to sunlight. In an outdoor experiment, the PMD was still able to detect the pillar as shown in Figure 2-5, while the Kinect hardly provided any valid data. This is probably because the PMD camera uses modulated signals so that it suffers less influence from sunlight.

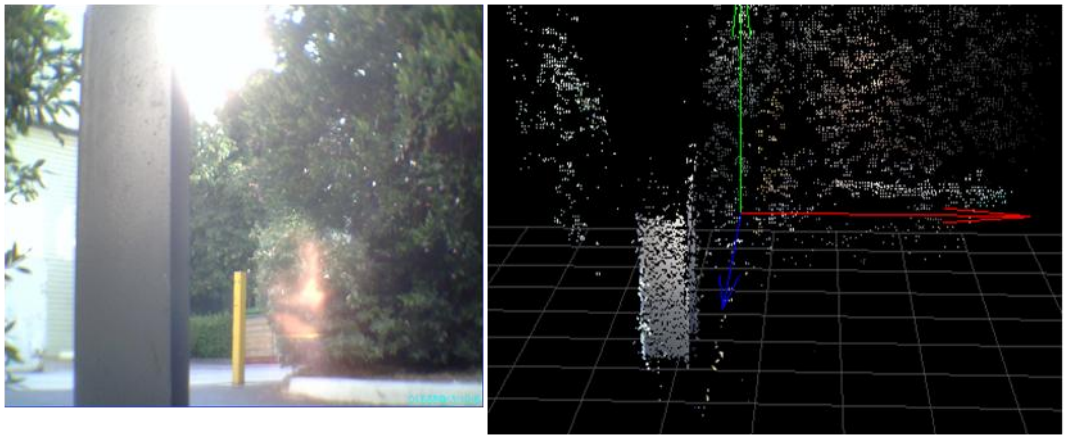


Figure 2-5: In an outdoor experiment, the PMD camera could still detect the pillar when it was exposed to the sunlight. The left picture from the web camera shows the scene and the right image plots the 3D data from the PMD range camera. However, Kinect hardly provided any valid data in this situation.

2.3 Image Registration and Camera Calibration

Camera calibration is a classical computer vision approach to determining the intrinsic parameters of a camera (such as lens distortion, focal length and optical centre on the image plane), and also the extrinsic parameters (i.e. the rotation and translation matrix with regard to a specific world coordinate frame). The calibration result is necessary for several purposes, including epipolar lines alignment, image rectification and image registration. In this section, the mapping method between the depth image and the colour image is first introduced, and then the details of the calibration procedures for the PMD camera set and the Kinect are presented.

2.3.1 Mapping between the Depth Image and the Colour Image

Based on a pinhole model, the perspective projection from 3D camera coordinates $P[x, y, z]^{\mathrm{T}}$ to the image coordinates $p[u,v]^{\mathrm{T}}$ can be obtained using Equation (2.1) [72]

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z} \cdot \begin{bmatrix} \alpha & -\alpha \cdot \cot\theta \\ 0 & \dfrac{\beta}{\sin\theta} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{2.1}$$

where α, β, θ are the distortion parameters of the camera, and $[u_0, v_0]^T$ is the optical centre on the image plane.

Given a depth image, the colour information for each pixel can be obtained using a mapping procedure. Take the pixel $p[u,v,d]^{\mathrm{T}}$ in the depth image as an example. First its 3D coordinate in the depth camera coordinate system $P_d = [x_d, y_d, z_d]^T$ is calculated

$$\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \frac{d}{\sqrt{f^2 + (u-u_0)^2 + (v-v_0)^2}} \cdot \begin{bmatrix} u-u_0 \\ v-v_0 \\ f \end{bmatrix} \tag{2.2}$$

where, f is the focal length of the camera. When $f>>u-u_0$ and $f>>v-v_0$ (i.e. $z_d>>x_d$ and $z_d>>y_d$), z_d is approximately equal to d and Equation (2.2) becomes

$$\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \frac{d}{f} \cdot \begin{bmatrix} u-u_0 \\ v-v_0 \\ f \end{bmatrix} \tag{2.3}$$

The coordinate is transformed from the depth camera coordinate $P_d = [x_d, y_d, z_d]^T$ to the colour camera coordinate $P_c = [x_c, y_c, z_c]^T$ using the extrinsic parameters $P_c = RP_d + T$, where R is the rotation matrix and T is the translation matrix between the two cameras. Finally, $P_c = [x_c, y_c, z_c]^T$ is projected onto the colour image using Equation(2.1).

It is obvious that the above mapping method of assigning colour information to the depth image requires the intrinsic and extrinsic parameters of the two cameras. The calibration procedures to obtain these parameters are presented in the following sections.

Using the above method, it is possible to map the colour information to the depth image and generate a 3D point cloud with colour value for each point. However, this generated 3D image may be quite noisy due to inaccurate depth data or incorrect calibration parameters, and its size is the same as the depth image. Therefore, this generated image may be low-quality or low-resolution for some computer vision applications, such as face detection and tracking (see Chapter 3). Therefore, it is preferred to first implement the detection and tracking on the original colour image, and then find the corresponding point in the depth image. Nevertheless, in theory, it is impossible to find accurate correspondences, because the depth value is needed to back-project a 2D point on the image to the 3D space as illustrated in Equation (2.2). Therefore, we aim to find approximate correspondences. First, the two cameras are placed carefully so that the two image planes are approximately parallel. Secord, the image from the web cam is divided

into several grids. Assuming that in each small region there is a linear function (Equation (2.4)) locating the pixel in the depth image (x', y') from the web image (x, y), several correspondence points can be manually picked out in each region of the colour-depth image pairs. These points are used to build the mapping function by the Least Square method. In this way, the parameters a, b, Sx and Sy are calculated. Note that they have different values in different grids of the image.

$$\begin{cases} x' = \dfrac{x-a}{S_x} \\ y' = \dfrac{y-b}{S_y} \end{cases} \tag{2.4}$$

Finally, the correspondences are searched for in a small window by the Normalized Cross Correlation (NCC) method [36]. NCC is a standard statistical method for determining similarity. Its normalization, both in the mean and the variance, makes it relatively insensitive to the gain and bias of the images. The colour image from the web camera is first converted to greyscale and the matching score between the pixel $I_1(x, y)$ in the webcam image and the pixel $I_2(x + d_1, y + d_2)$ in the depth image is calculated by Function (2.5) [36]. By adjusting d_1 and d_2, the local minimum of the matching score is obtained which represents a correspondence.

$$\frac{\sum_{u,v}(I_1(x,y)-\overline{I_1})\cdot(I_2(x+d_1,y+d_2)-\overline{I_2})}{\sqrt{\sum_{u,v}(I_1(x,y)-\overline{I_1})^2\cdot(I_2(x+d_1,y+d_2)-\overline{I_2})^2}} \tag{2.5}$$

Although the proposed mapping method from the colour image to the depth image is only an approximation, it performs well for the face detection application (see Chapter 3) and hand tracking (see Chapter 4).

2.3.2 Calibration of the PMD Camera Set

Stereo camera calibration aims to find the intrinsic parameters of both cameras and the extrinsic parameters (translation and rotation matrix) between the two. We considered the PMD camera and the web camera as a stereovision system and followed the common stereo calibration procedures. A flat checkerboard was used to provide corner feature

points. A set of image pairs, greyscale images from the PMD range camera and colour images from the RGB web camera, is captured simultaneously.

Due to the poor quality of the greyscale images from the PMD camera, pre-processing is necessary; otherwise, accurate corner feature selection cannot be achieved. A histogram equalization method [2] is used to enhance the contrast of the greyscale image. The images are also rescaled to 640×480 resolution using a bi-cubic resampling strategy. The corners of the grids on the checkerboard can be more easily and accurately located after these two steps of processing. The original images from the web camera are converted to greyscale, but without any further processing because their quality is sufficient for calibration purposes.

The existing camera calibration toolbox for stereo camera calibration [28] is utilized. This approach uses several image pairs of a planar checkerboard with different poses to estimate the intrinsic parameters of both cameras and their relative positions (i.e. extrinsic parameters).

More sophisticated and accurate calibration methods with regard to distance deviation due to the object reflectivity, temperature, accuracy deviation related to distance, shutter time and radiation behaviour of the illumination subsystem can be found in [131, 132, 239]. A track line is usually employed for the precise measurement of the ground truth of the reference distances. Some special devices, e.g. a special planar object with 25 infrared LED built by Kahmann et al. [132], were designed to deal with the low resolution deficiency of the sensor.

With the calibration result, the depth reading from the PMD range camera and colour information from the web camera can be registered to each other, as described in Section 2.3.1. An example is shown in Figure 2-6.

2.3.3 Calibration of the Kinect

Data from the IR camera of the Kinect is mainly used to generate depth maps, but they can also be adopted for calibration purposes. Burrus [39] generates greyscale images using the IR data so that the subsequent calibration procedures are the same as for normal digital cameras using the MATLAB toolbox [28].

Herrera et al. [103] proposed a novel method to simultaneously calibrate a colour camera, a depth camera and their relative pose. An initial estimation for the calibration parameters was obtained for each camera independently. For the colour camera, the checkerboard corners were extracted from intensity images as normal. For the depth camera, the four corners of the calibration plane were selected since the checkerboard was not visible in the depth image. Then the initial guesses of intrinsic and extrinsic parameters were refined using a non-linear optimization method to minimize the weighted sum of squares of the measurement re-projection errors. The error for the colour camera was the Euclidean distance between the measured corner position and its reprojected position. While for the depth camera, the error was the difference between the measured and the predicted disparity. Their experiments showed that the method had comparable accuracy to that was provided by the manufacturer. This method is applicable not only to the Kinect, but also to any stereo configuration that consists of a digital camera and any type of depth sensor. An example of the mapping result is shown in Figure 2-7.

2.4 Data Processing

The depth data provided by the PMD range camera contains a certain amount of error from several sources [158, 182]. First, a low Signal-to-Noise Ratio (SNR) indicates an inaccurate measurement; while increasing the exposure time and amplifying the illumination may cause an oversaturation problem. Second, because of the multi-reflection effect, the depth data on the edges of the objects appear to merge into the background in a smooth transition. Third, due to inhomogeneous illumination, the peripheral part of the scene may have a lower SNR than the central part. Fourth, although the working principle is based on an assumption that the transmitted light is sinusoidal, this is not exactly accurate in reality, which results in a distance-related error. Fifth, the non-linearity of the electronic components causes an amplitude-related error. Sixth, there is a fixed pattern of phase noise because the pixels on the sensor chip are connected in series, and the triggering of the pixels depends on their locations.

Some of the errors are inherent in the working principle of the device and cannot be corrected; others can be corrected or compensated for by distance calibration procedures and data processing methods [131, 170, 183] . Considering the complexities of different correction methods and the practical requirement in our application, the following data processing methods were implemented.

The standard deviation of the range depth data has been found to be reciprocal to the signal amplitude [182]. Thus the pixels, whose amplitude of the received signal is below a specified threshold, are removed. Then a two dimensional Median Filter is applied on the depth data array, which effectively reduces random speckle noise. However, the so-called *jump edges* effect still remains. The points on the edges of the objects tend to merge into the background, which is also called the 'tail comet' effect in [34] . An example of this effect is shown by the distribution of the points on the edge of the arm in Figure 2-6. One reason for this error is the limited resolution of the sensor chip [12], which results in multiple reflections from the edges and the background arriving at the same pixel on the sensor. The measurement is an averaging result of the distances from multiple paths. May et al. [183] proposed a method to filter out the points on the edges of objects, And we have made a modification to their method. For a given point $\boldsymbol{p}(x,y,z)$ and its eight neighbours $\boldsymbol{P} = \{p_n \mid n = 1\ldots 8\}$ in 3D, calculate

$$\theta = \max \arccos\left(\frac{\| p \| \cdot \sin \varphi_n}{\| p - p_n \|}\right) \tag{2.6}$$

where φ_n is the apex angle between the two points. If θ is bigger than a threshold, the point $\boldsymbol{p}$ is considered on an edge. However, the method is sensitive to noise, i.e. the valid points may also be removed if suitable noise reduction filters are not used first.

2.5 Conclusions

This chapter has introduced the hardware configuration of our robot and the features of the 3D sensing devices. The methods of camera calibration, image registration and data processing have also been described.

A: image from web camera showing the scene

B: mapping from the depth image to the colour image. The depth data contains a large amount of noise

C: The depth data in figure B is processed by removing low SNR points and then filtered by a 2D Median Filter. But the jump edge effect remains.

D: The depth in figure C is further processed by a jump edge filter. Comparing with figure C, the points on the edge of the arm have been removed.

Figure 2-6: An example of mapping between the colour image from the RGB web camera and the depth image from the PMD range camera and the data processing result.

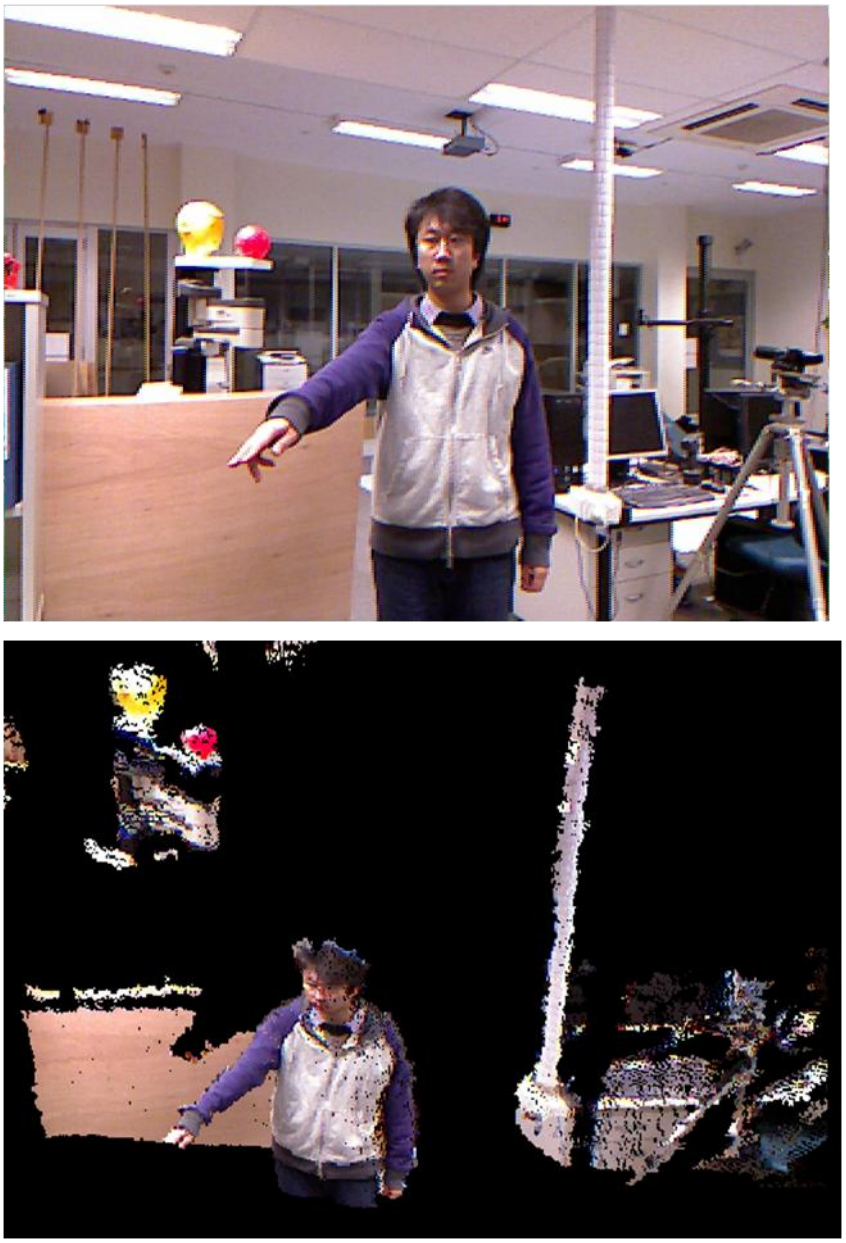


Figure 2-7: Mapping result of mapping the colour image and the depth image from the Kinect. It can be seen that, compared to the PMD camera, the Kinect has higher resolution, wider FOV and better quality of the depth data

3

Face Tracking and User Identification

For domestic assistive robots, awareness of the presence of humans is essential, partly because of safety concerns. The activity of the robot needs to be carefully planned when it is in a shared environment with humans as discussed in [198]. It is also related to their capability of effective interaction with humans. Identifying a user also plays an important role during human-robot interaction. Knowing who the user is could be advantageous for more effective customised interaction and robust performance. Different interpretation methods can be used for individuals, which can lead to more appropriate responses.

This chapter is organized as follows. A brief review of the related work is first presented, followed by our face detection and tracking method. Then the importance of user identification in our system is emphasized. Finally, future work is discussed. Part of the work in this chapter was published in [164, 166].

3.1 Related Work

3.1.1 Face Detection

According to a survey of methods for human presence detection [277], most early research work in person detection, such as [264] and [259], depended on background subtraction. The basic assumption in their work is that the background is either static or changing very slowly. The main advantage of this approach is that it is simple to implement and runs fast. It allows quick detection of non-background objects, which has been widely applied in visual surveillance applications. For dynamic backgrounds, it is advantageous to exploit depth information to simplify background subtraction. After the foreground objects have been extracted, object segmentation processing and classification

methods are used to identify persons or other objects of interest, based on shape, colour, motion or other features. For instance, Darrell et al. [55] searched for a human-sized connected components for person tracking and the same approach was used in [69] for pedestrian detection. Besides simple features like size, shape and height/width ratio, some classification approaches can also be employed to identify which of the connected components in the foreground represents a human. For example, a Support Vector Machines (SVMs) method was used in [174] to find and track the user in a human-robot teaming application.

Many systems require the accurate detection of a person's face rather than a rough estimation of his/her position. The extracted face images can be potentially used for the recognition of a person's identity. Face detection plays an important role in human-centric systems, such as video conferences and intelligent human-computer/robot interactions. Face detection is not a trivial problem, because a human face is a dynamic object and has a high degree of variability in location, scale, orientation, pose and facial expression. To achieve robust performance, the approach also needs to be invariant to lighting conditions.

According to [104], face detection approaches are mainly divided into two types: feature-based or image-based. Feature-based approaches extract the low level visual features from greyscale or colour images. Edge representation was applied in the early work of face detection by Sakai et al. [246]. This kind of methods normally includes an edge extraction step implemented with curvature constraint to prevent it from being distracted by the noisy edges of the background objects. Edges need to be labelled and matched to a face model based on the shape and position information of the face in order to verify correct detections [104]. More examples of face detection using edge-based techniques can be found in [79, 260, 300]. In addition to edges, the greyscale information of a face can also be used as features. Facial features such as eyebrows, pupils and lips appear darker than their surrounding regions. This property together with some basic face geometry properties can be exploited to locate the face and differentiate various facial parts. Some examples can be found in [13, 81, 293]. Compared to the greyscale information, colour is a more powerful means of discerning objects [104]. Skin colour has been widely employed for face and hand localization and tracking tasks [84, 211, 266] because of its fast implementation. The RGB colour format is usually mapped to other spaces which separate intensity and chrominance components. For example, Hue, Saturation, and Value (HSV) colour space is used in [113], YUV (where Y denotes the intensity, while UV

specify chrominance components) is used in [172], and YCrCb (where Y is the luminance component and Cr and Cb are the red-difference and blue-difference chrominance components) is used in [58]. It has been found that the chrominance components of the human skin colours, even among difference races, take up a very tight cluster in the above colour spaces. It is the intensity of the different types of skin that makes them look different [112, 184, 295]. By comparing the colour information of a pixel with respect to that of the pre-built skin colour distribution model, the likelihood of the pixel belonging to a skin region can be deduced. Colour segmentation can simply be performed using appropriate skin colour thresholds. The skin colour can be modelled through histograms [78, 108] or Gaussian distribution models [215, 295]. However, there is normally an implicit assumption that the face and the hands are the only skin-coloured objects in the scene and subsequent processing is required to further differentiate the face from the hands and arms. This can be done using motion cues [268] or by assuming that the face is the largest connected patch with skin colour in the foreground [212].

However, features generated from the low-level analysis tend to be ambiguous. For instance, when using a skin colour model, face detection is prone to be distracted by background objects of a similar colour. This is a classical many to one mapping problem, which might be solved by higher level feature analysis methods. One popular idea is that the confidence of a feature's existence is enhanced by the detection of nearby features [104]. A pair of eyes is the most commonly applied reference feature due to its distinct side-by-side appearance [13, 15, 80]. Other features such as nose, mouth and eyebrows can also be used to determine the most likely face candidate, as shown in [119].

More robust modelling methods might be necessary because the above feature searching approaches may fail due to their rigid nature, when faces appear in various poses against complex backgrounds. For instance, probabilistic face models based on multiple face appearances have been proposed in [272] and [298].

Unlike searching for low level face features, active shape models depict the higher-level appearance of faces. In these approaches, an active shape model interacts with local image features (edges, brightness) and gradually deforms them to obtain the shape of the feature [104]. There are generally three types of active shape models which have been widely used in face feature extraction applications, namely snakes [111, 140, 157, 214], deformable templates [83, 299] and point distributed models [50, 159, 160].

While face detection by explicit modelling of features often has trouble with the variability of face appearance and environmental conditions, image-based approaches are considered to be more robust. Image-based approaches treat face detection as a pattern recognition problem. The basic concept is to build face patterns via a training stage which classifies examples of face and non-face prototype categories. The simplest image-based approach is template matching [105]. Other image-based approaches include linear subspace methods, neural networks and statistical methods [104]. For example, in [195] the test images and training samples are projected to a subspace by the Principle Component Analysis (PCA) method. In [218], a Support Vector Machines (SVMs) method is used to determine whether to classify the image in the search window as in the face or non-face class. Most image-based approaches apply a window scanning technique for detecting faces. It is basically an exhaustive search of the input image for possible face locations at all scales. The scanning window size, the down sampling rate, the step size and the number of iterations vary in different proposed methods [104]. Although robust, they are normally computationally expensive. Viola and Jones [283] built a real-time face detection system using the AdaBoost method [74], which was considered a dramatic breakthrough in face detection research. The method includes a cascade structure of increasingly more complex classifiers. Because most of the non-face regions are discarded by simple classifiers in the early stages, computation time is mainly spent on the promising regions. Therefore, this algorithm runs quickly.

3.1.2 User Identification

Generally, a complete and applicable face recognition system includes five components: face detection, alignment, pre-processing, representation and classification [236]. Face detection approaches have already been reviewed in the previous section. Aligning a face image with a reference image requires finding the correspondences between the two images. A small number of feature points on the face like the eyes, the nostril and the corners of the mouth are usually selected to calculate the correspondence. Based on these correspondences, the input face can be warped to a desired orientation. In [196] an affine transformation is used to perform the warping. Pre-processing of face images normally consists of de-noise procedures and brightness adjustments in order to enhance the image quality for subsequent processing. Face representation aims to extract face features and build descriptors which can be used to distinguish between different faces

in the following classification step. Among these 5 components, face representation plays the most essential role and a vast amount of work has focused on this topic. Therefore, a review of face representation approaches is provided in the following section.

The Eigenface method is one of the most thoroughly investigated and widely used approaches [280]. Eigenfaces are eigenvectors of the covariance matrix of the set of face images. Each face can be represented by a linear combination of the eigenfaces. Normally, an approximation using only a small number of eigenvectors which correspond to the largest eigenvalues is chosen to construct a 'face space'. Pentland et al. [227] extended the idea of eigenface to eigenfeatures corresponding to various face components, such as eyes, nose and mouth. This method has been reported to be less sensitive to appearance changes than the standard Eigenface method. However, in general, Eigenface, though fast and simple, does not provide invariance over scale and lighting condition changes [279].

The Neural Network method is attractive due to its non-linear nature. Lin et al. [169] used a Probabilistic Decision-based Neural Network (PDBNN) which divided the network into several subnets. Each subnet was dedicated to recognize one person in the database. It was reported that the PDBNN recognizer was capable of recognizing up to 200 people and achieving up to 96% recognition rate. However, although the classification time was less than 0.5 second, the training time was 4 hours. Moreover, the Neural Network approaches normally require a very large number of training samples.

SVMs, as an effective method for general pattern classification purposes, have also been widely employed for face recognition. For example, Guo et al. [90] used SVMs with a binary tree recognition strategy. In [129], a client-specific solution was adopted which required learning client-specific support vectors. Jonsson et al. [128] used SVMs face authentication. They concluded in their work that (a) the SVMs approach is able to extract the relevant discriminatory information from the data automatically; (b) it can also cope with illumination changes; (c) however, SVMs involve many parameters and can employ different kernels. This makes the optimization space rather extensive, without the guarantee that it has been fully explored to find the best solution.

Graph matching [155] and template matching [37] have also been used for face recognition, but one drawback is that their computational complexity is very high.

3D face recognition is a newly emerging research field. Many approaches have been proposed to recognize faces using 3D data alone or combing it with 2D intensity images. Curvature-based segmentation and Extended Gaussian Image (EGI) were used in [77, 162, 275]. Lee et al. [163] located the nose tip first and then formed a feature vector based on the contours along the face. Achermann [3] extended the 2D Eigenface method for range images and used the Hidden Markov Models (HMMs) approach for recognizing 24 persons, with 10 images per person. They reported 100% recognition rate in their experiment. Beumier and Acheroy [16] proposed a multimodal approach by using a weighted sum of similarity measures in 3D points and 2D images. Because most of the 3D recognition algorithms assume the face is a rigid shape, 3D methods are actually more negatively affected by facial expressions than 2D methods [29]. Bronstein [35] used an isometric transformation method aiming to cope with face shape variation due to facial expressions. First, the range data and the face image were acquired. Next, the range data were pre-processed by removing certain distracting parts such as hair. Finally, a canonical form of the facial surface was computed and used for recognition.

In conclusion, current face recognition systems are generally not sufficiently robust if there is variation between test and training conditions. Changes in illumination, head pose, facial expression, hair style and eyewear could confuse the systems [279].

In addition to faces, gait is also considered a prominent feature for the identification of a person. Gait recognition has only recently started to attract high levels of attention in the computer vision community. Most gait recognition methods are strongly dependent on accurate silhouettes of a person, and tend to fail when people wear different clothing, carry objects or when the environment is highly cluttered which causes increased segmentation errors [277]. Compared to the silhouette, body joints' positions are considered to be more robust for gait recognition. For example, in [130, 213, 276], body joints were extracted first to construct motion features and then various classification methods were applied. Gait can be observed at a distance, making it particularly useful for detecting uncooperative persons.

3.2 Face Detection and Tracking

3.2.1 Skin Colour Model

Colour information plays an important role in our system, for instance, for face detection and tracking (refer to the following section), and hand tracking (see Chapter 4). An image template [71] which contains comprehensive types of skin image samples from people of different races is used to build a skin colour distribution model. The image template is first projected into CIE Lab colour space and then a 2D histogram is built for the 'a' and 'b' chrominance components (The 'L' component which corresponds to the image intensity is discarded). The histogram is normalized to represent a probability model of the distribution of the skin colour, which is then smoothed using a 2D Gaussian Filter. Normalization in this thesis means each item of data in the set is rescaled so that their summation equals one.

In the query stage, the RGB colour at each pixel is also projected to the CIE Lab colour space, and the probability of the colour belonging to the skin colour tone is then found by the function $p(a, b)$ which represents the 2D skin colour model.

Figure 3-1 shows an example of a skin colour detection result. The intensity in the picture Figure 3-1 (b) represents the probability of each pixel having the skin colour tone (the value is rescaled for the purpose of illustration). It can be seen that most of the skin surface has been detected, but the objects in the background may also have a similar colour.

(a). A sample of the colour images of the experiment scene.

(b) The skin-coloured pixels corresponding to the left image. The intensity represents the probability of each pixel having the skin colour tone according to the prebuilt skin colour distribution model

Figure 3-1: An example of the skin colour detection

It is important to note that actually projecting RGB colour to many other colour spaces such as HSV, YUV and YCrCb can also achieve a similar result. The essential feature of these colour spaces is that they separate the chrominance components from the image intensity.

3.2.2 Face Detection and Tracking

3.2.2.1 Face Detection

In this section, the widely used face detector which was developed by Viola and Jones [283] and extended by Lienhart and Maydt [168] is adopted, and the false positives then are eliminated by colour and depth constraints.

Viola and Jones [283] proposed a machine learning approach for rapid visual object detection and demonstrated this on a face detection application. This approach utilizes three types of simple Haar-like features which can be calculated very quickly by their proposed 'integral image' representation. Given a training set of positive and negative face/non-face images, a variant of AdaBoost is then used both to select a small set of critical features and train efficient classifiers. Finally, increasingly more complex classifiers are constructed in a cascade structure so that background regions of the image are quickly discarded. At each layer, the sub-windows which contain non-face regions are rejected by the classifier and the positive regions (which are more likely to be faces) enter the next layer and are examined by a more complex classifier. After several layers, the number of sub-windows has been reduced radically. This method is very efficient because the majority of background regions are rejected by simple classifiers at the early stages and most computation is spent on promising face-like regions. Lienhart and Maydt [168] extended the Haar-like feature set to include 45° rotated features and proposed a post-optimization procedure. The adopted implementation using OpenCV is available online [217]. The Haar-like cascade classifiers provided in the package are also used in order to avoid the task of manually labelling a large number of images as face and non-face regions.

However, using this method, false positives may occur, i.e. non-face objects are recognized as faces. Taking advantage of the colour information from the RGB camera and the depth data from the range sensor, the face detection results are refined in two ways.

First, using the pre-built skin colour model described in Section 3.2.1, the false positives can be reduced by the colour cue. The detected face region should have a non-zero skin colour probability value and the number of pixels with skin colour in the detected face region needs to exceed a specific threshold. Second, the false positives whose distances to the robot are either too far or too close are removed, since we assume that effective interaction occurs within a reasonable distance (1 to 4 meters). Figure 3-2 shows an example where the proposed method located the face position more robustly by eliminating the false positives detected by the Viola-Lienhart algorithm. To obtain the person's distance to the camera, the correspondence point of the face centre in the depth image needs to be found. This can be achieved using the mapping method between the colour image and the depth image described in Chapter 2. Then the 3D position of the face can also be calculated (Equation 2.2).

Figure 3-2: The left image shows the face detection result by the Viola-Lienhart algorithm (the blue circles indicate the face positions and sizes) which was implemented in OpenCV. It generated two false positives, which were eliminated by the proposed method as shown in the right image.

3.2.2.2 Face Tracking

The above face detection method suffers from the inability to handle significant head rotations (approximately 30). In addition, although face detection in each frame is fast enough for a real time application, we need to further shorten the processing time, since locating the user's face is only the first step in our system. Therefore, the face detection is only carried out in the initialization and re-initialization stages and then the face is tracked by the CAMSHIFT algorithm.

The CAMSHIFT algorithm has been widely adopted for object tracking ever since it was proposed by Bradski [31]. CAMSHIFT is based on the Mean Shift algorithm [46],

which is a non-parametric approach to searching for the local maximum of the probability distribution in the direction of density gradients. CAMSHIFT extends the Mean Shift algorithm by adaptively changing the size of the searching window for visual tracking in video sequences. For the face tracking in our system, the initial position and size of the face searching window is determined in the above-mentioned face detection method. Then the RGB colour is projected to the HSV colour space, and the colour distribution histogram of the face region is calculated using the 'Hue' component. After initialization, the face position is searched within the searching window by the Mean Shift algorithm until convergence and the searching window is resized and relocated according to the current size of the face area.

The tracking method not only saves processing time, but is also able to trace the face position when the face detection algorithm fails, e.g. when the head turns to the side. However, tracking can be lost due to distracting objects in the background. We consider the following as the indication that the tracker has a low confidence: (a) the centre of the tracked face stays at a fixed position for a long time; (b) similar to the face detection procedure, the number of pixels of skin colour in the tracked region is too small; and (c) the face is too far from or too close to the camera. When these occur, the face detection procedure is called again to re-initialize the face position and the searching window size for CAMSHIFT tracking.

It is worth mentioning that the other two programs ('VeriLook' for face recognition in section 3.3, and 'FaceLAB' for head pose and eye gaze estimation in Section 6.2.2) which are employed in our system can also track the position of the face. However, they have their own limitations. For instance, 'VeriLook' can only detect the person's face when he/she is very close to the camera (within a distance of 1.5 meters), and looks straight at the camera so that the face image is of good quality. 'FaceLAB' has a shallow field of depth and a narrow field of view, which means the user's movements are constrained. The method proposed here allows the user to move freely as long as he/she is in the view of the camera and within a distance of 1 to 4 meters.

3.3 User Identification

User identification is fundamental and essential for effective communications among people in our daily life. In our human-robot interaction system, the user's identity provides

significant benefits for the robot to achieve more effective interaction and robust performance. Individuals have different behaviour habits which are established from their personal experience and cultural background. Therefore, they may have various types of gestures even to express the same meaning. It is advantageous that the robot knows each user's preference and gives appropriate responses according to the user with which it is currently interacting. For example, the evaluation of a pointing direction will be more accurate if the person's preferred pointing method is taken into account (see Chapter 5 for details). Another example is that speech recognition can be significantly improved using the users' speech profile, which is built in the training stage (see Chapter 6 for details).

In our system, a profile is built for each individual. This includes several characteristics of a specific user, such as the body height, a set of self-defined gestures, the pointing method preference, a trained speech profile, a look-up table for mapping between certain pronunciations in specific contexts, temporal relationship between pointing gestures and speech and request frequency of certain objects. Some of these characteristics are assigned beforehand, and others can be learned and updated during the interaction. On the one hand, the robot's performance can be more effective and robust using individuals' profiles, which will be discussed in the following chapters in detail. On the other hand, the system should also be able to interact with a new/unrecognized user. If the robot has not met the person before, the person can either choose to ask the robot to enter a learning mode, so that he/she can teach the robot to remember his/her appearance and setup his/her personal profile, or the robot can use the default profile which contains generic parameters to interpret his/her visual and voice input.

To realise this idea, a commercial face recognition program called 'VeriLook' [209] is adopted. In the training stage, for each person, an image containing his/her face is captured and saved in the database. In the test stage, it gives a matching score between the test image and each enrolled face image. In a test carried out within a small group of 8 people, the software could successfully recognize each person when the face image was of good quality and facial features could be extracted. However, when the person is standing far away from the camera, the software often fails to extract facial features and thus the person cannot be recognized. The working distance generally depends on the video zoom, resolution and quality of the images.

Besides facial features, the height of the body also contains some information on a person's identity. Although this feature alone is not able to give accurate identification results, it provides another cue and may be useful when the face recognition system fails or returns an ambiguous result, e.g. the similarity scores between an input image and two or more samples in the database are very close. The height information is used only in this situation in our system. The person's height information can be inferred from the 3D face position. The height value is compared with the records of all users or the users who have large scores returned by the face recognition program. The robot will seek confirmation of the person's name by uttering the name using a synthetic voice.

During the interaction between a user and the robot, the identification procedure is called into action in the following two situations: (a) when the robot first detects a person in front of it and (b) the person leaves and returns later or a new person enters (i.e. the tracked face disappears and a while later a face is detected).

3.4 Conclusions and Future Work

This chapter has presented the approaches for face detection and tracking and user identification. The well-known Viola-Lienhart face detection method is enhanced by the inclusion of colour and depth information. The CAMSHIFT algorithm is adopted for face tracking not only to save processing time but also to track the face when the face detection method fails. Consideration of individual differences in behaviour is an essential feature in our system, which may enable more effective and natural interaction. To realise this, a commercial face recognition program and the body height information are utilized to identify a user. A profile that is built for each individual user can help to achieve more effective interaction and detailed discussions will be provided in the following chapters.

Currently, we assume that only one user is communicating with the robot at one time. In the future, it is possible to allow multiple users to interact with the robot alternately by detecting the person who is speaking to the robot. This has been inspired by some lip tracking work. For instance, Tian et al. [278] presented a lip tracking approach by combining shape, colour and motion.

Rather than waiting for the user to come to it to initiate a conversation, in future work, the robot will be able to detect a person waving a hand at a long distance and come

to him/her. This can also be done by localizing the sound source when a person calls the robot's name.

4

Hand Gesture Recognition

The direct use of hands as a communicative or manipulative input channel is attractive in the field of Human-Robot/Computer Interaction. Natural hand gesture recognition allows a wide range of applications in sophisticated interaction environments, such as augmented reality [250] and smart rooms [202]. Of the different body parts, hands provide the most natural, intuitive and effective interaction method. Both hand postures (i.e. static gestures) and motion patterns (i.e. dynamic gestures) have been applied in control and communication interfaces.

In this chapter, two hand tracking methods are proposed to locate the gesturing hands. A simple method based on the depth information for hand extraction works under some constraints, while a more robust 3D Particle Filter-based hand tracking method combining the colour, depth and motion cues generates a complete hand trajectory for dynamic motion pattern recognition. The hand shapes are segmented, rectified and matched with the templates in the database. Part of the work in this chapter was published in [116, 165].

This chapter is organised as follows. After a review of the related work, the proposed methods for hand tracking, dynamic motion patterns recognition and static hand posture recognition are described in three sections, respectively. The results of experiments are presented in each section.

4.1 Related Work

4.1.1 Hand Tracking

The positions of the hand(s) are essential for subsequent processing such as hand segmentation, hand shape recognition and motion pattern recognition in most of the

gesture recognition systems. Currently the most effective and accurate tools for capturing hand motion and joint angle configuration are electro-mechanical or magnetic sensing devices (e.g. data gloves) [68, 270]. These devices deliver a complete set of real time measurements independent of the experimental environment; however, they are very expensive, require complicated calibration procedures and, most importantly, hinder the naturalness of hand motion in daily use.

Computer vision technology provides more natural and non-contact interaction methods. Motion, colour, contour and shape have been used as features for tracking. For example, many approaches [113, 194, 266] used colour information, since the skin colour is a salient feature different from most of the background objects. Edge and motion information is extracted as the features in [110]. Lu et al. presented a model based approach to combining edges, optical flow and shading information [176]. Compared to colour, depth is more robust to illumination conditions. Liu and Fujimura detected the user's hand using only a simple depth constraint [173]. Nickel and Stiefelhagen used a stereo camera system to track the hand in 3D space [212].

Although hand detection can be implemented in each frame independently, tracking methods are often used to speed up the performance by reducing the region for calculation. Particle filter [190] has been widely adopted for hand tracking [32, 40, 210, 286]. As it uses multiple hypotheses, it has advantages in visually cluttered environments compared to single hypothesis tracking methods such as CAMSHIFT and the Kalman Filter [134].

4.1.2 Spatial-temporal Hand Motion

Hidden Markov Models (HMMs) have been used prominently and successfully in the field of speech recognition [237] and gesture recognition [266, 289]. Automatic sign language recognition research began to appear in late 1980s. Tamura and Kawasaki [274] presented an image processing method which could recognize 20 Japanese signs. Starner and Pentland [266] proposed an HMMs-based system for recognizing American Sign Language (ASL) at the sentence level. Eight element features including 2D hand position, angle of axis of least inertia, and the eccentricity of bounding ellipse were chosen to form a feature vector. In their experiments, no intentional pauses were placed between signs within a sentence but the sentences themselves were distinct. Liu [172] introduced a visual system for recognizing 26 hand written letters using the HHMs method. A sequence of

angles of motion along the trajectory was calculated and quantized to form a discrete observation. Mori et al. [200] used Continuous Hidden Markov Models (CHHMs) to model human daily-life actions. Their hierarchical recognition structure started from the root and competed among the likelihoods of child-nodes.

The Finite State Machine (FSM) algorithm has also been employed to recognize gestures. Kang-Hyun et al. [137] used this approach to recognize manipulative hand gestures such as grasping, holding and extending. By using a rest state, all unintentional actions were considered as taking a rest and ignored. Davis and Shah [56] used this technique to recognize simple hand gestures. The FSM was used to model four qualitatively distinct phases of a generic gesture. Gestures were represented as a list of vectors and were then compared to pre-stored gesture vector models. Recognition of seven gestures for actions of "left, right, up, down, grab, rotate and stop" were demonstrated.

Dynamic Space Time Warping (DSTW) was used by Alon et al. [6] to simultaneously locate and recognize dynamic hand gestures. The dynamic programming method was used to compute a global matching cost and a warping path to align the query and model gestures and find the best hand candidate in each frame.

4.1.3 Static Hand Posture

Hand posture estimation mainly involves extracting the shape and adjusting the orientation of the gesturing hands and then matching the new capture with pre-stored templates to classify their meanings.

For static hand posture recognition, although it is possible to recognize hand posture by extracting some geometric features such as the position and orientation of each fingertip, such features are not always available nor reliable due to noise, self-occlusion and variation of lighting conditions [296]. Wu et al. [291] used the dot-product of two vectors on the contour points to approximate the *k*-curvature in order to locate the fingertip in images. Another method [137] found the farthest point from the centre of the hand. Only one pointing finger was located using this method. Sensitivity to noise is a serious problem for these methods in the case of noisy silhouettes and contours.

Alternatively, the contour and silhouette of the whole hand can also be adopted as features of the hand shape. Machine learning methods such as SVMs, Neural Network and

Nearest Neighbour, are then used to recognize the meanings of the hand postures. [24, 121, 179].

Some researchers have used 3D Model-based hand matching methods [34, 47, 267] to achieve an accurate reconstruction and animation of the hand. The hand model is placed into a captured 3D point cloud and then its pose is iteratively adjusted so that the overall distances between the points on the surface of the model and the measurement are minimized. Good initialization is normally essential to prevent the iteration from converging to a local minimum. Articulated three-dimensional models built by computer graphic technology often consist of a large number of vertices and triangles and have high Degrees of Freedom (DOF), which often cause these methods to be computationally expensive.

4.2 Hand Tracking

Hand tracking is a crucial component of hand gesture recognition. The gesturing hands must be located first, so that recognition of static hand posture or dynamic hand motion is possible. Depending on the application and the different assumptions of the experimental scenario, methods of different complexities can be used. Some methods are simple to implement and run fast, but give good performances in only relatively simple environments (e.g. simple static background, few distracting objects in terms of colours or motion patterns or the user's action is under certain kinds of constraint), but are prone to fail in complicated situations. In the robotics research field, a trade-off between accuracy and speed should always be taken into account. In this section, two methods of hand detection/tracking are proposed. The first method is simple to implement but makes an assumption that the user's hand position is always in front of the torso. The second method combines three cues, i.e. colour, depth and motion, and tracks the hands using the Particle Filter method.

4.2.1 Hand Detection by Depth Information

In [173], two assumptions were made to simplify the detection of the hand, i.e. the hand was the closest object to a Time-of-Flight camera and the hand was at a distance from the main body. Under these assumptions, the hand could be easily located and extracted using a depth threshold. These assumptions are reasonable for hand posture recognition

applications, because people usually put the gesturing hand in front of their body when they intentionally want the robot to recognize the hand posture during communication.

The depth information can be easily obtained using the PMD range camera or the Kinect on our robot as illustrated in Chapter 2. The method in [173] is extended in this sub-section so that the hand extraction is more robust to noise.

Considering the noise in the depth data, the points which have the smallest depth values may be the result of noise instead of the gesturing hand. The smallest depth value $d_{\min}$ is found first and then the number of the points in the neighbourhood of $d_{\min}$ in the depth direction is counted by

$$N = \sum_i (d_{\min} \le dist_i \le d_{\min} + T) \tag{4.1}$$

where $dist_i$ is the depth value of the point i, and T is a specific threshold representing the thickness of the hand palm. If the number N is under a threshold, they are considered as noise (note as set S_1), and update the $d_{\min}$ in the complementary set

$$d_{\min} = \min_i (dist_i \mid i \notin S_1) \tag{4.2}$$

The loop between these two steps continues until N is beyond a specific threshold (i.e. the number of the points is large enough to be considered as a hand). In the meanwhile, the points on the hand are extracted and the central position of the hand is determined by averaging over these points.

Hand detection by this method is simple and fast. It can be adopted to locate the hand's position under the assumption that the hand is at a distance from the main body. However, this kind of method cannot perform a complete tracking of the hand continuously in each frame when the user is allowed to move the hands freely. Depth data alone are not sufficient to track the hand when the hand is moving close to, or behind other parts of the body, such as when the user frequently moves the hands back to the side of the body in the transition state between two dynamic actions. Another limitation is that, when the gestures involve two hands, both of the hands have to be at the same depth level; otherwise only the hand which is closer to the camera can be extracted.

The proposed method based on a 3D Particle Filter combining multiple cues achieves a more complete tracking of the hand with fewer constraints on the hand movements.

4.2.2 3D Particle Filter-based Hand Tracking

Basically, the Particle Filter represents the Probability Distribution Function (PDF) as a set of weighted samples. A particle can be considered as a hypothesis of the object's state. Each particle is assigned a weight that is determined by the observation probability, which measures how well the hypothesis fits the actual measurement. A new set of particles is generated by sampling from the current set based on the particles' weights. Particles with low weight may die off, while those with high weight are kept over frames. The Particle Filter is of a predict-update mechanism like the Kalman filter. However, the difference is that the Particle Filter maintains multiple hypotheses. A multiple hypotheses framework allows the tracker to handle clutters in the background, and recover more easily from failure or temporary distraction.

In the recursive procedures of the Particle Filter algorithm, old particles are selected based on their weights and then drifted and diffused by a dynamic motion model to generate new particles. Since there is no assumption about the possible motion pattern of the hands, in the implementation, the dynamic model to predict the positions of the new particles is chosen to be a Gaussian random process with the previous position as the mean, and the same variance in each direction.

In the case of hand tracking, the density distribution of the hand should be updated at each frame. The weights of the particles are determined by their observation probability. The proposed method calculates the weights of the particles using the combination of colour, motion and depth information to overcome the limitation of the single colour cue. The details are described in the following sections.

Skin Colour Cue

Since the chrominance components of human skin colours, even among difference races, take up only a very tight cluster in many colour spaces, the same skin colour distribution model and the projection method described in Section 3.2.1 in Chapter 3 is

used to evaluate the probability of a pixel having the colour of the skin tone. It is assumed that the hands and the face share the same colour distribution model.

Motion Cue

Image differencing is a widely used technique to find the motion region between successive frames. It is generated by pixel intensity differencing the current frame with the previous frame. However, it provides only rough motion regions without the accurate magnitude and direction of the motion.

Therefore, the Optical Flow algorithm [11] is used here to calculate the motion cue of the particles. For a pixel at location (x, y) with intensity $I(x, y)$ at time t in the images sequences, denote it as $I(x, y, t)$. It will move to the location $(x + dx, y + dy)$ at the time of $t + dt$. This gives

$$I(x+d_x, y+d_y, t+d_t) = I(x, y, t) \tag{4.3}$$

Assuming the time interval d_t and the movement are both small, expand $I(x+d_x, y+d_y, t+d_t)$ to be [27]

$$I(x+dx, y+dy, t+dt) \approx I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dx + \frac{\partial I}{\partial t} \tag{4.4}$$

This results in

$$I_x V_x + I_y V_y + I_t = 0 \tag{4.5}$$

where V_x, V_y are the velocity in the x, y directions, respectively (the optical flow at location (x, y)), and $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ and $I_t = \frac{\partial I}{\partial t}$ are the partial derivatives of the image at (x, y, t). The Lucas-Kanade method [177] for calculating the Optical Flow assumes that the velocity is constant in a local neighbourhood and solves the equations by the least squares criterion. In detail, the local image flow (V_x, V_y) satisfies [177]

$$\begin{aligned} &I_x(p_1)V_x + I_y(p_1)V_y + I_t(p_1) = 0 \\ &I_x(p_2)V_x + I_y(p_2)V_y + I_t(p_2) = 0 \\ &\vdots \\ &I_x(p_n)V_x + I_y(p_n)V_y + I_t(p_n) = 0 \end{aligned} \tag{4.6}$$

where $p1, p2 \ldots pn$ are the pixels in the neighbourhood. It can be written in a matrix form $Av = b$, where

$$A = \begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_n) & I_y(p_n) \end{bmatrix}, v = \begin{bmatrix} V_x \\ V_y \end{bmatrix} \text{ and } b = \begin{bmatrix} -I_t(p_1) \\ -I_t(p_2) \\ \vdots \\ -I_t(p_n) \end{bmatrix}$$

Thus,

$$v = (A^T A)^{-1} A^T b \tag{4.7}$$

Pyramidal implementation of the Lucas-Kanade method can be used for the optical flow estimation [27]. Image pyramid representation enables large pixel motion while keeping the size of the searching window relatively small.

However, because it is a purely local method, it often fails to provide accurate results in the interior of texture-insufficient parts. In addition, the matrix $G = A^T A = \sum_{p_x - w_x}^{p_x + w_x} \sum_{p_y - w_y}^{p_y + w_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$ in Equation (4.7) is required to be invertible, i.e. the minimum eigenvalue of *G* must be large enough. Thus, easy-to-track features which are defined as the pixels that have a minimum eigenvalue λ_m larger than a specific percentage of $\lambda_{\max}$ (the maximum value among λ_m) are selected. The available implementation of the Lucas-Kanade method in OpenCV is adopted [216].

Given the optical flow of easy-to-track features, each particle P_i is assigned the same values with the feature points F_j in its vicinity (Equation (4.8)).

$$v(P_i) = v(F_j), if \ \ j = \min_k d(P_i, F_k) \text{ and } d(P_i, F_j) < T \tag{4.8}$$

where $d(\cdot,\cdot)$ is the distance between two pixels and *T* is a threshold indicating the range of the vicinity.

If the largest value of the velocity among all particles is lower than a threshold, it indicates that there is not sufficient motion in the current frame, and so the positions of particles are therefore kept unchanged.

Depth Cue

The depth of the hand is not expected to have a sudden jump in successive frames, so higher weight will be assigned if the new particle and its predecessor have a small difference in depth. In addition, since the gesturing hand is often close to the camera, higher weight is given to the particles which have smaller depth values. The Probability Related Score (PRS) for the depth cue is calculated by Equation (4.9).

$$W_d = a_1 \cdot e^{-(d_{new} - d_{old})^2} + a_2 \cdot \frac{d_{\min}}{d_{new}} \tag{4.9}$$

where $d_{\min}$ is the minimum depth value among the all particles and a_1, a_2 are the weights for each component respectively.

Combination

Combining these three cues, i.e. colour, depth and motion, we calculate the weight (observation probability) for the new particles as

$$\begin{cases} W = 0 & if\ W_c = 0 \\ W = k_1 W_c + k_2 W_m + k_3 W_d & if\ W_c\ != 0 \end{cases} \tag{4.10}$$

where W_c, W_m, W_d are the PRSs of the particle by the colour cue, motion magnitude and depth cue, respectively. The three components are normalized individually before the combination. Higher weight is given to the colour cue but the depth and motion cues are also considered. This method enables the hand tracker to work in the presence of skin-coloured objects in the background, but will not be distracted by the moving non-skin-coloured objects in the foreground. The only situation in which the tracker could be distracted is when a large skin-coloured object moves in the foreground close to the camera. However, even if the tracking is lost occasionally, it can recover automatically, because of its multi-hypothesis nature and the random motion model.

4.2.2.1 Repulsive Force

When the two hands move close to each other, the trackers may lose tracking. We employ a repulsive force proposed by Hayashi et al. [100] between the two trackers (for the left and right hand). Every particle is assumed to have a round "cover area". The particles will have an overlapped area if they are too close. The repulsive force finds the

overlapped area and decreases the observation probability for the particles from different trackers to $W = W_0 \cdot [1 - R/(\pi r^2)]$, where W_0 is the original weight and R is the shared area calculated as [100]:

$$R = \frac{1}{2}\pi r^2 - \frac{d}{2}\sqrt{r^2 - \frac{d^2}{4}} - r^2 \sin^{-1}(\frac{d}{2r}) \tag{4.11}$$

where r is the radius of the area covered by a particle, and d is the distance between two particles.

4.2.2.2 Resampling

Resampling is used to avoid the problem of degeneracy of the algorithm, i.e. to avoid a situation where most of the importance weights are close to zero. Resampling re-calls the initialization step, in which all the particles are evenly scattered in the whole image, with the same weight for each particle.

In three situations, the resampling procedure is called. The first is when the maximum of the original (i.e. before-normalization) colour cues is under a threshold. It means that all the particles do not have a skin colour tone. The second is when the number of effective particles $N_{eff} = \frac{1}{\sum_k (w^k)^2}$ is too small, where w^k is the normalized weight of particle k. The third is when the tracker stays in a fixed position for a long time.

4.2.2.3 Experiment and results

Two hands have been successfully tracked simultaneously in our experiments, in the presence of some skin-coloured distracting objects in the background. Even more challenging situations, such as when the person wears short-sleeved clothes exposing the forearm, or when two hands move across each other, were tested as shown in Figure 4-1. The hands were correctly tracked most of the time. Even if the tracker lost the hand occasionally, it could recover within a few frames.

However, it is noticed that the fluorescent light above the user's head may have a negative influence, the strong light causing the skin to appear white. This effect can be

seen in the bottom picture of Figure 4-1. Some parts of the skin on the arm and hand appear white instead of the skin colour tone.

In addition to dynamic motion patterns, which will be discussed in the following section, two hand tracking results can also be applied to convey information about size and distance. For example, the user can say to the robot “Move to the left about this much”, when the two hands demonstrate the distance. The integration of gestures and speech utterances will be discussed in Chapter 6.

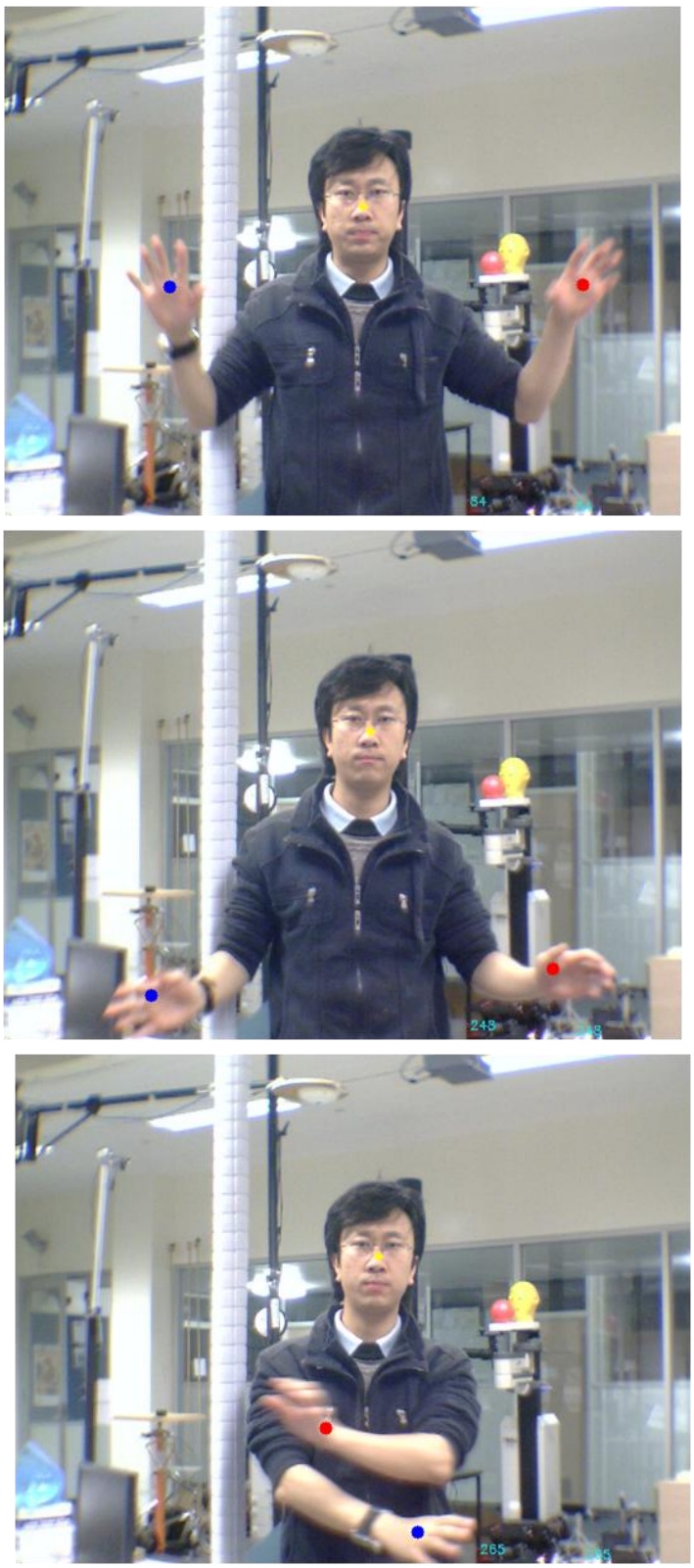


Figure 4-1: Examples of two hands tracking result with several skin-coloured objects in the background in a cluttered environment. The user rolled up his sleeves, making the tracking task more challenging.

4.3 Dynamic Hand Motion Pattern Recognition

In this section, the recognition method of the motion pattern which is conveyed by the trajectory of a moving hand is presented. The 3D hand trajectory can be captured by the hand tracking method described in Section 4.2.

Since there is normally an obvious pause between a new gesture and the previous one, this is taken as the sign of the starting point of a new gesture. The subsequent 3D hand position is relative to the starting position. Hand trajectories are recognized by the Finite State Machine (FSM) method. Each gesture is defined as a sequence of state transitions in the spatial-temporal space.

A state $\boldsymbol{s}$ is defined as a simple vector <$\boldsymbol{u,d}$>, where $\boldsymbol{u}$ is the centre of the 3D region, $\boldsymbol{d}$ is the distance threshold in the *x*, *y*, and *z* directions. The 3D space is divided into several cells representing different states. Note that each FSM recognizer has its own way of splitting the spatial space. A 3D position $\boldsymbol{p}(x,y,z)$ may be in the i^{th} state of one recognizer, while in the j^{th} state of another one.

When a new hand position $\boldsymbol{p}$ arrives, each FSM recognizer determines whether to stay at the current state or enters the next state based on the spatial parameters as in Equation (4.12).

$$s^k = i+1, if \begin{cases} | \boldsymbol{p} - \boldsymbol{u}_i^k | > \boldsymbol{d}_i^k \\ | \boldsymbol{p} - \boldsymbol{u}_{i+1}^k | < \boldsymbol{d}_{i+1}^k \end{cases} \tag{4.12}$$

where superscript k represents the k^{th} recognizer. $\boldsymbol{u}_i^k$ is the centre position of the state i of recognizer k, and d_i^k is the corresponding thresholds in three directions. The state of recognizer k, (s^k), is updated from i to $i+1$ if the conditions are met.

A gesture is recognized if a recognizer reaches its final state. As each FSM recognizer may have a different number of total states, this method allows simple movements to be represented by fewer states and complicated gestures by more states. If more than one recognizer reaches the final state, the one with the smallest error score by Function (4.13) is chosen.

$$e^k = \sum_j \sum_i || \boldsymbol{p}_{i,j} - \boldsymbol{u}_j^k || \tag{4.13}$$

where $\boldsymbol{p}_{i,j}$ represents the hand position *i* which belongs to state *j* of recognizer *k*.

This online recognition method determines a state transition for each recognizer whenever a new hand position arrives. It is different from the approaches that require complete gesture data before a recognition procedure begins.

Notice that if a gesture *A* is the prefix of another gesture *B*, recognizer *A* may report a recognized gesture while the hand is still in motion and the intended gesture is actually *B*. Therefore, a threshold of the time T_{thr} is defined in which a state transition must be completed. If no state transition happens during the time period of T_{thr}, the gesture is considered to be finished.

4.3.1 Experiments

In the experiments, five types of hand trajectories, including "wave hand", "draw a triangle", "pick up", "put down" and "come here", were modelled and tested. The recognition results were generated online and in real time.

Several key frames of the 'wave hand' movement from the video are shown in Figure 4-2. The subject first waved his hand from right to left and then from left to right. On the right side of each image, it shows the current states of each gesture recognizer according to the 3D hand position, and their maximum number of states. As only 'wave hand' went through all its states (the final state is indicated in red colour), this motion was recognized as 'wave hand'.

The overall recognition rate of the hand trajectories is 88%. Details for each gesture recognition results are shown in Table 4-1. Obviously, the accuracy of the hand position determines mostly whether the motion can be correctly recognized.

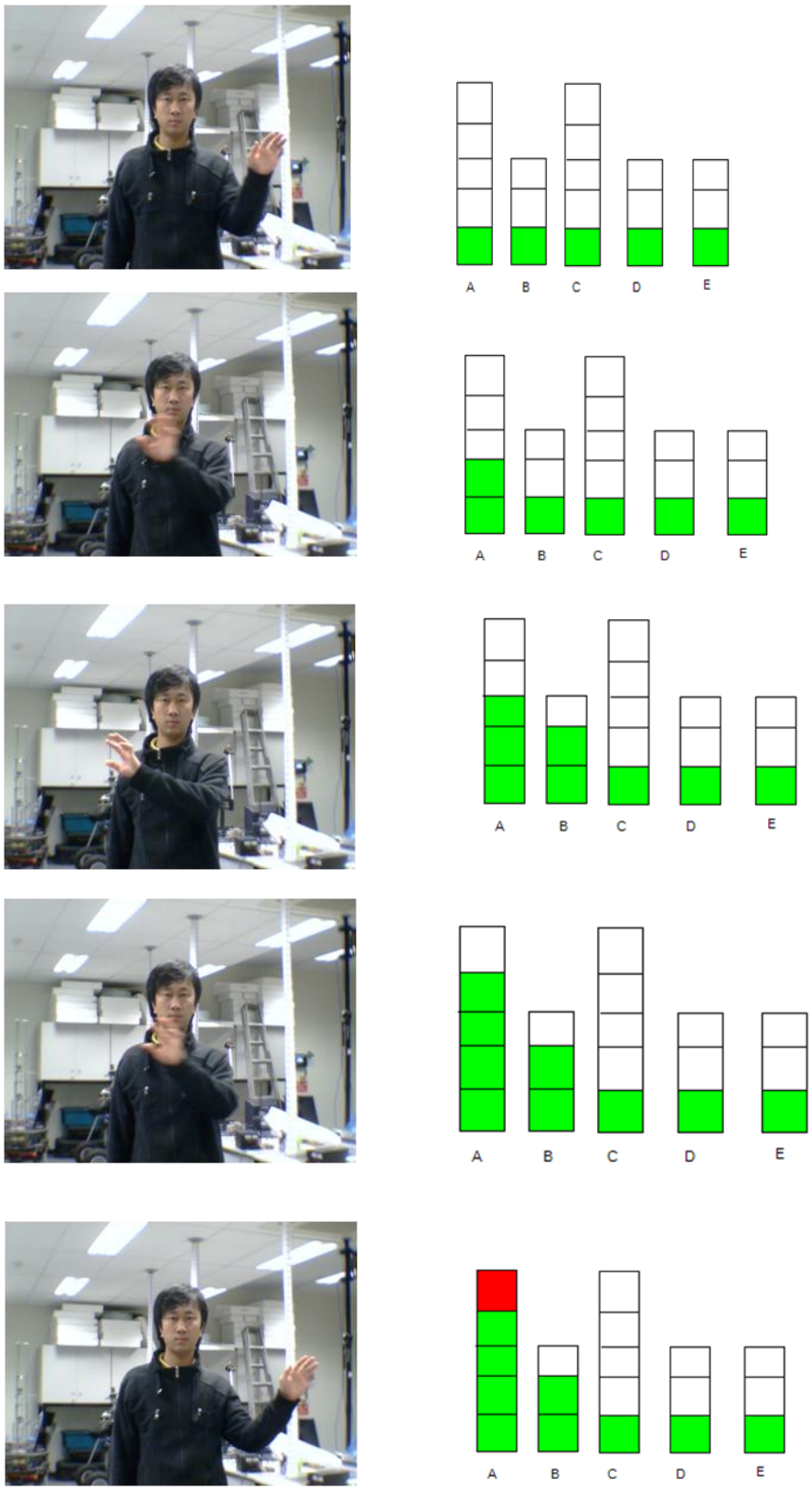


Figure 4-2: An example of the hand waving gesture and the state transition of each gesture recognizer. A: Wave hand; B: Draw triangle; C: Come here; D: Pick up; E: Put down

Table 4-1: Hand motion recognition results

Gestures	Performed times	Recognized times	Recognition Rate
Wave hand	10	9	90%
Draw triangle	10	8	80%
Come here	10	10	100%
Pick up	10	7	70%
Put down	10	10	100%

Using this method, simple and complicated movements can have different numbers of state transitions. It can detect the start and end point of a gesture automatically, and a gesture is not required to start at a particular 3D position. However, dramatic changes of the scale of the movement may affect the results.

4.4 Static Hand Posture Recognition

4.4.1 Hand-Forearm Segmentation

After the 3D position of the hand is found, the hand shape can be extracted by the points whose depth values are in the range between $[d_h-\Delta d_1, d_h+\Delta d_2]$, where d_h is the hand distance, and Δd_1 and Δd_2 are specified thresholds. Morphological operations, i.e. erode and dilate, are then applied to eliminate noise. However, this depth-based method may include part of the forearm when the hand and the forearm are at the same depth level.

Segmentation of the hand (including palm and fingers) from the forearm is important for accurate hand posture recognition. The first step is to determine whether the forearm is included in the extracted region. If the image contains the hand only, this procedure of hand-forearm segmentation is skipped. A simple way is to count the number of skin-coloured pixels. However, the size of the skin-coloured area is affected by the types of hand shape, the distance between the hand and the camera, and the illumination condition.

Therefore, the PCA method is used. PCA is a linear transformation that transforms the data to a new coordinate system such that the first coordinate (called the first principal component) indicates the axis which has the greatest variance of the projection of the

original data, the second greatest variance on the second coordinate, and so on. PCA is theoretically the optimal transformation for given data in least square terms [126, 127].

If the forearm is included, the shape of the hand-forearm region tends to be elongated. In this situation, the ratio between the first and second eigenvalues is large. In Figure 4-3, the left image shows an example of the connected hand-forearm region and the right image contains the hand only.

Figure 4-3: In the left image, an example of the segmented region contains the hand and part of the forearm. In the right image, an example of the segmented region contains the palm and fingers only

For the cases that include both hand and forearm, similar to the work in [75] and [125], a Gaussian Mixture Model (GMM) method is used to model the distribution of the points. The probability distribution of a point *x* associated with this model is

$$p(x) = \pi_1 \cdot p(x \mid hand) + \pi_2 \cdot p(x \mid forearm) = \pi_1 \cdot N(x; \mu_1, \Sigma_1) + \pi_2 \cdot N(x; \mu_2, \Sigma_2) \quad (4.14)$$

where π_1 and π_2 represent the prior probabilities of the hand and the forearm, respectively. Given the observed data and the number of mixtures, the maximum-likelihood estimation of the parameters π_1 μ_1 Σ_1 and π_2 μ_2 Σ_2 is found by means of the Expectation-Maximization (EM) algorithm [57].

EM is an iterative procedure and each iteration includes two steps. At the Expectation step, we find a probability $p(k \mid x_n)$ of sample x_n belonging to cluster *k* using the currently mixture parameters. At the Maximization step the parameters of each mixture are refined. More concretely, the implementation can be achieved as follows [125].

Algorithm: the implementation of the Expectation-Maximization (EM) algorithm

Expectation: given the estimated model parameters from the previous iteration, the posterior probability of point x_n is

$$p(k \mid x_n) = \frac{\pi_k \cdot N(x_n; \mu_k, \Sigma_k)}{\sum_j \pi_j \cdot N(x_n; \mu_j, \Sigma_j)} \tag{4.15}$$

Maximization: the parameters are re-estimated:

$$\mu_k = \frac{\sum_n x_n p(k \mid x_n)}{\sum_n p(k \mid x_n)} \tag{4.16}$$

$$\Sigma_k = \frac{\sum_n (x_n - \mu_k)(x_n - \mu_k)^T p(k \mid x_n)}{\sum_n p(k \mid x_n)} \tag{4.17}$$

$$\pi_k = \frac{\sum_n p(k \mid x_n)}{\sum_n \sum_k p(k \mid x_n)} \tag{4.18}$$

where k refers to the hand (k=0) or the forearm (k=1). The parameters are refined by iterating between the Expectation and Maximization steps until they converge.

Initialization: fit all the points to a single Gaussian distribution,

$$\mu = \frac{\sum_n x_n}{M} \tag{4.19}$$

$$\Sigma = \frac{\sum_n (x_n - \mu)(x_n - \mu)^T}{M} \tag{4.20}$$

where M is the number of the points in total, and use the PCA result (the eigenvectors v_1 and v_2) to initialize the parameters as:

$$\mu_1 = \mu + v_1, \mu_2 = \mu - v_1, \Sigma_1 = \Sigma_2 = \Sigma \tag{4.21}$$

The EM method may converge to an incorrect local maximum and gives different result according to the initial values of the mean and covariance. The implementation starts from the initialization step which considers the whole point cloud as a single Gaussian

distribution and then determines the mean values of the two Gaussian distribution μ_1 and μ_2 by Equation (4.21), while the initial guess of covariance is unchanged. Then the Expectation and Maximization steps are executed iteratively until a maximum iteration number is reached or convergence is achieved whichever comes first.

Figure 4-4 shows two examples of the segmentation results of the EM algorithm. The palm and finger region is marked in white and the forearm segment is marked in grey.

Figure 4-4: Two examples of the hand-forearm segmentation

4.4.2 Orientation Rectification

The orientation of the hand can be meaningful. For example, in Figure 4-5, three samples of the same hand shape with different orientations may have different meanings. However, small variations of the orientation should be rectified. We calculate the orientation of the hand shape by the PCA method. The eigenvector associated with the largest eigenvalue indicates the direction of the shape. We take $\pm 45°$ as the delimitations. The hand shapes are rotated to vertical or horizontal orientation. For example, in Figure 4-6, the hand shape in the left image is slightly slanted. It is rotated to the vertical (shown in the right image) since the angle between its first eigenvector and the x axis is larger than $45°$.

Figure 4-5: Hand gestures with different orientations may have different meanings: (a.) thumb up (means 'good'), (b.) thumb down (means 'bad') and (c.) go to right

Extracted hand shape, slightly slanted

Rotated result of left image

Figure 4-6: An example of the hand shape orientation rectification

4.4.3 Distance Transformation and Matching

Extracted images that contain the hand palm and finger region are rescaled to the same size (96×128), and the contours are then found. (See Figure 4-7 (a) and (b)).

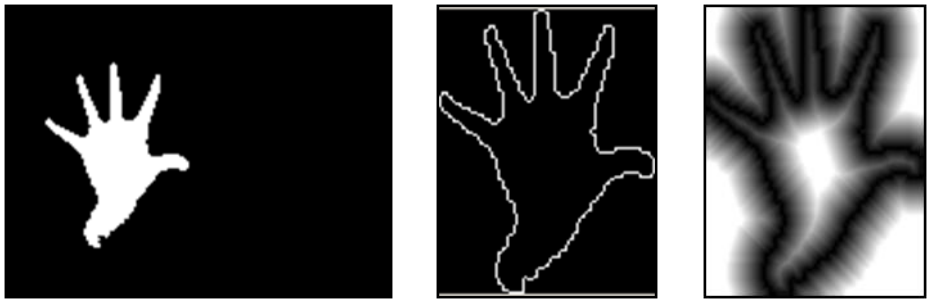
For hand shape recognition, a database needs to be established. A set of images containing various hand shape patterns is recorded with known labels in the training stage. To match a query hand image against our database, the distance transformation and matching method [26] is employed.

The true Euclidean distance is resource-demanding and usually not necessary, as the edge points are often influenced by noise. Distance Transformation (DT) [25] is a reasonable approximation of the Euclidean distance. Each non-edge pixel is assigned a value that is a measure of the distance to the nearest edge pixel. In a binary edge image, edge pixels are set to zero and non-edge pixels are initially set to infinity. The value at the position (*i,j*) is updated iteratively. At iteration step *k*, it is computed by Equation (4.22) [26].

$$v_{i,j}^{k} = \min(v_{i-1,j-1}^{k-1} + 4, v_{i-1,j}^{k-1} + 3, v_{i-1,j+1}^{k-1} + 4, v_{i,j-1}^{k-1} + 3, v_{i,j}^{k-1}, \\ v_{i,j+1}^{k-1} + 3, v_{i+1,j-1}^{k-1} + 4, v_{i+1,j}^{k-1} + 3, v_{i+1,j+1}^{k-1} + 4) \tag{4.22}$$

The iterations continue until there are no value changes. Figure 4-7 shows an example of distance transformation. The hand is first segmented out from the background (left image). The edges are found and resized (middle image), then the Distance Transformation is calculated (right image), where its intensity is rescaled for illustration purposes). It can be seen that the further the pixel is away from the edges, the larger its

value is. Distance Transformation of the hand shape template images are calculated in advance before the query stage.



Extracted hand shape | Resized contour | Distance transformation

Figure 4-7: An example of Distance Transformation of a hand silhouette image

At the testing stage, a matching procedure is needed to measure the similarity of the query image and the templates in the database. The matching score of a binary edge image $I(i,j)$ and a Distance Transformation $DT(i,j)$ is calculated using Equation (4.23).

$$s = \sqrt{\frac{1}{n}\sum_{i,j}(I(i,j)\cdot DT(i,j))^2} \tag{4.23}$$

Borgefors [26] compared four different criteria for the matching measure: median, arithmetic, root mean square (r.m.s) and maximum. The r.m.s was found to give fewer false minima than the others.

In order to measure the similarity between two images, for example,a query image $\boldsymbol{A}$ and a template hand image $\boldsymbol{B}$ **(**the Distance Transformation of the template, $\boldsymbol{DT_B}$, has already been obtained in advance), the bilateral matching method is used as in Equation (4.24).

$$s = \sqrt{\frac{1}{n}\sum_{i,j}(I_A(i,j)\cdot DT_B(i,j))^2} + \sqrt{\frac{1}{n}\sum_{i,j}(I_B(i,j)\cdot DT_A(i,j))^2} \tag{4.24}$$

First, the query image is resized and extracted to obtain a binary edge image $\boldsymbol{I_A}$, then a matching score is calculated by equation(4.23); second, the Distance Transformation of the resized query image $\boldsymbol{DT_A}$ is derived, then another matching score is calculated by Equation (4.23). These two scores are added together as the final similarity measurement between the query image $\boldsymbol{A}$ and the template $\boldsymbol{B}$. A smaller matching score indicates a

higher similarity. The similarity measurements between the query and all the templates in the database are calculated and the smallest score indicates the most likely hand posture.

In a human-robot interaction scenario, a communicative or manipulative hand posture gesture is considered to be deliberately enacted. Therefore, the user always uses a hand posture to convey the intention explicitly. A stability counter is used to exclude the unstable recognition results when the hand is in motion or in the transition state between different postures. Only stable gestures are considered to be a valid output.

4.4.4 Experiments

The hand shape templates were captured and stored in advance. It is worth emphasizing that individuals can define their own sets of hand gestures. The robot could first recognize the user using the identification method described in Chapter 3, and interpret the meanings of the gestures accordingly for each individual. However, a group of generic gestures are available for un-recognized users. The self-defined set for each individual always includes the generic gestures as a subset, excluding the ones which are already explicitly defined by that user. The hand shapes used by the author in the experiments are shown in Figure 4-8.

Both left and right hands could be used for gesturing, but to save space, only left hand shapes are shown here. Numbers, directions and the stop sign are considered as they are commonly used in human-robot interaction applications. Two kinds of templates were built for some gestures, including ‘turn right’, ‘turn left’ and ‘stop’.

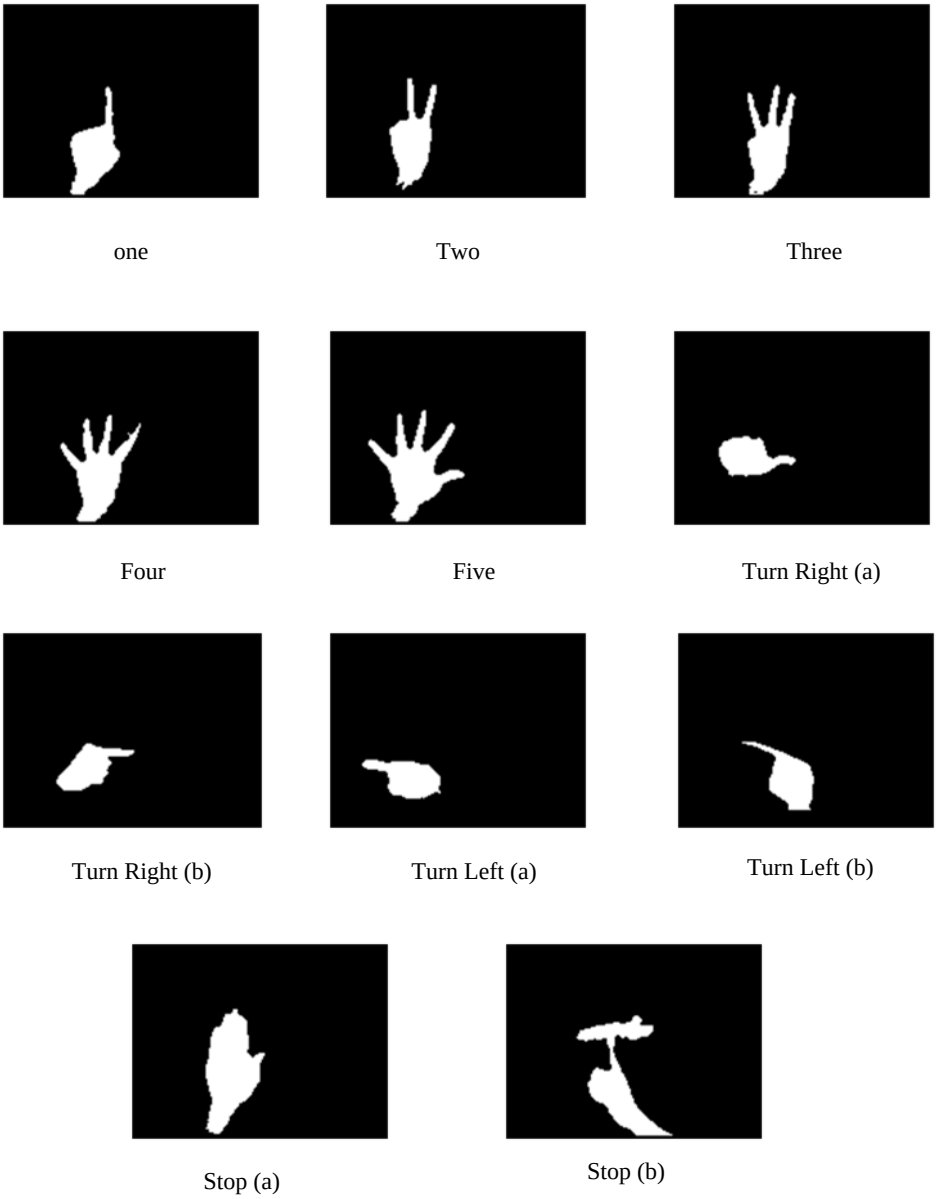


Figure 4-8 Hand shape templates

Depending on the viewpoint of the robot and the orientation of the hand, the hand may be self-occluded. For evaluation purposes, we excluded the hand images which cannot be recognized even by a human. A stability counter ensures that the system only outputs stable hand shape recognition results. The values are empirically determined, depending on the frame rate, quality of the depth data and the tremor of the hand. If the value is too large, there will be more mis-detections; on the other hand, if the value is too small, there will be more false positives. Currently, in our experiment, four continuous recognitions of the same result are considered to be stable. As the recognition results were generated online and the stability criterion of four frames is short, the system is considered to run in real time.

The detailed recognition confusion matrix is shown in Table 4-2. The overall recognition error rate is under 3%.

Table 4-2: The confusion matrix of the hand shape recognition results

		Recognized Results											
		One	two	three	four	five	Turn right (a)	Turn right (b)	Turn left (a)	Turn left (b)	Stop (a)	Stop (b)	total
Conducted hand postures	One	82	0	1	0	2	2	0	0	0	0	0	87
	two		76	3		2							81
	three		3	64	3								70
	four				57	3							60
	five					59							59
	Turn right (a)			3	1	1	45						50
	Turn right (b)							51					51
	Turn left (a)								47				47
	Turn left (b)								1	59			60
	Stop (a)						2				68		70
	Stop (b)											70	70

4.5 Conclusion and Discussion

Using the proposed methods the hands were robustly tracked, and both dynamic hand motion patterns and static hand postures were successfully recognized in the experiments. The hand gesture examples in this chapter are still within a small vocabulary;

however, the set of the gestures can be easily enlarged. Furthermore, the user can define his/her unique set of customised gestures.

The main limitation of the current hand shape recognition method is that it is not viewpoint independent. The gesturing hands need to directly face the camera. This is because both the depth sensor and the colour camera are mounted on the robot. Hence, the commonly used multi-view techniques cannot be applied. Therefore, the hand could be self-occluded when it is observed from a non-optimal viewpoint. In this case, the system failed to recognize the hand postures.

One limitation of the hand motion recognition method is that dramatic changes of amplitude of the motion could cause failures. This problem could easily be solved if the length of the hand trajectories is rescaled to the same size. However, a sophisticated approach will then be required to segment the meaningful gestures out of the continuous hand movement.

Gestures alone can comprise a large vocabulary to express one's intentions, such as sign language. However, in a natural HRI scenario, it would be more attractive when the visual recognition of hand gestures is combined with a speech recognition system.

5

Visual Interpretation of Natural Pointing Gestures

During interpersonal communication, deictic gestures are often adopted to convey spatial and directional information. Humans use pointing gestures in communication as early as when they are infants. This phenomenon is widely known in developmental psychology as joint attention [199].

In daily life, pointing gestures are often integrated with speech utterances. Pointing gestures make expressions requesting objects much simpler. Describing the position of an object can be complicated. Instead, a pointing gesture simply and explicitly indicates the location of the target. For example, rather than saying "give me that cup on the right side of the book", a person can simply say "give me *that* cup", accompanied with a pointing action towards the intended cup. The advantage is conspicuous in a human-robot interaction scenario when there are multiple similar objects in the scene. A detailed discussion of multimodal interaction will be provided in Chapter 6, while this chapter focuses on the interpretation of pointing gestures.

Like other gestures discussed in previous chapters, deictic gestures can also work alone. In [249], the robot guessed what a user pointed at and why by examining the pointing behaviour with other contextual information, then decided its action. For example, if the user wanted to guide the robot to a certain place, he simply pointed to the robot and then to the destination; if the user wanted the robot to move an object, he simply pointed to the object and the desired location. This kind of self-decision mechanism is essential in the situation where the environment is too noisy or a speech recognition module is not available on the robot.

This chapter describes the detection of the occurrence of pointing gestures and the calculation of two pointing measures, i.e. the Eye/Face-Hand line and the forearm direction. An object selection method is proposed, considering individual preferences associated with objects locations' and the quality of the pointing vectors. Part of the work in this chapter was published in [166].

5.1 Related Work

A pointing hand can be used as a virtual laser pointer on a large scale display during a slideshow presentation, as demonstrated in [94]. In a human-robot interaction scenario, researchers have also proposed experiments to use pointing gestures for purposes including object selection [145, 212, 224, 271] , robot path control [248], parking guidance [248] and so on. Various camera configurations and different methods have been adopted to estimate the pointing direction.

Pointing directions can be estimated even with a traditional monocular camera. Cernekova [44] compared two different camera placement arrangements in their experiments where a user pointed to a big screen. In the first setup, the camera was located on the right side of the user, and in the second, the camera was placed on top of the screen. Their experiments showed the frontal camera setup provided better results. In both cases, the screen size was 1.2 by 1.2 meters, divided to 6×6 cells. The fingertip and the shoulder were extracted to generate a feature vector, which was fed to trained multi-class SVMs in order to find which cell was pointed at.

Richarz et al. [240, 241] utilized a single fish-eye camera on their low-cost prototype of a home robot system. They first used a combination of head-shoulder detection, empiric factors and a distance measurement to form a multimodal person tracker to find the Region of Interest (ROI) in the images. A left/right classifier first determined whether the person was pointing to the left or right. Then the face and pointing arm region were extracted. A cascade of Multi Layer Perceptron (MLP) classifiers were used to estimate the pointing direction. However, their spatial resolution was low: the spot pointed to on the floor was only classified to three coarse radius levels and the estimation of angles was prone to errors.

To allow the users to move freely inside a wider environment while pointing at some targets at various locations, Guan [84] used multiple un-calibrated cameras, but only one camera was employed to determine the pointing target based on the selected best view.

Face skin information is explicitly adopted as the measure of view quality for selecting the optimal viewpoint.

Because the depth information is not available when using a traditional monocular camera, the actual 3D pointing vectors cannot be calculated. The above monocular camera methods often use classifiers such as SVMs and MLP, to train the relationships between detected features and targets or discrete pointing vectors; as a result, the spatial resolution is relatively low. To obtain the explicit representation of pointing vectors, stereo cameras have often been employed.

Yamamoto et al. [292] used four pairs of surrounding stereo cameras in the corners of a ceiling. In this arrangement, they could capture the entire body and face simultaneously. Their system integrated 3D information from multiple stereo cameras and used a crossing hierarchical method to form a 2D image sequence projected in parallel planes. The system then segmented this 2D image sequence and obtains the position, orientation and length parameters of the body and the pointing arm. However, the subjects are required to hold the pointing posture for 15 seconds, which limits the applicability of the system.

Compared to the ubiquitous stereo-vision systems, a single pair of stereo cameras is a more popular configuration. Given a dense disparity map, Jojicy et al. [125] subtracted the background, and broke the 3D points on the body into two parts using a hybrid uniform-Gaussian mixture model. They extracted extreme points as the head and the finger. These two points then defined the pointing vector.

Nickel et al. [212] integrated the colour and disparity information from a calibrated stereo camera to track the body parts using a multi-hypotheses tracking framework. At each frame, an n-best list of hypotheses (i.e. head and hand position) is kept, in which each hypothesis is connected to its predecessor in a tree structure. The path with maximum overall probability is chosen. Similarly, the head-hand line is taken as the pointing direction.

A system was constructed in [249] which used pointing gestures to tele-operate networked robots. A screen showed a virtual room interface which was a graphical representation of the real scene. Three cameras, mounted on the screen, tracked the user's hand motion. The user pointed at a spot on the screen so that the robot, which was

connected with a wireless network, was guided to the corresponding location in the real room. Their robot decided its own action according to the recognized objects or the pointing direction. The virtual interface may lower the requirement of the accuracy for the pointing direction, because the positional relationships are more important than the actual location in their application, and the virtual interface can be zoomed in and out to facilitate the recognition of the pointing direction. In contrast, we are more interested in the direct face-to-face interaction between the user and the robot, which is more natural and effective in the assistive robotics scenario.

Most researchers consider the line-of-sight, which connects from the centre of the eyes to the fingertip as the pointing vector. It is a fine estimation of the actual pointing direction when the person is pointing with his arm outstretched. However, in real life people may also point with a non-outstretched arm and the actual pointing direction aligns with the direction of the forearm [223, 224].

In order to achieve mutual ways of interaction between human and robot, Sugiyama et al. presented a three-layer model for generation and recognition of attention-drawing behaviour. The three layers include the pointing space model, the reference term model and the object property model. The robot recognized the user's pointing gestures and also performed similar pointing behaviours to confirm its interpretation with the user. However, they captured 3D positions of body joints using markers attached to the user.

After pointing gestures are spotted and the pointing direction is determined, the robot should be able to extract the referred-to object from the background and locate its 3D coordinate in the real world. Then object recognition and learning can be carried out to make the application more interesting. In [145], Kim et al. segmented an unknown object from 3D point clouds in the region of interest, which was defined by the pointing direction. Schauerte et al. [251] utilized a bottom-up saliency map generated by the locations of objects and a top-down saliency map defined by pointing directions to identify pointed-at objects. Their pan-tilt-zoom camera could be steered towards the intended object and a Scale-Invariant Feature Transform (SIFT)-based recognition or learning (if no matched object was found) was implemented. If the object could not be identified due to distracting objects in close proximity, an iterative shift of attention would be applied, sequentially focused on different objects.

In our assistive household robot project, pointing gesture recognition can be followed by physical object manipulation of the intended object. Objects are extracted by a 3D Shape Recovery method [61] in the region indicated by the pointing direction. The collaborative work on object manipulation is detailed in chapter 6.

5.2 Pointing Direction Calculation

Two main pointing methods, the Eye/Face-Hand line (also referred to as line-of-sight) and the forearm direction are investigated in this section.

As mentioned in [125, 212], people have different preferences in relation to pointing methods. In addition, as observed in our experiments, the method chosen by the subject depends not only on the preference of the person, but also the locations of the targets. As shown in Figure 5-1 and Figure 5-2, the subject chose the Eye/Face-Hand line to point at the spot on the floor while he used his forearm to point at a teapot on the table.

Figure 5-1: The person is pointing at a spot on the floor using the Eye/Face-Hand line

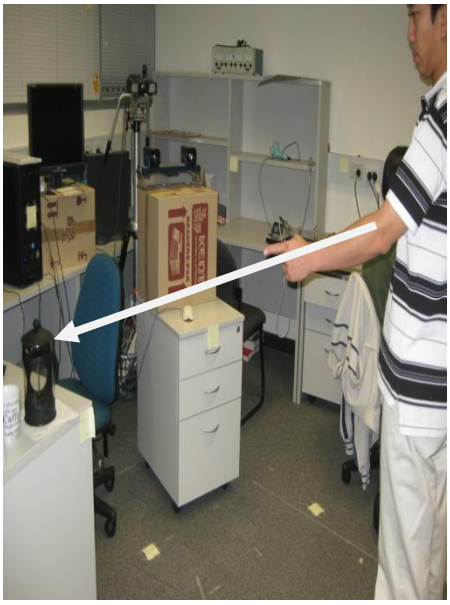


Figure 5-2: The person is pointing at a teapot on the table using the forearm direction

However, not all of the pointing methods can always be calculated, depending on the relative position and orientation between the subject and the robot, and the quality of the data captured from the cameras. Only stable and meaningful vectors are considered as the deliberately aimed pointing direction.

This section describes the methods of calculating two different pointing directions, and discusses their availability. Experiments were carried out to test the accuracy of the pointing direction and verify our above-mentioned hypotheses about the choice of pointing methods. Finally, an object selection method in a probabilistic framework is proposed considering individual preferences associated with the locations of objects and the quality of the pointing vectors.

5.2.1 Eye/Face-Hand Line

The line connecting the central position of the eyes to the pointing finger represents the line-of-sight. However, it is difficult to locate the pointing fingertip in 3D space in a robust and reliable way. The hand position (described in Chapter 4) is used instead as a reasonable approximation of the fingertip position. To find out the eyes' positions, a close-up view of the face is normally required. A commercial stereo camera system, "faceLAB" (see Section 6.2.2), is employed for this purpose. It can track the features on the face and around/in the eyes to locate the eyes. However, this device has a narrow field of view and a shallow depth of field, which requires the person to remain within a tolerance region

without too much body movement. To reduce the constraints on the user's movement, the position of the centre of the eyes is approximated by the centre of the face when the faceLAB loses tracking of the person's eyes (the face tracking method is described in Chapter 3). Therefore, the Eye-Finger line is approximated to Face-Hand line in many cases.

The Eye/Face-Hand line is always available as long as the user remains in the view of the camera, and the face and the pointing hand are being tracked. As stated in Chapter 3, even if the subject turns his head towards the intended object before and/or during the pointing action, the centre of the face is still being estimated.

However, the line-of-sight makes a good estimation of the pointing direction only when the subject stretches his/her arm out. Schauerte et al. [251] explicitly imposed this requirement on the participants in their experiments and Park et al. [224] also discussed this in their work .

It is observed in our experiments that, when a person points to a target low on the floor, up on the ceiling or at a relatively far distance, he/she tends to use this pointing method, probably because the person tries to aim at the object using the Eye-Finger line. When the person "sees" the line having an intersection with the target, he/she has a high confidence of pointing correctly.

5.2.2 Forearm Direction

A pointing gesture using the forearm direction is called a "small pointing" gesture in [224]. In our implementation, the forearm is assumed to be a rigid elongated object, and the 3D points on the forearm are used to calculate a vector which can represent the main trend of the distribution of the point cloud.

The 3D points corresponding to the pixels on the depth image can be calculated by equation (2.2) in chapter 2. Therefore, given the position of the pointing hand, the points on the forearm can be extracted by a depth threshold. If the person is pointing to the front, the forearm is behind the hand and before the face. However, if the person is pointing to the side, the depth difference between the hand and the face is small, and a threshold on the side is also needed to prevent it from including points on the torso. Hence, a 3D point P belongs to the forearm, if

$$\begin{cases} |P.z - P_{hand}.z| < \max(T_z, 0.2) \\ |P.x - P_{hand}.x| < T_x \\ |P.y - P_{hand}.y| < T_y \end{cases} \quad (5.1)$$

where, P_{hand} is the position of the pointing hand, coordinate z represents the depth direction. T_x, T_y and T_z are specific threshold values. The threshold T_z is set to be

$$T_z = 0.6 \cdot |P_{face}.z - P_{hand}.z| \quad (5.2)$$

where 0.6 is an empirical value representing the length proportion of the forearm in the face-hand distance in the depth direction.

Since the depth information on the edges contains much noise, the points on the edges of the arm are removed using the method introduced in Section 2.4 in Chapter 2 before the subsequent processing.

The forearm direction is calculated using the PCA method. First we calculate the covariance matrix $\boldsymbol{C}$ of the 3D coordinates of the segmented points on the forearm. Then the eigenvectors and eigenvalues are calculated by the Singular Value Decomposition (SVD) method [7]. The eigenvector $\boldsymbol{e_1}$ with the largest eigenvalue denotes the direction of the largest variance of the data set. Although the full arm is an articulated object, the forearm is considered as a rigid elongated object, and $\boldsymbol{e_1}$ is expected to be the measurement of the forearm orientation. Figure 5-3 is an example showing the principlal axis (the red line) of a set of 3D points (the blue dots) calculated by the PCA method.

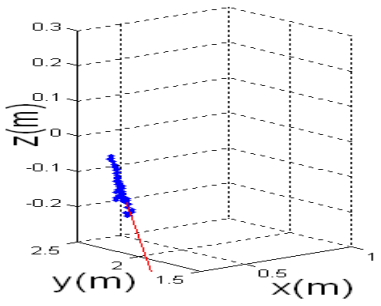


Figure 5-3: An example of the calculated first principal component of a set of 3D points

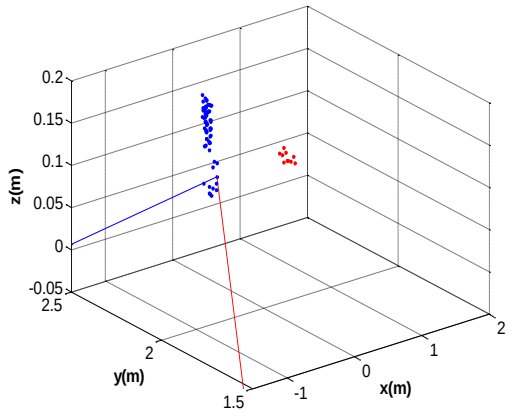


Figure 5-4: An example that shows the PCA result (the blue line) is not a good estimation of the main trend of the 3D points under the influence of outliers. Iteratively employing PCA under the RANSAC framework, it overcomes the influence of outliers and provides the correct estimation (the red line)

However, the PCA method is sensitive to outliers. It only has a good performance when the observed data contain no outliers. As shown in Figure 5-4, in the presence of outliers (shown as the red dots), the blue line represents the PCA result of the whole data set, which gives an obviously wrong estimation far from our expectation. Therefore, the Random Sample Consensus (RANSAC) method is used to iteratively calculate the PCA results of several subsets of the 3D-point cloud.

RANSAC is an iterative method to estimate the parameters of a mathematical model from a set of sample data which may contain outliers. The RANSAC method is used to iteratively evaluate the models, i.e. the principal axes of the forearm points sets calculated by the PCA analysis method. We randomly choose a certain number of the points (sample set $\boldsymbol{S_1}$) to estimate the pointing vector (model $\boldsymbol{M}$) using the PCA method; then all points not in $\boldsymbol{S_1}$ are evaluated whether they are inliers (fit the model) or outliers (do not fit the model) individually. Here 'fit' means the 3D distance from the point to the calculated pointing line is smaller than a threshold. After a specific number of iterations, the model which has the most inliers is the output of the RANSAC procedure. The red line in Figure 5-4 shows the result on the same set of 3D points processed using this method. It shows that this approach provides reliable and robust results even in the presence of outliers.

However, the direction of the eigenvector $\boldsymbol{e_1}$ generated by the method may need to be negated. The proper direction of $\boldsymbol{e_1}$ can be inferred based on the face and the hand positions: if $\boldsymbol{V}_{\text{face-hand}} \cdot e_1 < 0$, where $\boldsymbol{V}_{\text{face-hand}}$ is the vector from the face to the hand

(i.e. $\boldsymbol{e_1}$is pointing in the direction from the hand to the face), then the direction of $\boldsymbol{e_1}$needs to be altered.

Moreover, the eigenvalues associated with the eigenvectors are used to eliminate an invalid dataset. Since the forearm is assumed to be a rigid elongated object, we exclude the cases in which the ratio of the first and second eigenvalues is under a specific threshold, which means there is no distinct trend of the point cloud.

5.3 Pointing Gestures Occurrence

Inspired by [125] and [251], a simple approach is proposed for the detection of the occurrence of pointing gestures. It is based on the notion of a meaningful gesture, which is a stable gesture that is deliberately aimed. For a pointing gesture, these requirements are defined and implemented as follows.

Meaningful: A pointing gesture is aimed if the angle between the vector of the pointing direction and the user's body is larger than a threshold ($30°$ in our implementation). This distinguishes a pointing gesture from a situation where the person is standing with his/her arms casually downwards. It is worth noticing that one drawback of this method is that it cannot detect the pointing gesture when the person intentionally points to a spot close to his/her feet.

Stable: A gesture is considered as stable when it has been held in a direction over a certain length of duration (at least 1 second in our implementation). This corresponds to the hold phase in the three phases ("Begin-Hold-End") of a complete pointing action in the work of [212]. Hold phase is the most distinct state of the pointing gesture distinguishing from other gestures. To account for small tremors in a user's hand or arm, vectors are first filtered by a Median Filter and then the ones with small angular differences are grouped. A mean vector is calculated by averaging all the vectors in the group. For each pointing method (i.e. the Eye/Face-Hand line and the forearm direction), a new vector could be generated in each frame. If the angular difference between this vector and the mean vector of this group is less than a threshold (currently set to $3°$), the new vector adds into this group and the mean vector is updated. In this way, a complete pointing action will, in theory, generate a single pointing vector. The quality of the generated estimation is measured by its duration in time and the variance of the within-group vectors by Equation (5.5).

Another reason why we can simplify the detection of the occurrence of the pointing gestures is because a speech recognition module is available on our robot which can also be used to indicate the occurrence of a pointing gesture. For example, in speech utterances like "give me that book" and "take it over there", "that" and "there" are intuitively connected with deictic gestures. The temporal alignment of speech and pointing gesture is described in Chapter 6.

5.4 Object Selection

As noticed in our experiments, the choice of the pointing method by each individual depends on his/her personal preference and the location of the target. Some people like to use the Eye/Face-Hand line method, while others tend to use forearm direction more often. Some locations are easy to be pointed at by the Eye/Face-Hand line, while others by the forearm direction.

The general individual preference for each method can be calculated in a statistical way using Equation(5.3):

$$\overline{p_u}(m) = \frac{C_m}{\sum_m C_m} \tag{5.3}$$

where C_m is the number of correct object selections achieved using the method m by a specific user u. Here m refers to the Eye/Face-Hand line or the forearm direction.

The object location factor may also affect the choice of pointing method. Each object has two attributes with regard to its location, i.e. their distance to the user and their height above the floor. The Probability Related Score (PRS) that the user chooses the method m to point at an object, whose height property is j and distance property is k, is calculated by

$$p_u(m; j, k) = \frac{C_{m,j,k}}{\sum_m C_{m,j,k}} \tag{5.4}$$

where $C_{m,j,k}$ is the number of correct object selections achieved using method m, to the object with properties of j and k; j refers to the height level (high, medium, low), and k refers to the distance level (far, middle, near). $p(m; j, k)$ indicates that it is a function of m, parameterized by j, k.

We calculate the quality (q) of the estimated pointing vector as a function of its duration (d) and within-group variance (v).

$$q = w_1 \cdot \min(\frac{d}{MAX_D}, 1) + w_2 \cdot \min(\frac{MIN_V}{v}, 1) \tag{5.5}$$

where MAX_D is a specific maximum duration value, MIN_V is a specific minimum variance value, the weights w_1 and w_2 are set as 0.8 and 0.2 respectively. The duration is given higher weight, because the duration time varies in a great deal, while the with-group variance is normally small according to how the vectors are grouped, as described in Section 5.3.

Considering the above factors, the PRS that an object is pointed at is calculated by Equation (5.6)

$$p(obj_i) = \sum_m p(m) \cdot p(obj_i \mid m) \tag{5.6}$$

where $p(m)$ is the PRS that the pointing method m is used. It is defined by the product of Equations (5.4) and (5.5): including the factors of the individual preference associated with object locations and the quality of the pointing vector.

$$p(m) = q \cdot p(m; j, k) \tag{5.7}$$

$p(obj_i \mid m)$ is the PRS that the object i is targeted under the condition that the pointing method m is chosen. It is a Gaussian distribution function with the angle between the hand-object line and the pointing vector as the variable x.

$$p(obj_i \mid m) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{5.8}$$

Note that $p(Obj_i)$ in Equation (5.6) is not yet normalized. If a unimodal pointing gesture is used for the object selection task, the object with the largest PRS $p(obj_i)$ is considered as selected.

Figure 5-5: An example of a pointing gesture in an office environment

Figure 5-6: A snapshot of the experiment setup. The potential targets are marked by red circles. They are placed at various locations, at different height and distance levels. Some of the objects are not shown in this picture due to the limited view of the camera.

5.5 Experiments

We carried out our pointing experiments first using the PMD camera set and later using the Kinect, both of which were in an indoor office environment, as shown in Figure 5-5.

Eight volunteers participated in our experiments with the PMD camera set and three volunteers participated in similar experiments with Kinect a few months after the device was released. Most of the subjects were not familiar with our work, and they were asked to conduct pointing actions in their natural way. They were free to move around as long as they remained in the view of the cameras on the robot. They could choose any order, in which the objects would be pointed at, and they could also point to any object multiple times. The author was standing behind the subjects by a short distance to take notes of which object was actually pointed to. During the experiments, there was no communication between the author and the subjects to ensure that their behaviours were natural and not influenced externally.

Nine objects were employed with the PMD camera set, and 14 objects with the Kinect. The employed objects are commonly used in our daily life, including a biscuit box, apples, a banana, cups (green, red, white and transparent), a bunch of grapes, plates of different sizes and books. The locations of the objects were designed to show variations in height and distance in 3D space, and Figure 5-6 is a snapshot of the objects in the experiments. The potential targets are marked with red circles. The locations included on the floor, near the ceiling, on the computer desk, on a small table, the vicinity of the subject and as far as 4m away. The locations and the sizes of the objects were manually measured before the experiments. Figure 5-7 and Figure 5-8 show the objects locations in the experiment with the PMD camera set and with Kinect, respectively.

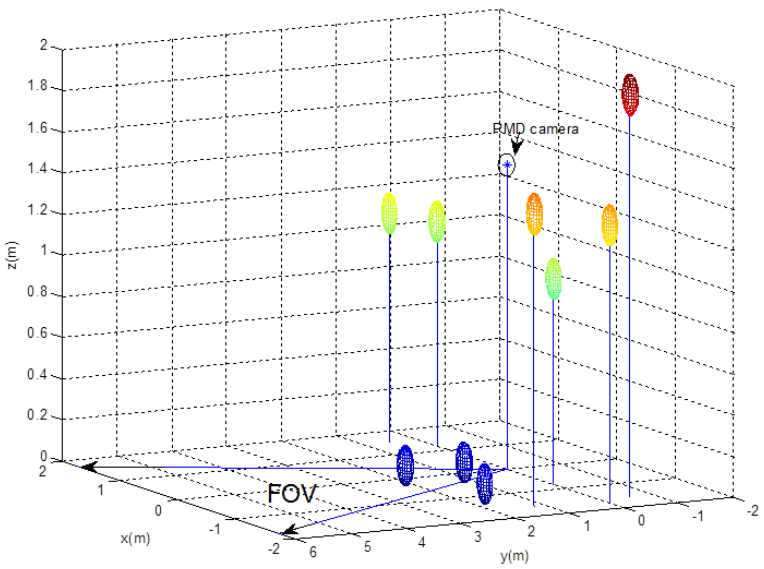


Figure 5-7: Objects locations in the experiments with the PMD camera set

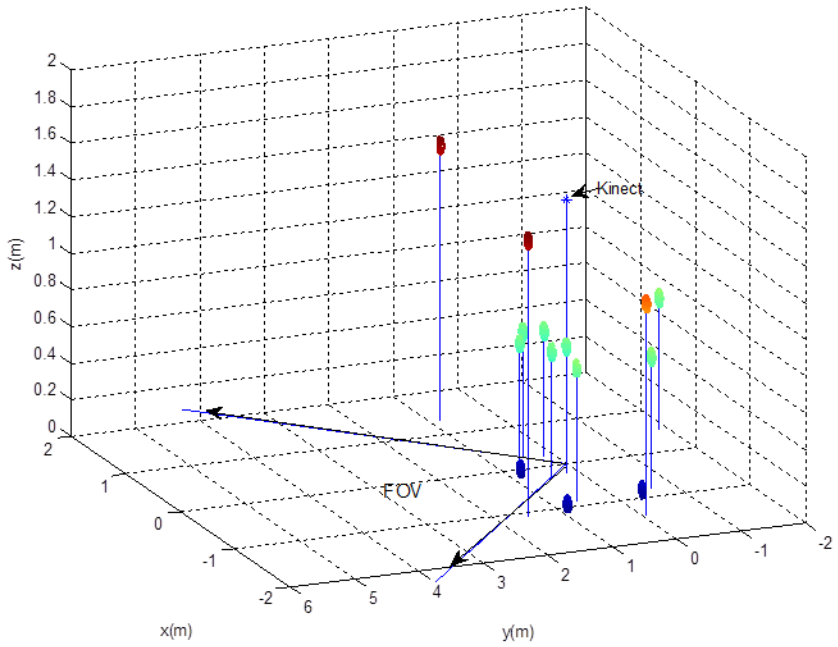


Figure 5-8: Objects locations in the experiments with the Kinect sensor

The experiments were conducted not only to evaluate the ability of each pointing method to select correct objects, but also to investigate the preference of pointing methods that people tend to use when the targets are placed at different heights and distances. We used two ways to categorize the objects: first, with regard to the height of the targets, they were classified into three levels, i.e. low (<0.7m, e.g. on the floor), medium (0.7m≤ height ≤1.5m, e.g. between the waist and the neck), and high (>1.5m, e.g. above the head); second, with regard to the distance, they were classified into three levels, i.e. near (nearer than 1.5m), medium (in between 1.5m and 3m), and far (further than 3m). Thus, each object had two attributes with regard to height and distance.

In total, 134 pointing gestures were conducted with the PMD camera set, and 90 with the Kinect. Object selection ability (correct selection and false selection), pointing gesture detection and the average angular error were analysed.

The angular error of a pointing direction is defined in the following way. Denote the pointing hand position as H, and the target location as O. Thus, the vector connecting the hand and the target is $V_r = O - H$. We take the angle between the estimated pointing vector and V_r as the angular error of this pointing direction.

First, we evaluated the overall performance of the pointing methods with different 3D sensing devices on the robot. Instead of using Equation (5.6), a correct selection is achieved if the Hand-Target line has the smallest angle with the pointing direction among all of the objects. With the PMD camera set, the Eye/Face-Hand line produced an average angular error of 15°, allowing for 99 correct target selections out of 134 pointing gestures, whilst the forearm direction resulted in an average angular error of 23°, allowing for 78 correct selections. With Kinect, the Eye/Face-Hand line performed an average angular error of 12°, allowing for 76 correct target selections out of 90 pointing gestures, whilst the forearm direction performed an average angular error of 20°, allowing for 62 correct selections. As we expected, Kinect slightly outperformed the PMD camera, because its depth map has a higher resolution and contains less noise, so that the detected hand position and 3D coordinates of the points on the forearm were more accurate. Details of the result can be seen in Table 5-1 and Table 5-2. The meanings of the abbreviations in these two tables are: CS (Correct Selection), FS (False Selection) and MD (Mis-detection). It shows that, on average, the Eye/Face-Hand line had better performance than the forearm direction.

Table 5-1: Overall estimation result of different pointing methods with the PMD Camera set

PMD Camera set	CS	FS	MD	Overall	Angular Error
Eye/Face-Hand Line	99	22	13	134	15 º
Forearm Direction	78	34	22	134	23 º

Table 5-2: Overall estimation result of different pointing methods with the Kinect

Kinect	CS	FS	MD	Overall	Angular Error
Eye/Face-Hand Line	76	8	6	90	12 º
Forearm Direction	62	18	10	90	20 º

To include the influence of the object locations, Table 5-3 and Table 5-4 show the overall preference of pointing methods associated with objects locations. The number in the table is the ratio of these two preferences (i.e. Eye/Face-Hand line: Forearm direction), which are calculated by Equation (5.4).

Table 5-3: The ratio of the preferences associated with object locations (using PMD)

	Low	Medium	High
Near	0.6624: 0.3376	*0.5485: 0.4515*	0.8324: 0.1676
Middle	0.6630: 0.3370	*0.5492: 0.4508*	0.8356: 0.1644
Far	0.7597: 0.2403	0.6667: 0.3333	0.8889: 0.1111

Table 5-4: The ratio of the preferences associated with object locations (using Kinect)

	Low	Medium	High
Near	0.6019: 0.3981	*0.5657: 0.4343*	0.7759: 0.2241
Middle	0.6113: 0.3887	*0.5753: 0.4247*	0.7683: 0.2317
Far	0.8398: 0.1602	0.8187: 0.1813	0.9231: 0.0769

It can be seen from these two tables that, among all the participants, the Eye/Face-Hand line is significantly preferred compared with the forearm direction when the

participants pointed to the objects whose location is high, low or far away, while there is no obvious difference in the preference for the objects which are at a medium height and also at a near or middle distance (as shown by the numbers in italic). In addition, these two tables also show that people's preferences are indeed independent of what capturing device is adopted. This result provides a general guide when the robot interacts with an unrecognized person, in which case the robot has no prior information of his/her particular preference as to pointing method.

However, the average result can not represent each individual's preference. As observed in the experiments, some people seemed to never use the forearm direction in all of the experimental trials. The same table as Table 5-3 or Table 5-4 was built for each individual subject. This database stores the preference of the pointing method for each known user, which can be taken into account when the robot interprets the pointing direction conducted by a recognized user (the user's identity can be determined by the method proposed in Chapter 3).

We analysed the recorded data for the second time, considering the preferences of each individual associated with object locations and the quality of the pointing vectors. The PRS of each object to be selected by the pointing direction was calculated using Equation (5.6) and the one with the largest PRS was chosen as the selection. The result was 112 correct selections out of 134 pointing gestures for the PMD setup; 83 out of 90 for the Kinect setup. This means that the object selection rate is improved by about 9%.

Mis-detection in the experiments occurred due to two main reasons. One is because some subjects did not hold the pointing arm for at least 1second. Due to the motion blur effect, the 3D data is not sufficiently consistent to produce stable vectors. If the duration is too short, the hand is considered as still in motion. Another factor depends on how we set the angle threshold which decides if a newly arrived vector is similar enough to the current mean-vector of the group. If the threshold is set too small, then the new vector is less likely to be grouped, causing the duration of the mean vector to be short and thus not stable. Currently, we empirically set it to be 3 °.

5.6 Discussion and Conclusion

This chapter has described the methods for detecting the occurrence of a pointing gesture and calculating two kinds of pointing direction the Eye/Face-Hand line and the

forearm direction. The hypothesis that the choice of a pointing method could be related to each individual's preference and the location of the intended target has been verified in the experiments.

A person's pointing direction is very subjective. We conducted an experiment involving two persons. One person pointed to a specific location and another person tried to assess the targeted location. The result was surprising, showing that even humans may not always recognize the actual pointing direction issued by another person, depending on the viewpoint of the observer. Schauerte et al. [251] also asked several humans to identify the referred-to object in the recorded images. Interestingly, the participants were able to identify the correct object for only about 87% of the images. In addition, in object selection tasks, if the person is facing many objects, he/she may point carefully, otherwise, he/she may conduct the pointing gestures casually, which may result in larger angular errors.

It is also noticed that the subjects did not conduct pointing gestures as naturally as in their daily life. This may be because they were aware that their movements were being recorded and analysed, and they were communicating with a robot rather than a real person. This may cause the magnitude of their movements to be larger than actually needed, making the arms outstretched more often in experiments. We designed a questionnaire about the personal feelings of their behaviour after they had finished the experiments to ascertain whether their gestures were different from their normal daily behaviour because of being recorded, and their answers accorded with our hypothesis.

Another limitation of the current system is that it cannot deal with the pointing gestures when the person mainly uses a rotated wrist to perform a pointing action as shown in Figure 5-9. Although the hand shape can be recognized to tell whether the finger is pointing towards the left or right, as described in Chapter 4, it is difficult to calculate the 3D vector. One possible way is to use model-based method to reconstruct the hand, but it hardly works when the person is standing at a 2m distance making the size of the observed hand small and producing few valid 3D points. In addition, it cannot deal with the cases when the user is pointing behind his/her body. Hand tacking and forearm extraction become difficult and unreliable in this situation.

Figure 5-9: An example of pointing with a rotated wrist.

We noticed through the experiments that the objects' arrangement and the person's viewpoint can make a significant difference to object selection ability. A large angular difference makes it easier to select the correct object. As shown in Figure 5-10, the distance between A and B is larger than that between A and C, but obviously vector 2 is more likely to select the target C because the angle between PC and PA is larger than the angle between PB and PA. Therefore, it would be helpful to diminish the ambiguity if the robot can ask the user to point again from another point.

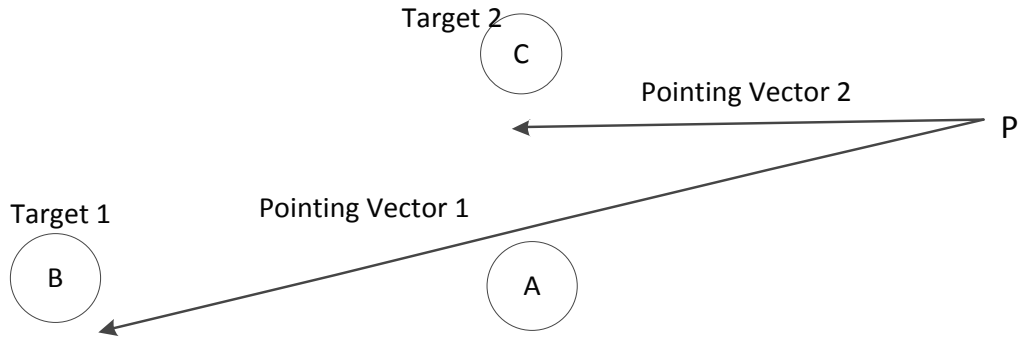


Figure 5-10: Object arrangement and the viewpoint make a difference to the object selection ability. Both pointing vectors start from point *P*.

Instead of selecting a single object, a list of the top-n objects can be kept as candidates. Some researchers define an apex angle for the estimated pointing direction, making the pointing direction as a cone [271], and consider all the objects within this cone as candidates. If the apex angle of the pointing cone is a constant value, the number of object candidates is variable. When multiple information sources are available, the objects

are not selected by the pointing direction only, but by the combination of the information provided from various channels.

6

Multimodal Interaction: Using Gestures, Speech and Gaze

For a humanoid robot to participate in our daily life and interact with humans in a natural style similar to human-human communication, it is essential that the robot is able to understand humans' natural communication modes, including speech, gaze, gesture and facial expression. A multimodal interaction system can provide the flexibility that allows users to select from or mix different input modes. The flexibility of a multimodal interface can accommodate a wide range of users and environments, for example, handicapped users, noisy environments and many other cases which cannot be recognized by unimodal input.

It is natural, effective and thus overwhelmingly preferred for humans to convey their intentions through multiple means. For example, it is visually vivid and commonly used for a person to say "the table is this long" and accompany the statement with a two-handed gesture showing the length. Another example is that the speech utterance "I like that one" accompanied by a pointing gesturing indicating the desired object. In these situations, it is impossible for the robot to fully understand the user's intention if it does not recognize both the speech and the gesture at the same time. Furthermore, in addition to explicit pointing gestures, the person's head pose or eye gaze direction is also a critical clue of his/her intention. Although a pointing gesture is conducted by a pointing hand, people naturally look at the target before carrying out pointing actions. Another example is when a person is staring towards a certain direction, which shows the person's interest lies in this direction even without any speech or action. Linguistic analysis reveals that the spoken and gesture modes consistently provide complementary, rather than redundant information [221].

Many multimodal interface architectures [97, 124, 141] are designed to cope with ambiguity. Disambiguation of error-prone modalities is an important motivation for the use

of multiple modalities. Spoken utterances can be ambiguous; even a correctly recognized sentence may lead to several hypotheses [102]. Likewise, gestures are also uncertain. A gesture can have multiple interpretations in different contexts or by different expressers.

Multimodal interfaces have already been proven to significantly prevent errors and enhance the effectiveness of the communication. For example, task-critical errors and disfluent language were reported to drop by 36-50% during multimodal interaction [219]. Using multimodal pen/voice interaction, a temporal speed up of 10% was also noticed compared to an unimodal speech input [219].

This chapter first presents a review of multimodal interaction systems, followed by the description of our approaches to recognizing a person's speech utterances, and estimating his/her attention direction. A multimodal interaction system is proposed, which integrates speech, head pose, eye gaze and the gesture recognition (described in Chapter 4 and 5). Moreover, human-robot interactions are often in a two-way dialogue paradigm [106]. A dialogue manager uses a task-orientated command table to determine whether the fusion result is complete and ready for execution. If not, the robot needs to request more information from the user to eliminate any ambiguity in its understanding. In other situations, when for example, the robot thinks the task is high cost or risky to implement, it may need to confirm its understanding with the user before execution.

Furthermore, the collaboration work towards our final goal of a mobile assistive robot is presented here, which combines the transactional intelligence in this thesis with the spatial intelligence system developed by colleagues. Part of the work in this chapter was published in [167].

6.1 Related Work

Since Bolt demonstrated his seminal "Put-That-There" system [23], which processed language commands together with deictic hand gestures, a number of multimodal systems have emerged, such as Virtual World [48], CUBRICON [206] and QuickSet [123]. Early investigations integrated speech with mouse or pen pointing or drawing [123]; while more sophisticated systems bound language commands with natural gestures [97, 268], facial expressions [87] and/or eye gaze [141, 302].

Johnston et al. [124] utilized a unification method [203] over typed feature structures [42]. The unification method determines the consistency of several pieces of information, which can be combined into a single result if they are consistent. The multimodal integrator agent determined and ranked all possible unifications of speech and gestures and issued complete commands as the output. In this way, the ambiguity of the gestures was resolved by speech and similarly gestures also compensated for the errors in speech recognition. The method was implemented in "QuickSet", which was a distributed interactive simulation system with a multimodal interface (i.e. pen and voice).

A constraint-based multimodal fusion system for speech and pointing gestures was proposed by Holzapfel et al. [107]. They also employed a typed feature structure method on the semantic level. This work extended Johnston et al's work and provided a fusion that considered possible false detections. The pointing gesture recognition subsystem returned a sorted list of objects based on their relative distance to the pointing direction. The speech recognition subsystem mainly specified the action command and the type of the desired object. The multimodal fusion component synchronized and combined the output of the subsystems and sent the result to a dialogue manager. It took the speech as the main modality and used gestures for disambiguation purposes. They also considered the difference between the arrival time from the recognizer and the actual occurrence time of the event. Events remained in a pool if they were not compatible with fusion rules, and were faded out after a predefined time window.

Kaiser et al. [133] proposed a multimodal interaction system in augmented and virtual reality. It fused symbolic and statistical information from 3D gestures, spoken language and referential agents. They used four 6-DOF magnetic sensors attached to the user's hands, arms and head to achieve accurate detection of the hand gestures, pointing direction and the looking direction. The unification method with typed feature structure was then employed to fuse complementary logical variables.

In [64] the salience of candidate gesture-speech bindings was calculated using a hybrid of data-driven and knowledge-based methods. A penalty function was built, which considered the factors including the time gap, the order of each gestural and spoken symbol, and affinities between different words and gestures. A naïve gradient descent algorithm was used to find the binding with the minimum penalty score.

A semantic network was established in [244], in which the nodes were activated depending on their relation to the given commands. The fading mechanism decreased the activation values of the nodes in the network according to a fading function. The interpretation history was also considered in their semantic network.

Harte and Jarvis [97] presented a robotic system that fused speech, vision and laser-depth data to perform tasks in a domestic environment. Contextual information, which included the history of recently uttered phrases and the objects' attribute, was taken into consideration. A probabilistic method was used to cope with the possible incorrect recognition.

Schmidt-Rohr et al. [252] described a method of modelling human-robot interaction as Partially Observable Markov Decision Processes (POMDPs), which can model uncertainties in robot perception and human behaviour. With POMDPs used for decision making, the robot was able to handle uncertainty in both observation and environment dynamics and could balance multiple, conflicting goals in their experiments on a service robot.

Some researchers view speech as a self-sufficient primary input mode and consider gestures, gaze direction and other input as secondary simply providing redundant accompaniments that carry non-significant information [220]. Holzapfel et al. [107] took speech as the main modality and used the gestures to disambiguate speech input. In this way, their fusion method was tolerant against falsely detected pointing gestures.

However, speech signal can be degraded (e.g. in a noisy environment) and the speech recognizer is not guaranteed to provide correct result even when the audio input signal is of good quality. More importantly, other modes can convey important information. Mutual disambiguation and error prevention are desirable features in multimodal interaction systems. Interaction systems that ignore some sources of input information will systematically fail to recognize many cases of spontaneous multimodal construction [220]. The unification-based integration of spoken and gestural input by Johnston et al. [124] allowed the modalities to mutually compensate for each other's errors.

6.2 Multimodal Interaction in a Probabilistic Framework

A household assistive robot mainly serves in domestic places such as living room, office and kitchen. For the moment, we are generally interested in two types of tasks which include navigating the robot (e.g. “turn left/right”, “go there” and “go to the kitchen”.) and requesting the robot to bring an object to the user (e.g. “give me the red cup on the table”, “bring me my cup over there”).

The system integrates three types of information, speech, gestures and attention direction. Three channels of input data are first captured and recognized independently, generating semantic results, which are then fed to the multimodal fusion agent. A probabilistic method finds the most likely instance of each class in a task-orientated command table. The system then determines whether the recognized information is complete to trigger execution by the robot. A dialogue is activated if the robot needs to confirm with the user before its execution or more information is required to implement any task. The complete flow chart of the multimodal interaction system is shown in Figure 6-1.

Several sub-systems have already been described in previous chapters, including user identification in Chapter 3, hand gesture recognition in Chapter 4, and object selection by pointing gestures in Chapter 5.

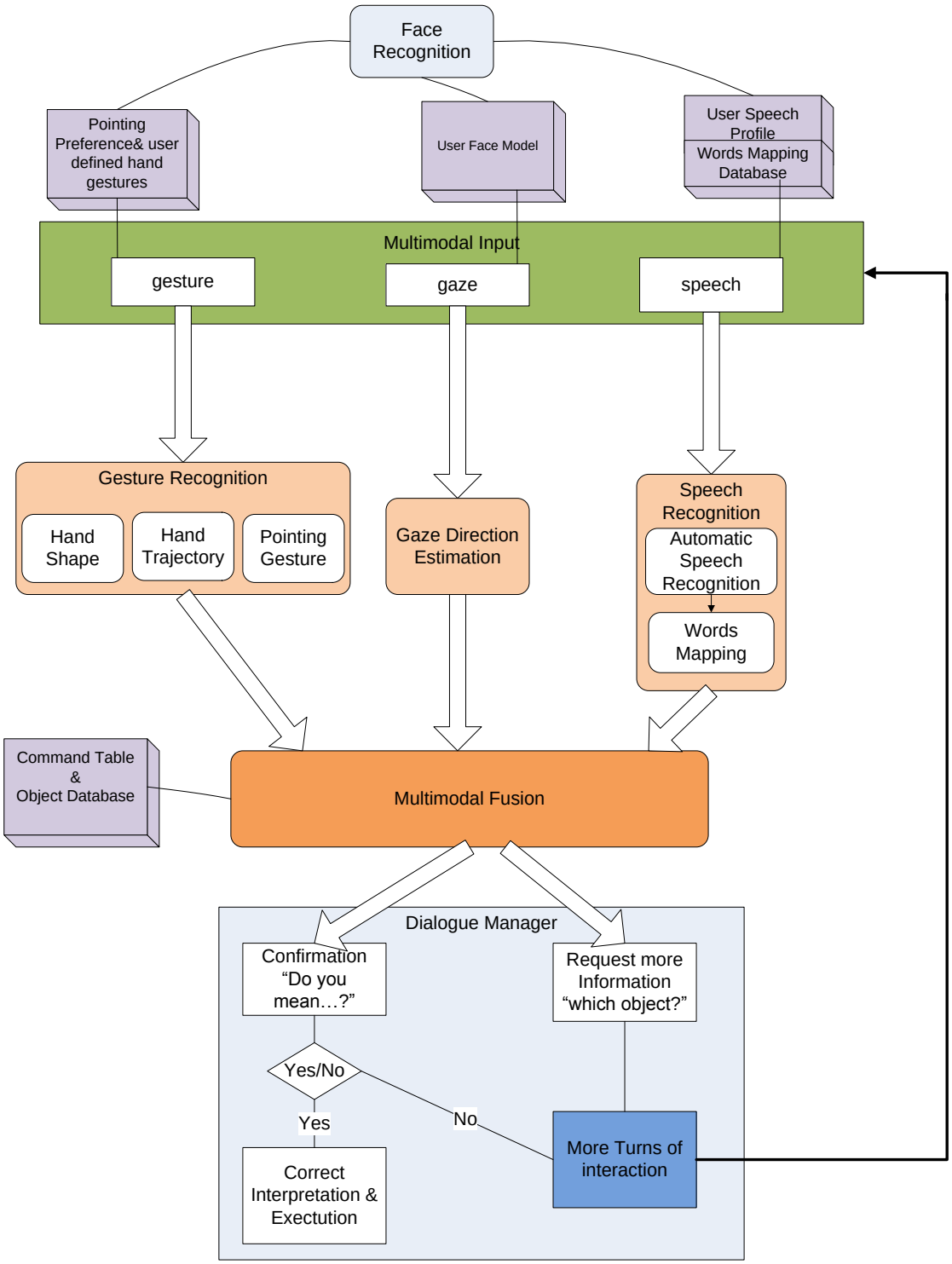


Figure 6-1: Flow chart of the multimodal interaction architecture

6.2.1 Speech Recognition and Word Mapping

Since the main research area of this thesis focuses on the visual interpretation of humans' gestures, our speech recognition method is simple, taking advantage of an existing speech recognition engine followed by an intuitive word- mapping approach.

Microsoft Speech SDK 5.1 [191], which includes the advanced Automatic Speech Recognition (ASR) engine, is employed for our speech recognition sub-system. Its speech synthesis engine is also used for the robot to generate audio feedback in Section 6.2.5. A confidence value for each single word is generated which reflects the confidence of the ASR result.

A training stage, which involves reading a list of articles, is normally important to improve the engine's ability to understand the user's voice. Building a speech profile for each individual user is highly recommended, rather than using the default profile. In our system, the speech profiles can be automatically switched to the current user, after the user's identity is recognized by the user identification component proposed in Chapter 3.

However, the effort needed in the training stage may be non-trivial. A researcher, who was born and grew up in New Zealand, spent approximately seven hours to train this speech recognition engine before it could recognize most of the 60 words that were used in their system [97]. In fact, recognition performance may remain far from satisfactory even after hours of training, especially for non-native speakers.

As mentioned in previous chapters, the pointing methods are chosen according to each individual's preference. Users can also define their own sets of hand gestures. Similarly, to compensate for insufficient training effort, the speech recognition results could be refined for each individual by a word-mapping method.

The ASR provides a context-free recognition result, which is then enhanced by the scenario constraints. The interaction scenario determines, to some extent, the vocabulary about which the users are likely to talk. For our household assistive robot, possible words include the common objects in a living room, office and kitchen, their properties and simple manipulation actions and navigation commands.

Therefore, it is reasonable to map some words which are unlikely to be spoken to the robot in these scenarios, such as "cop" or "copy", to certain words which may be

frequently used in these places, such as "cup". This is similar to how humans interpret sentences considering the context of the conversation. As shown in Figure 6-2, each recognized word (W_i) is given a confidence value (p_i) and occurrence time (t_i) by the ASR. The words in the vocabulary are kept without changes while others are mapped to new meanings. Non-keywords, such as "please" and "thanks", and words out of our current vocabulary are ignored.

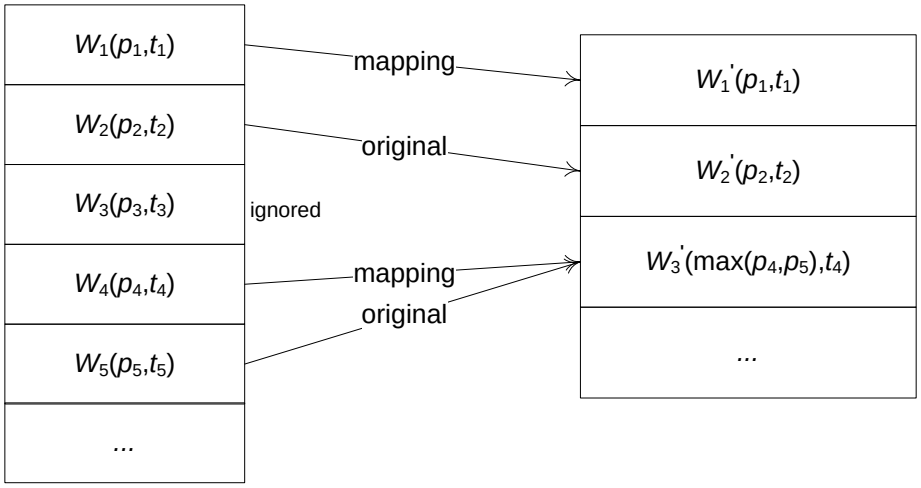


Figure 6-2: Examples of word mapping.

When some words are wrongly but consistently recognized as certain words by the ASR, the built mapping table projects them to their intended meanings. The disadvantage of this method is that it may narrow the range of topics which can be understood by the robot, because it may be a many-to-one mapping, and their meanings are changed. However, this trade-off is acceptable and showed advantages in our experiments because we currently focus on the interaction in the kitchen or living room scenario. The mapping database can be reduced or removed if we are given a better speech recognition engine in the future, but is currently important to compensate for the errors due to the accents of the individuals, insufficient training effort and low quality of the speech signal in the noisy environment.

In addition to pronunciation, many other factors may be considered when applying the word-mapping method for each individual. For example, some people may have difficulty in distinguishing certain colours, such as green, blue and cyan, and some people may not make a distinction between cups and mugs. This consideration would be desirable to endow the robot with humanization and sociality. As mentioned in [114, 258], for an even more sophisticated robot, the relation of a person's behaviour with his/her personality,

cultural, mood and the context in which the observed behavioural cues are encountered should also be considered.

6.2.2 Attention Direction Estimation

A body of research work [141, 178] suggests that people tend to look at the objects with which they interact. Nickel et al. [212] noticed in their experiment that people tend to look at the target before and during the time when the pointing gestures are conducted. It is intuitive since people normally need to see where the target is before they conduct the pointing action. Where the user is looking indicates where his/her attention lies. Therefore, objects along the direction of the user's gaze are more likely to be the intended target. This suggests that integrating the tracking of the subject's gaze into our system would probably improve object selection performance by the robot.

6.2.2.1 Related Work

A number of methods have been proposed to estimate a person's head pose or eye gaze direction. Breitenstein et al. [33] estimated the 3D face poses from low-quality range images. Their algorithm generated many pose candidates from a signature to find the nose tip based on local shape (regions with high curvature) and then evaluated candidates by an error function that used pre-computed reference pose range images, motion direction estimation and temporal consistency. Their system was claimed to achieve an accuracy of 83.6% for a maximally allowed error of 30 °.

Chutorian and Trivedi [204] proposed a driver-assistance system which used head pose estimation for monitoring driver awareness. Their initial pose estimation module used localized gradient orientation histograms as input to support vector regressors. The tracking module provided a fine estimate of the 3D motion of the head using an appearance-based particle filter. The system used a single camera and was able to estimate a continuous range of head pose at 30 fps.

Yoo et al. [297] presented a 3D user interface combining gaze and hand gestures for large-scale display. Three colour cameras and one time-of-flight depth camera were employed. Face features were detected and tracked in multi-views and then fed to a Support Vector Regression algorithm to estimate the face 3D position and orientation (pitch and yaw). Their algorithm achieved 10 degrees of accuracy. However, they

considered the gaze direction to be the same as the head orientation. In fact, where the eyes are looking can be different from the face orientation. Detection of the real eye gaze direction normally requires close-up views of the face.

A number of gaze estimation methods rely on glints (i.e. the reflection of light off the cornea) to build 2D or 3D gaze models [96]. Alternatively, one can use the pupil or iris contour or the distance of the iris centre from some reference point such as the corner of the eyes for gaze estimation. 3D rendered eyeball models can also be used [238].

Reale et al. [238] detected the user's face from a fixed, wide-angle camera, estimated a rough location for the eye region, and then directed another active pan-tilt-zoom camera to focus on this eye region to find the 3D eyeball location, eyeball radius and fovea position. They also mapped both the iris centre and iris contour points to the eyeball sphere to find the optical axis, then rotated the fovea to find the final gaze direction.

6.2.2.2 FaceLAB

A person's attention direction can be estimated from the head orientation or, more accurately, the gaze direction. For this purpose, some intrusive systems strap a camera to the user's head, or attach markers on the user's face [17], which are inconvenient and uncomfortable for the users. A commercial stereo-vision system called "faceLAB" (version 1.1, developed by the Seeing Machines Company) [253] is adopted as shown in Figure 2-2, to capture the user's head pose (position and orientation) and eye gaze direction. Images from the cameras are analysed to work out characteristics of a person's face. Then the head position, orientation and gaze direction are calculated from these features.

By adjusting the relative position of the stereo cameras, five configurations are allowed on the installation slots, so that the cameras converge at different distances (0.5m, 0.75m, 1.0m, 1.5m and 2m). Like other stereo systems, calibration is required to calculate the intrinsic and extrinsic parameters of the two cameras each time the placement of the stereo cameras is changed. To achieve good results, a face model for each user is manually built by capturing images from five viewpoints, selecting visual features and adjusting their corresponding positions in left and right image pairs. A model should include stable visual features of the face. These features include the corners of eyes, mouth, nostril and sideburns. They will not change because of slight alterations of viewpoint and illumination

conditions during head movement and non-extreme facial expressions, and the features of the eyes (such as the radius of the iris). A gaze calibration procedure is conducted by staring at each camera directly for 2~3 seconds. The gaze direction relative to the head pose is calculated by tracking the location of the pupils within the frame of the eyes. Each user can build his/her own face model. Automatic selection of which face model to use can be achieved, in theory, if combined with the user identification sub-system. However, this is not implemented for the moment, because the SDK of the faceLAB is not yet available to access deep control of the program. Figure 6-3 shows an example of a detected head pose (the green line, starting from the central point between the two eyes slightly to the front-right) and the gaze direction (the red lines, starting from the centres of the two eyes to the front-left). Figure 6-4 depicts a 3D model of the user's head.

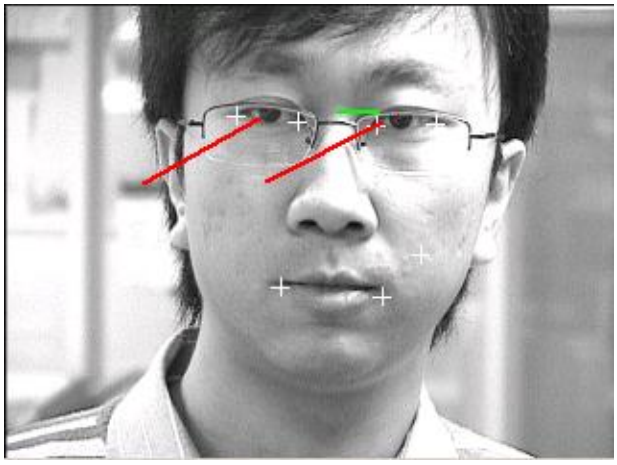


Figure 6-3: A snapshot from the faceLAB, showing the head orientation (the green line), and the gaze direction (the red lines).

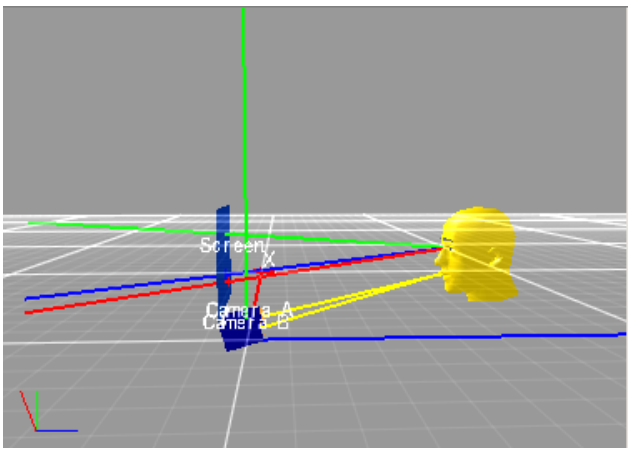


Figure 6-4: The corresponding 3D head model of the person in Figure 6-3.

The output from the faceLAB is sometimes quite noisy and unstable for the following four main reasons.

1. Non-uniform lighting conditions，especially under fluorescent lights in the office environment, cause the appearance of the visual features to vary greatly. Gaze estimation is particularly sensitive to the shadows around the eyes. Controlled environmental conditions with uniform lighting are preferred to minimize the shadows on the face.
2. Glasses are not ideal for tracking because they introduce reflection and inconsistencies into the face model; however, glasses are often inevitable. Most of the participants in our experiment need to wear glasses.
3. The features of the eyes are subtle (especially observed from a distance of 1.5~2m) and unstable (eye movement has extremely high angular velocity). The features are also frequently affected by blinking and frowning.
4. When the head rotates, the changes of the viewpoint also make the features look different (although the face model includes images from five viewpoints, it does not solve the problem completely).

Although with markers, the faceLAB is reported to give much better estimations, we insist on conducting our experiments without markers, consistent with our goal of developing a natural and convenient system.

In [115], Jarvis demonstrated that the faceLAB could be applied to control a wheelchair by the user's gaze direction. It can be potentially used by disabled people who lack the capacity of operating a vehicle using their hands or in situations where mower drivers want to free their hands for other operations. On a vehicle, the driver does not move his/her body much, so that it is possible for the faceLAB to track the gaze continuously. However, because the faceLAB can only focus on the user's face within a narrow shallow region, in a situation where the subjects can move freely, the faceLAB loses track frequently. Therefore, to investigate the use of the faceLAB, in our experiments, constraints are imposed on some participants to limit their body movement on the premise that they can still perform pointing gestures efficiently.

The head moves much more slowly than the eyeballs, and the features on the face are more stable and easier to detect than those in the eyes. Therefore, the estimation of the head pose is more stable than the eye gaze direction. However, in real life, people do not really face the target precisely, provided that the target can be seen by means of simple

eyes movements. Therefore, although eye gaze gives the actual attention direction, the direction of the eye gaze is much more unstable and difficult to estimate.

Given the confidences values of these two estimations generated by the program, they are combined to form an attention direction. If both of the two confidence values are too low, the current estimation is discarded. If the head orientation is within a specific value (e.g. 45 °), the attention direction is calculated using Equation (6.1)

$$v = \frac{c_1}{c_1 + c_2} v_1 + \frac{c_2}{c_1 + c_2} v2 \tag{6.1}$$

where c_1 and c_2 are the confidence values of the estimations of the head direction v_1 and the eye gaze vector v_2 respectively. Otherwise, the gaze direction is ignored, taking the head pose as the attention estimation. This is because the faceLAB is more likely to generate accurate gaze estimation when the user is approximately facing it, with the features easier to be observed and analysed.

The data from the faceLAB is processed by a Median Filter and the stable vectors are selected as the valid attention directions (stability is measured in the same way as the pointing vectors as described in Section 5.3). The quality of the attention direction is also calculated according to the duration length and the with-group variance, which is the same as Equation 5.5. In contrast to the pointing vectors, the attention direction is always considered as meaningful. The PRS that an object is looked at is calculated by Equation (6.2)

$$p(obj_i) = q \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\theta^2} \tag{6.2}$$

where q is the quality of the attention direction and θ is the angle between the eye-object line and the attention direction.

In addition, the changes of the head pose can be used as head gestures, such as shaking or nodding head, which can be recognized by the Finite State Machine method as described in Chapter 4.

6.2.3 Alignment

In this subsection, the alignment problem between different modalities including speech, gestures and attention direction is discussed.

Gesture-speech alignment involves choosing the appropriate gesture to ground a key verbal utterance. This is important if more than one gesture occurs during speech. For example, there may be two pointing gestures accompanying the sentence "put that cup over there". One pointing is coupled with the word "cup" and the other explicitly points out the spatial position referred to by the word "there". Some researchers have investigated this problem and Eisenstein [64] summarized notable findings about gesture-speech bindings. In our system, the following points are considered:

1. The gesture is usually close in time to the relevant keyword and normally precedes the keyword [220].
2. Some word-gesture combinations are particularly likely. For example, a pointing gesture and the word "this", "that", "here" or "there" [64].

However, the above are merely general phenomena; there are also individual differences in binding patterns. As discussed by Oviatt [220], in their experiments with a multimodal pen/voice system, users adopted either a simultaneous or sequential alignment pattern when combining speech and pen input. The integration pattern for each user was established early and remained consistent through all the trails.

In addition, falsely-detected gestures can also be filtered out by gesture-speech alignment. Take the pointing gesture as an example. As discussed in Chapter 5, a threshold for the duration is used to determine whether a pointing arm is sufficiently stable to be considered as a valid pointing gesture. Lowering this threshold would reduce the risk of mis-detection, but has the side effect of more false positives. The false positives can be filtered out based on two factors: the first one is the occurrence timeslots of the words and the detected gestures. If the pointing vector is temporally far away from spoken keywords then it is filtered out (shown in Figure 6-5); the second factor is the quality of the pointing vector itself (the quality is calculated based on the duration and within-group variance of the pointing vectors, see Section 5.4 for details). Therefore, if there are two pointing gestures in the temporal neighbourhood of the keyword, the following method is used

rather than simply choosing the closest one. A Probability Related Score (PRS) of the gesture to be coupled with the spoken word is found by Equation (6.3).

$$p_i = \begin{cases} 0 & , if\ \Delta T_i\ is\ out\ of\ T_w \\ w_1 \cdot Q_i + w_2 \cdot e^{-\frac{\Delta T_i^2}{T_w^2}} & , otherwise \end{cases} \tag{6.3}$$

where Q_i is the quality of the i^{th} gesture, ΔT_i is the temporal distance between the occurrence of the i^{th} gesture and the spoken word, w_1 and w_2 are their weights respectively, and T_w is the predefined time window. Let $j = \arg\max_i p_i$, if p_j is over a specified threshold then the j^{th} gesture is coupled with this spoken word.

Similarly, only the captured attention vectors within a time window that is specified by the spoken keyword are chosen using Equation (6.3).

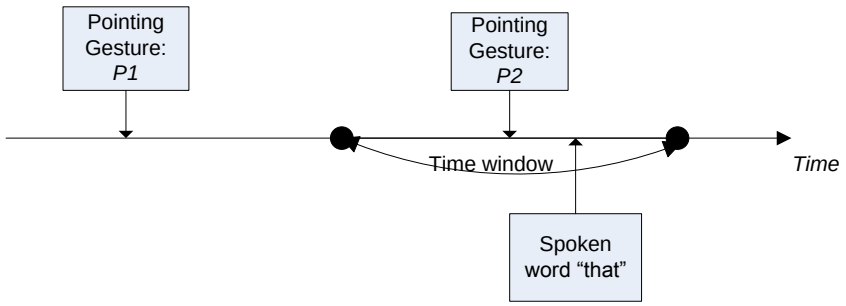


Figure 6-5: The pointing gesture P_1 is filtered out by the recognized spoken word "that", because they are temporally far away from each other.

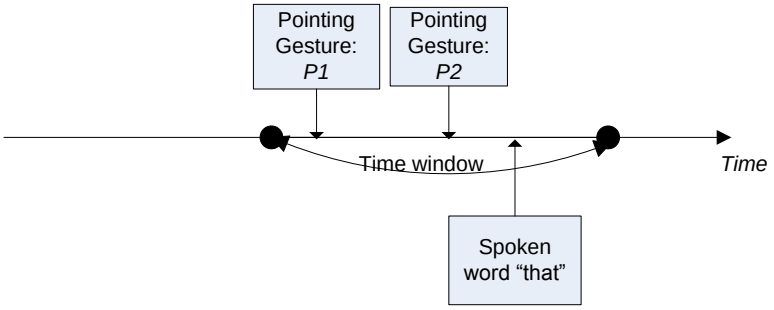


Figure 6-6: When two pointing gestures are detected within the time window of the spoken word "that", we do not simply choose the closest one. Instead, the valid one is chosen using Equation (6.3).

However, if none of the demonstrative pronouns are spoken or recognized, estimated attention directions are aligned with meaningful and stable pointing gestures using Equation (6.3) in the same way as the gesture-speech alignment method, while Q_i is the

quality of the i^{th} attention vector and ΔT_i is the temporal distance between the occurrence time of the i^{th} attention direction and the valid pointing gesture.

In the implementation, the actual occurrence time of the event and the output time from the recognizer could be different. For example, the ASR only gives the recognized text after the voice signal has an obvious pause which indicates a sentence is complete. Holzapfel et al. [107] also took this into consideration in their work. However, they did not explicitly find the occurrence time, but simply extended the constraints so that the fusion agent waited for a certain amount of time for a gesture event if the speech was ambiguous. In our system, the occurrence time of each event is found explicitly. The occurrence time of each spoken word is calculated by the start time of the phrase and the offset time of each word from the start time, both of which are generated by the ASR. The occurrence time of the pointing gesture is calculated by the output time generated by the recognizer subtracting its duration time. The time of the gaze direction is calculated by the output time subtracting the estimated delay time, both of which are provide by the faceLAB data stream (as described in Section 6.2.2).

6.2.4 Information Fusion at Decision Level

Information fusion methods are categorized into three levels (i.e. data, feature and decision) in [256]. Fusion at the low level of raw data is the integration of information from the sensors of the same type; feature level fusion refers to the closely- coupled modalities such as speech and lips movement; decision fusion fuses the interpretation results of each individual mode [244]. In our task-orientated interaction system, the fusion is implemented at the decision level. Three recognizers for speech, hand gestures and attention direction process the captured data from different information sources independently and generate recognition results at the semantic level which are then fed to the fusion agent.

6.2.4.1 Task-orientated command table

According to the tasks possibly involved in our household robotics applications, a command table is defined which decides what information is needed to form an executable task. Different constituents of a complete command are categorized into several classes such as action, object, direction and place. Each class has several instances. For example,

the action could be “pick up”, “put down” and “turn” and so on. Some examples are shown in Table 6-1. “√” and “×” represent whether an instance from the class is required or not for a task that is characterised by the action in the first column. The “Go to” command changes the location of the robot itself, while the “Move” command here means an operation to change the position of an object. The “Put down” command may need an explicit location to indicate where the robot should place the object, or the robot can simply open its hand and the user will catch the object. Therefore, there are two variants of the “put down” command. The second one was actually used in our experiments, which is an easier operation for the robot.

Table 6-1: Examples in the command table

Action	Object	Place	Direction
Pick up	√	×	×
Turn	×	×	√
Go to	×	√	×
Put down 1	√	√	×
Put down 2	√	×	×
Move	√	√	×
…			

An action word is almost always necessary to define a task for the robot to execute. In some situations, the verb is omitted because it can be inferred from the context. Other basic constituents of a sentence may include objects, (“pick up that *apple*”), directional words (“turn *left*”) and places (“go to the *kitchen*”). Each object has a unique ID since the robot must localize a specified object when there are several identical ones in the scene. An object has some attributes including type, colour, location, size and movability. A place here refers to a 3D coordinates that can be specified by a pointing gesture or a predefined location. For example, for the command “go to the kitchen”, the spot where the robot should stand when it enters the kitchen is pre-defined in advance.

6.2.4.2 Probabilistic method for multimodal information fusion

In multimodal systems, in the ideal case each mode provides complementary information without errors and the fusion agent simply assembles them as the final output.

Koons et al. [152] presented a multimodal interface that accepted speech, gestures and gaze input from a user. The command was given verbally while the location was provided from the user's pointing gestures or gaze. It was assumed that the recognition from each mode was accurate.

However, the recognition results from each mode can be redundant, complementary or even contradictory. More importantly, in our realistic experiment, no recognizer can be assumed to be error-free: not all these spoken words are correctly recognized, the pointing gesture may be far away from the target, the gesture may be mis-detected, or several pointing vectors may be false positives generated by the recognizer.

A probabilistic method is proposed here for information fusion. It determines whether the accumulated information is complete at the decision level and also takes into consideration the possible errors from each recognizer.

Different input modes may provide information to determine the instance in the same class. Take the object class as an example. When the user says "pick up that red cup" accompanied with a pointing gesture, the spoken words "red" and "cup" provide the type and colour properties of the intended object, while the pointing gesture designates its 3D location. At the same time, he/she may be looking at the object and the attention vectors could also distinguish the object from some other objects in the scene. These three channels of information can be used together to determine which object is the target. Similarly, information of other classes can also be provided through multiple input channels. For example, the speech utterance "pick up" can be accompanied with a mimetic hand movement; or "turn left with a static hand shape that indicates the left direction.

First, each recognition system processes the data and generates the recognition result with Probability Related Score (PRS) individually.

For speech recognition, as mentioned in Section 6.2.1, the ASR gives a confidence value for each recognized text. Among the properties of the objects in our system, colour and type are most likely to appear in the spoken utterances. Sometimes the user says the approximate region of the target, but only some simple cases are tackled, for example, "*centre of* the table" and "bring me the box *on the floor*". Therefore, the PRS of an object to be referred-to by speech is calculated by Equation (6.4)

$$p_{speech}(o_i) = w_c \cdot colour(o_i) + w_t \cdot type(o_i) + w_r \cdot region(o_i) \quad (6.4)$$

where w_c, w_t and w_r are the weights for colour, type and region attributes respectively. The value of the function $colour(o_i)$ equals the confidence value of the 'colour' word from the ASR if the colour attribute of the object o_i accords with the spoken word; otherwise $colour(o_i)$ equals 0. The same calculation applies to the functions $type(o_i)$ and $region(o_i)$.

For static hand gesture recognition, each hand shape template is assigned a matching score which corresponds to its dissimilarity with the test image (described in Section 4.4). Therefore, the PRSs of hand-shape gestures are calculated using Equation (6.5)

$$p(shape_i) = \frac{\min(score_i)}{score_i} \quad (6.5)$$

where $score_i$ is the matching score between hand template image i and the test image.

For dynamic hand motion gestures, the FSM recognizers have different numbers of total states. The state transition of each recognizer depends on the 3D position of the gesturing hand. When a recognizer reaches its final state, the gesture is recognized (described in Section 4.3). The PRSs of hand-motion gestures are calculated using Equation (6.6)

$$p(motion_i) = \frac{\#StateTrans(i)}{\#AllStates(i)} \quad (6.6)$$

where $\#StateTrans(i)$ denotes the number of finished state transition and $\#AllStates(i)$ is the total number of states of recognizer i.

The pointing gesture recognizer (Chapter 5) and the attention direction estimator (Section 6.2.2) generate deictic vectors. Pointing gestures, attention vectors and speech are aligned and false positives are filtered out using the proposed alignment method in section 6.2.3. Then the PRSs that an object is selected by the valid pointing gesture and the attention vector are calculated using Equation (5.6) and Equation (6.2), respectively.

The PRS from each mode are first normalized and then used to calculate the final PRS of each instance to be the intended action, target or direction. We take an object as an example to illustrate this method. During a command requesting an object, each object has

three PRSs from speech, pointing gesture and attention direction. The final PRS of each object to be selected is calculated by weighted addition.

$$p(o_i) = \sum_j w_j \cdot \bar{p}(o_i, m_j) \tag{6.7}$$

where w_j is the weight for each mode, $\bar{p}(o_i, m_j)$ is the normalized PRS of object i selected by mode j. A high weight is given to speech as 0.5. Because the pointing gesture is strongly aimed, its weight is assigned as 0.35. Gaze direction estimation in our system is relatively error prone, so its weight is 0.15. Eye gaze plays a critical role in some cases, although its weight is low. For example, in the situation when the user says "give me the red cup" without a pointing gesture (assume that all the spoken words are correctly recognized), if there is more than one red cup in the scene, all of which have equal probability of being selected by speech, only the one close to the user's attention direction then has a higher probability of being selected.

The objects are sorted by their PRSs ($p(o_i)$). A unique maximum PRS value indicates that the information for the "object class" is provided. Otherwise, it is still ambiguous about which object is intended.

It is important to state that, an autonomous and intelligent robot should be able to schedule the tasks according to the order of their urgency or importance. However, this consideration is simplified because for the moment we do not issue a second command until the current task is finished or given up by the robot. However, the "stop" command is one exception. This command is given the highest priority in consideration of safety issues. If the stop command is recognized by any mode, even with a low probability, the current operation is terminated immediately.

6.2.5 Dialogue Manager

The dialogue manager decides whether the accumulated information from all input modes is complete based on the command table. If all the necessary information to accomplish a task is available, the system is then able to translate the command, according to a predefined translation table, in several sequential steps of executions which can be performed by the robot. For example, "bring me that cup" can be translated to three steps: localize the cup, grab it, and go to the user.

Otherwise, more than one cycle of interaction will be requested by the robot, mainly under three kinds of situations that are listed as follows. Figure 6-7 depicts the flow chart of the dialogue manager.

1. The perceived inputs are not complete for any task. This is usually because the information for a certain class is not recognized. The robot needs to request the missing information.
2. The perceived result is ambiguous. This is because more than one instance of a certain class has an equally largest PRS. For instance, when the user says "pick up the cup" when more than one cup is in the table, the robot needs to clarify the ambiguity by asking, for example, "which cup do you want me to pick up?"
3. The robot's interpretation is incorrect. If the execution of a task involves a risky operation to the user or the robot itself, or requires a complicated operation or takes a long time to finish, the robot should confirm the command with the user before it starts the operation. This kind of confirmation is normally simple. For example, the robot asks "do you want me to go the kitchen?", and waits for a "yes or no" answer. However, for the purpose of testing the cognitive system without the physical operations by the robot, the robot is required to confirm its interpretation with the user for every command.

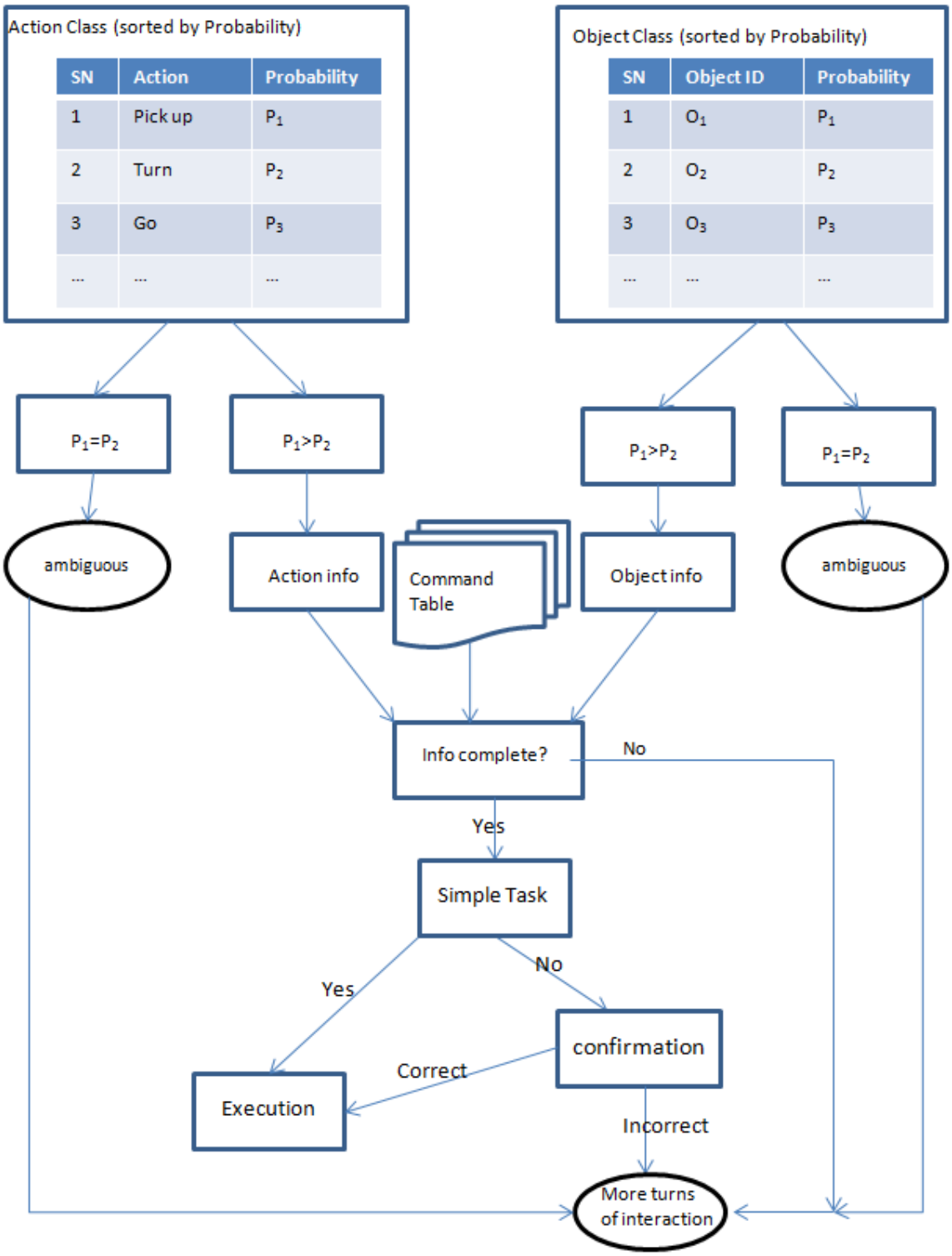


Figure 6-7: The flow chart of the dialogue manager. It shows an example of requesting an object, so only action and object classes are included here. An instance with the unique maximum probability related score indicates the valid information for the class is recognized. If the accumulated information suffices any item in the command table, the specific execution is called. The robot may need to confirm its interpretation before the execution

Although requesting more information is necessary and asking for confirmation reduces the risks of incorrect operations, they influence the naturalness and effectiveness of the interaction, which may decrease the satisfaction of the user if too many turns of conversation are required.

To reduce the cycles of interaction, our system tries to utilize the recognized information as far as possible. When possible, it tries to infer the missing information rather than simply requesting the user. Unlike the work by Harte [97], in which the whole utterance is ignored if the action word is not recognized, in our system the action word can be inferred if other classes, e.g. objects and directions, are recognized. It searches in the database to find what action is associated with this recognized component. If only one is associated, then the action can be determined with a simple confirmation. For example, the robot will ask "do you want me to *turn* left?", if the word "left" is correctly recognized while the action "turn" is inferred because there is no other verb related to "left". This inference is also affected by the state of the object or the robot. For example, if the robot is holding a cup, and it hears the user say "the cup" but misses the verb, it is most likely that the user wants the robot to put it down rather than pick it up.

It is also important that the robot reports its state to the user, such as "waiting for command", "working on the task", "task finished" or "task failed". Synthetic audio feedback is implemented using the synthetic speech engine in the Microsoft speech SDK.

6.3 Experiments

The evaluation was conducted in a laboratory-like office. The set-up includes many objects which are often used in a kitchen, office or living room, including cups, plates, bowls and boxes. Some objects were distinct in type or colour, while some identical objects were also included. The objects were placed at various locations such as on a small table, on the floor, or on a chair. The setting of the objects was similar to that in Chapter 5 (Figure 5-6, Figure 5-8), but slightly more crowded. The participants gave commands to the robot, such as "go there", "turn left", or requested the robot to bring some objects to them.

6.3.1 Experiments design

Some spoken descriptions of the objects, such as big, small, long and short, are not included in our experiments, although they are widely used in our daily language, because these words are subjective for each individual and relative to other objects in the scene. This could impose more difficulties on the system. The subjects were also advised not to refer to the usage of objects, such as "water bottle" and "coffee cup".

The user's identity was often recognized before the subsequent interactions, so that the user's profiles, including the self-defined hand gestures, pointing method preference and trained speech profile, were loaded. This improved the interpretation results of each sub-system, giving the multimodal fusion a good starting point. However, if the user is new to the robot, the default profile could be used.

To evaluate the multimodal interaction method and compare it with the unimodal speech input, three types of experiments were designed and conducted.

1. The subjects tried to communicate with the robot by speech only. Hypothetically, this would be easy for navigating tasks, but may become difficult for requesting certain objects, because we arranged some identical objects in the scene which could only be distinguished by their spatial locations.
2. The users chose the method of interaction in their natural way. A number of researchers have noticed that, although users prefer to interact multimodally rather than unimodally, they nonetheless do not issue every command multimodally [220, 221]. In this stage, the subjects were free to choose their methods to communicate with the robot naturally. They could still use speech alone, and they could also combine speech with gestures. In addition, as stated before, people tend to look at the target before and during the speech utterances and the pointing gestures are conducted. We tracked the head pose and eye gaze direction of some participants during their trials in the experiments.
3. The users were encouraged to use both speech and gestures to provide as much information as possible to the robot. They attempted to use pointing when referring to an object and also spoke the properties of the object, even if there was only one object of that type. The information provided can be redundant at this stage. The idea is to treat the robot like a young child. People do the same thing when they

meet some foreigners or small children in order to increase the chances of being understood. However, no strict restrictions were applied.

The robot tried to figure out the user's intention using the available information from all modes together with the pre-built database, including the user's profile, the objects' properties and the command table. If a task was determined, the robot confirmed with the user, for example by saying "Do you want me to pick up the cup whose ID is number 1". Note that this number was only used for the robot to confirm with the user and not was used by the user to request objects. This is because, for example, if there is only one cup, the robot can omit the ID number in its enquiry, while if there is more than one identical cup, using the ID number would be simple and direct. If the interpretation was correct, it was considered as a successful interaction. Indeed, it would be more natural and preferred if the robot could also point to the referent to confirm its interpretation as in [271], however, it is more effective to say the object's ID number when we only wanted to test the cognitive part without the real operation by the robot.

6.3.2 Results

During the experiments, when the robot determined that the commands were ambiguous, it would request the missing information by a synthetic voice. However, in order to evaluate and compare the effectiveness of the three interaction approaches, the recognition rates in the following sections were only calculated based on whether the commands could be correctly understood in the first cycle of the interactions.

6.3.2.1 Speech only

The ASR is supposed to work well under certain conditions, requiring a quiet environment, good quality input signal and extensive training. However, our experiments were conducted in an office environment, with colleagues working and a copy machine and an air conditioner operating in the background. Only one of the five subjects is from an English speaking country and only two persons spent short a time training the ASR, one person spending half an hour and another about 10 minutes.

Some spoken utterances by each subject were used to build the word-mapping database for each individual. This significantly improved the speech recognition. Of the testing trials by speech only, a total of 100 sentences were spoken, comprising 594

keywords. Only 296 words were correctly (with non-zero ASR confidence value) recognized by the original ASR, while 437 words were correctly recognized when the word-mapping database was applied. Therefore, the subsequent processing used the mapping results rather than the original results from the ASR. Note that the mapping method only works when the errors of the ASR in the training stage are consistent with those in the testing dataset.

In total, 48 out of the 100 spoken commands were successfully recognized using the word mapping method in the testing stage. In addition to inconsistent ASR errors, many failures were caused by the pauses and repetitions in long sentences. Unnecessary pauses separate one complete sentence into two. Repetition, hesitation and correction of mispronunciation, for example, "give me, *erm*…, that red, sorry, blue cup (*pause*) at the right corner of the table", could not be handled by our current system. Long sentences had a low probability of being understood. To interpret the spatial relationship between the intended target and other referents, like "the cup at the centre of the table", every keyword needed to be correctly recognized.

In addition, the user's command itself could be ambiguous. For instance, in seven trials, the user did not notice the existence of other identical object outside of his/her view when he/she was looking directly at the target. This shows that head pose and eye gaze could provide a good indication of where the user's attention is and has the potential to resolve ambiguity.

6.3.2.2 Interaction in a Natural Way

When the subjects were free to choose unimodal or multimodal methods at will, spoken sentences were much shorter in most cases, with few pauses and repetition. In total, 106 out of 151 commands were successfully recognized. Users normally selected the interaction methods that could eliminate linguistic complexities. The typical commands combined a speech utterance like "give me that cup" with a pointing gesture indicating the location of the desired cup. Our experiments confirms the finding in [219] that users show an obvious preference to interact multimodally rather than unimodally, especially in the spatial domain.

However, this preference does not guarantee that people issue every command multimodally. In most cases in our experiments, the action words and the colour/type

properties of the objects were spoken verbally, while the spatial information (the object's position) was usually indicated by pointing or gaze. However, due to the limitation of the faceLAB (see section 6.2.2), the movement of the participants needed to be constrained so that the attention direction could be captured. In the experiments, three participants volunteered to accept this constraint. They were required to restrict the body movement on the premise that they could still perform pointing gestures efficiently. In addition, if there was only one object of a type (e.g. only one box in the scene), the users might only use speech, because this property was sufficient to distinguish it from other objects.

In the trials on requesting an object, inputs of the three modalities were used together to calculate the PRS of the objects to be selected as in Equation (6.7). Overall, 96 out of 140 object-requesting commands were successfully recognized. The pointing gestures were critical to select the correct object among several identical ones, in situations where unimodal speech could not succeed. The attention direction lowered the chance of selection of the objects which were outside the user's view. The attention direction showed its advantage in some cases, as for example, when the user said "give me the red cup" without a pointing gesture since he only noticed the red cup in front of him and did not notice there was another red cup by his side. The spoken words were also important when the pointing vector and the attention direction were not able to select correctly from a crowded arrangement.

Since multiple modes tend to provide complementary, rather than redundant information, the loss of information by the independent recognizers is likely to lead to failures. Many failures were because some spoken keywords were not recognized and meanwhile the deictic gestures were not detected or were not accurate enough to enable selection of the correct target.

It was also noticed that the participants rarely used gestures for some navigation commands such as "turn left" and "go forward". This may be because they assumed the robot would be able to understand the speech phrases accurately.

6.3.2.3 Preferred Deliberate Multimodal Interaction

When the users were encouraged to provide as much information as possible, the system provided the best performance in the three types of experiments, as expected. 125 out of 148 trials were successfully recognized, which improved the recognition rate by 14%

compared to the trials when the participants expressed their intentions in their natural ways. Although the users' input information seemed redundant, however, due to the possible errors in the recognizers of each mode, some important information might be lost and the originally redundant information may have become complementary rather than redundant. Even if the recognized results from different information sources were really redundant, it would not affect the final decision adversely. However, the participants were only encouraged, not restricted, to interact multimodally. Some participants in fact did not use both speech and gestures in some trials.

Multimodal interaction showed great advantages in the object-requesting commands. For example, when the user was free to use their natural approach, shorter sentences such as "give me that bowl" were often preferred, whereas if the user was encouraged to provide more information, sentences such as "give me that red bowl in the centre of the table" were used. Both utterances were accompanied by a pointing gesture. If the pointing vector was not accurate, and the noun "bowl" was mis-heard, the word "red" could still greatly increase the chance of the robot picking out the correct object, if there were only one or two red objects. However, multimodal interaction did not show improvement for navigating commands when combining speech and hand gestures, because these commands were really short utterances, (such as "turn left/right") and already had a high recognition rate by speech recognition alone. However, the gestures would be helpful if the audio input degraded too much and speech recognition failed.

The experiments showed that the system generates better, or at least no worse, results when the user provides redundant information. It is particularly practical when the user wants to maximize recognition rate, but puts naturalness as the second priority.

6.4 Collaborative Work

Some preliminary collaborative work has been conducted toward our final goal of a household assistive mobile robot. Two kinds of collaborative experiments have extended the ability of the robot, especially in aspects of spatial intelligence such as path planning, object recognition and manipulation. The mobile robot implemented several types of tasks requested by users, including finding its path to different places, moving to a position specified by a pointing gesture, and recognizing an intended object and picking it up. All

the experiments were implemented online. Although the robot received the commands and started to respond quickly, the movement of the platform and the robot arm was slow.

6.4.1 HRI for Robot Navigation

Navigation of the robot is achieved by combining the multimodal interaction method in this research and the path planning system developed by Gupta and Jarvis [91, 92]. In our experiments, a large room was virtually divided into several cells representing the kitchen, the living room and the bed room. To successfully approach the destination, the robot needs to know its own location in the environment. A panoramic camera was mounted on the ceiling of the room facing vertically down with 360 degree azimuth and 90 degree zenith views (i.e., 180 degree field-of-view). It could observe the robot and the whole room at the same time enabling the robot to track its own position, plan the path to the destination and avoid static and dynamic obstacles. The advantage of using a panoramic camera is that a single camera can view the entire room. It makes the projection analysis much easier compared to using multiple cameras [92].

Figure 6-8: A snapshot from the panoramic camera. It demonstrates the experimental environment. The room is virtually divided into cells. The robot needed to find its path to the destination and avoid the obstacles. The image is from [92].

Two kinds of navigation commands were issued to and implemented by the robot. First, the user asked the robot to go to a place using speech utterances such as "go to the kitchen". The spot where the robot should stand after it entered the kitchen was predefined. In the experiments, the robot calculated its optimal path to the specified place in the presence of obstacles and successfully moved to the correct location. Second, the user asked the robot to adjust its position to a new location nearby. This time the commands were issued by the combination of a sentence ("go there") and a pointing gesture. It was necessary to transform the pointing vector from the local coordinate system which originated at the robot's eyes, to the global coordinate system which was determined by the panoramic camera. The robot could successfully move to the new location requested by the user.

Another type of commands was also designed to ask the robot to adjust its position by a small amount. The user said "move to the left by this much", accompanied with a two-handed gesture indicating how far the robot should move. This was, however, simulated rather than physically implemented by the robot, because the size of the robot is large and the accuracy of its position as observed from the panoramic camera, which depends on the robot's location, is sometimes not very accurate [92].

6.4.2 HRI for Object Manipulation

In the experiments described in Section 6.3, when the user requested an object, the robot identified the target based on the user's speech, gesture, gaze inputs and the object properties in the pre-built database. The positions of the objects were manually measured beforehand. The robot did not physically implement the object manipulation until the combination of our multimodal interaction system with the object recognition and grasping manipulation system developed by Effendi et al. [62, 63]. For the purpose of object manipulation, the robot is equipped with an arm manipulator with grasping hand (UMI RTX 100) and a set of stereo colour cameras [62]. Figure 6-9 shows the setup of one of the collaborative experiments. Due to the reachable range and the size of the robot grasping hand, the objects were placed near the edge of the table. It would be better if the above mentioned path planning system could also be integrated so that the robot could move freely to find objects at various locations, but the experiments reported in this sub-section were conducted without movement of the robot platform.

Three types of interaction were trialled. First, the user pointed at the target and the robot picked up the object nearest to the intersection of the pointing vector on the table. In this case the robot simply assumed the only task was to pick up an object whenever a valid pointing gesture occurred. Second, the user used unimodal speech input where the objects on the table were easy to be distinguished from each other in terms of type and/or colour. Third, the user combined speech utterances and pointing gestures for the cases where there were identical objects and they could not be selected easily by speech commands alone. Figure 6-9 shows that the pointing gesture was able to solve the ambiguity when two identical juice boxes were nearby on the table. Due to the inability of the robot to approach the user, the user went to the robot after the robot successfully grabbed the target and issued a "put down" speech command, and then the robot's hand released the object.

An important difference of these collaborative experiments from those described in section 6.3 is that the locations of the objects were not measured beforehand. The objects were detected using the Scale Invariant Feature Transform (SIFT) features [62] or 3D shape recovery method [63]. The objects were segmented from each other and their poses including position and orientation were calculated, so that the grabbing operation became possible. Object localization and segmentation were implemented using the stereo cameras mounted on the robot. The intersection of the pointing vector on the table was used to specify an area where the robot should look for objects. The objects in the specified region were extracted as shown in Figure 6-9 and their positions were calculated. Then the target was selected based on the method described in this chapter and picked up by the robot.

For object manipulation experiments, the robot needs to detect the positions of the objects online, even if they are measured in advance, because inaccurate measurement will cause the grasping operation to fail. However, it would be a good idea to save the properties of detected objects, such as type, colour, location, in a database. When the user says "bring my cup in the kitchen", the robot does not need to search the whole kitchen. In this way, the robot could find the requested object very effectively, particularly when the objects are mostly arranged by the robot and their new positions in the database are updated every time they are moved.

Figure 6-9: Human robot interaction and object manipulation. This experiment demonstrated that the robot correctly interpret the user's command requesting a specified object. The robot could resolve the ambiguity which arose from the presence of two identical objects.

6.5 Discussion and Conclusion

This chapter has described our speech recognition component and the integration of it with the gestures recognition and attention vectors estimation sub-systems discussed in the previous chapters. This multimodal interaction system uses a probabilistic fusion method to find the most likely instance of action, direction and object. It determines whether the accumulated information is complete to perform a task according to a predefined command table. The robot reports its status to the user and can also request

more information from the user in a two-way interaction pattern. Experiments show that users prefer to interact multimodally, and our multimodal interpretation system significantly outperforms unimodal speech input. Furthermore, the proposed probabilistic framework for multimodal information fusion is flexible and can be expand to include more input modes.

During the experiments, it was noticed that type and colour are the properties that are most frequently used by the participants. However, the feedback from some participants indicated that shape, size and usage properties are preferred in natural communication, but currently not included in our system. A more sophisticated speech interpretation method appears to be needed to deal with the ASR errors and the variety of spoken utterances. For example, when people say “give me the orange juice”, they actually mean the container of juice. Another interesting situation mentioned in [243] and [180] is that the robot should also consider the human’s perspective. An object cannot be requested by the user if it is occluded by another object from his/her perspective. Currently, we only consider whether the objects are in or outside the user’s field of view. The size of the object, the relative position between different objects and the user’s perspective could be considered together.

Because the faceLAB has a narrow field of view and a shallow depth of field, it frequently lost tracking of the person’s head if the subject freely moved around or turned his/her head too much. Constraints were imposed on some participants to restrict limited body movement due to the limitation of the faceLAB when we wanted to capture and investigate the attention direction. If the faceLAB was equipped with a tilt-pan platform controlled to adjust its own position and angle according to the face being tracked, this would diminish the constraints on body movement. A system with an active pan-tilt-zoom camera for tracking eyes has been proposed by Reale et al. [238]. Gaze depth estimation is helpful to locate the spot where the user’s attention is more precisely if there is a more than one object along the gaze direction. The measurement of the pupil centre distance can be used to find the gaze depth as in [144].

7

Recognizing Emotional Gestures

Neurological studies have demonstrated that emotions have a significant influence on decision-making, problem solving, cognition and intelligence [54]. Emotions have been researched in various scientific disciplines including neuroscience, psychology and cognitive science [89].

Humans convey and recognize emotions from a broad spectrum of verbal and non-verbal modalities [18]. As Picard pointed out in [228], the ways in which humans convey emotional messages in general are influenced by many factors, such as age, gender, personality, physical body characteristics, culture, context and so on. Therefore, each individual has his/her own way of expressing emotional feelings in different scenarios.

During interpersonal communication, people can often easily sense the changes in others' emotional states, from their facial appearance, tone of voice and body language. Humans' emotional states provide essential information which implies their intentions, and can be used to predict their next actions. Since domestic robots are expected to act as social companions, the capacity to recognize human's emotional states has become a critical feature, especially for the assistive robot's development in support of its social role. Therefore, recognition of human's emotions has recently been an attractive research topic in the domestic robotics research field. More importantly, based on the interpretation of the user's emotions, the robot may also initiate helpful interactions rather than simply respond to human's commands.

In addition to the robotics research field, automatic recognition of affective behaviours has a wide range of applications. For example, it can be used for video surveillance in hospitals to detect patients' negative emotions and offensive behaviours. Patients who express negative emotions for a long time could be given more care; the long

duration of pain may also be reflected through the facial/gestural expressions of a patient. Other examples include the automatic assessment of students' boredom in classrooms and workers' fatigue in workshops where operational hazards are often encountered.

This chapter is organized as follows. After a review of the related work in Section 2, our SVMs-based classification method to recognize a person's emotional states through body gestures is presented in Section 3. In Section 4, the experimental results show that the proposed method has a good performance in general and hand gestures can make a positive contribution to the recognition results.

7.1 Related Work

A person's inner emotional state reflects one's subjective feelings, and will incur internal bio-signal changes and be expressed externally by audio/visual features. In the work by Picard [228], the user's emotional states were sensed through physiological signals such as heart beat and galvanic skin response. Wang et al. [284] built a chat system which also used galvanic skin response sensor to measure the user's internal physiological signals. However, the disadvantage of this kind of systems is that the user must wear sensors, which are difficult to use naturally and on a daily basis.

Humans' emotions are often expressed through speech utterances, not only by the content but also the tone of the voice. Cowie et al. [53] developed an instrument to examine the emotional dynamics of speech episodes based on the activation-evaluation space. The work by Grandjean et al. [82] also tried to judge emotional states on vocal expression using the appraisal-base method [67]. Morris et al. [201] reported that in a social interaction scenario, facial expressions, the tone of voice and gestures are more meaningful than verbal content, especially in demonstrating changes in emotional states.

The use of gestures can be just as or more effective than speech [142]. The experiments by Mehrabian and Friar [187] showed that changes in a person's emotional states were reflected by changes in her/his body posture and eye contact. It is also demonstrated in [65, 188] that humans could display their emotions effectively through facial expressions and body gestures.

The recognition of facial expressions has been investigated in depth [229, 245] in the Human Computer Interaction (HCI) community. Ekman and Friesen developed the Facial

Action Coding System (FACS) [66] for describing emotional facial expressions. It has been the most popular database for the description of emotions through facial features. FACS is a comprehensive and anatomically-based system to measure facial movements. 44 face action units (FAUs) are defined which are anatomically related to the contraction of specific face muscles. Other famous facial expression databases include [135, 136].

In addition to facial expressions, choreographic studies [146] have shown that expressive body gestures are a powerful and frequently used method of emotional communication. Coulson [52] concluded from his experiments that human recognition of emotions from postures was comparable to recognition from the voice and the facial expression. However, there are as yet no standard models like FACS, for describing and classifying emotional body gestures from low-level features.

Emotional body gestures can be measured by various means in different scenarios. The work by Kapoor et al. [139] attempted to classify a child's different levels of interest into 3 categories (high interest, low interest, and refreshing) according to the postures of the child. The postures were detected through pressure sensors embedded in a chair when the child was using a computer to solve a puzzle. Similarly, Silva et al. [263] created a gesture recognition system that recognized a child's emotional states with intensity levels through his/her body gestures in a computer game situation. A motion capture system was used with eight markers attached on the child's upper body parts. Abbasi [1] proposed a Bayesian Network approach to relate subjects' gestures to the most probable emotional state in a given situational context of emotional tutoring, which was to understand the emotional response of students attending a lecture. The system used a skin colour-based segmentation method. The relative position of the gesturing hand from the head is extracted to distinguish three states including thinking, recalling and tired.

In non-verbal dance performances, gestures are not intended to support speech as traditional natural gestures, but the information conveyed by the gestures is mainly related to the emotional domain [41]. In the work by Woo et al. [290], the emotions conveyed in dance performances were modelled according to three types of features: the amount of space occupied by the body, the acceleration of the motion and its occurrence time. The gestures were mapped into seven emotion categories (natural, happy, fluent, lonely, lively, sharp, and solemn) by a Neural Network method. Camurri et al. [41] proposed a unifying layered conceptual framework to recognize four basic emotions (anger, fear, grief and joy).

Analysis was implemented by four layers of processing, computing from low-level physical measurements to obtain high-level semantic information. A number of motion features were extracted using vision-based methods. Instead of tracking particular body parts, a set of 40 points randomly distributed on the whole body was tracked by the Optical Flow method. The motion cues, such as quantity of motion, contraction index, motion trajectories, kinematic cues (velocity and acceleration) of motion trajectories, motion fluency and impulsiveness were evaluated. Decision tree models were built as the classification tool. Their results showed that the rate of correct automatic classification (35.6%) is between chance level (25%) and the rate of correct labelling by human observers (56%). In their experiments, fear, often classified as anger, was the emotion receiving the lowest rate of correct classification. This same phenomenon was also observed in the spectators' ratings.

Following the multilayered framework proposed by Camurri et al. [41], Glowinski et al. [76] found the low-level motion features (i.e. 3D positions and kinematics of the hands and head) in a frontal and a lateral view, then five sets of high-level qualitative features (including energy, spatial extent, smoothness, forward-backward leaning of the head, spatial symmetry of the hands) were computed. PCA was used to reduce the dimensionality of the feature data, from 25D to 4D. Twelve emotions expressed by ten actors were classified using the 4D model according to their valence (positive, negative) and arousal (high, low) level.

Although dance is a typical source for the study of emotional gestures, the gestures involved may be exaggerated and would not naturally be performed in daily life. The active research group led by Bianchi-Berthouze [20, 147, 148, 262] at the University College London Interaction Centre (UCLIC) has focused on investigating which low-level features of body postures provide the information necessary to recognize emotion states. They collected the 3D motion data of the participants with a VICON motion capture system [281]. Each actor was dressed in a data suit with 34 markers on the joints and body segments. To obtain the 3D spatial position of each marker, eight cameras were used to capture in consecutive frames. In [20], an associative Neural Network called CALM (Categorizing and Learning Module) [205] was used. CALM incorporates brain-like structural and functional constraints such as modularity and organization with excitatory and inhibitory connections [20]. They selected the key frames which were considered to be the most expressive instant of the gestures. Posture data were projected on the three-

dimensional orthogonal planes and extensions of the body along lateral, frontal and vertical axes were computed. Gestures were described using 18 to 24 feature vectors based on the concept of sphere of movement [153], including rotation angles of the head and a set of distances between key anatomical landmarks. In [147], they extended their work by combining static postures with motion features such as dynamics of the hands, shoulders and head. Paterson [225] also considered speed a very discriminative feature. A low speed motion is often associated with sadness or tiredness, while a high speed motion is associated with excitement. The CALM network was used in [147] for classifying input gesture data into four basic emotional categories (angry, fear, happy and sad). They compared the evaluation ability of the system with human observers. Since the features captured by the VICON system do not include facial expression, for the purpose of a fair comparison, a set of avatars, with main body parts except for the face, were built for the selected postures. Human observers, 143 Japanese students, participated in labelling the emotion of the avatars. The results showed that the recognition rate of the observers (69%) was even lower than their system (79%) for the four emotional categories.

To examine if the feature set could identify nuance within a same emotion class, Silva et al [262] applied Mixed Discriminant Analysis (MDA) to the descriptors of the postures.. They grounded nuances of emotional states on low level features of expressive postures using an unsupervised method. The number of the clusters to be identified within each category was chosen by the unsupervised EM clustering algorithm. Using an iterative process, the MDA algorithm creates the model based on linear combination of the most discriminating features. They noticed in their experiments that the most discriminant feature was the bending of the head, followed by the lateral opening of the elbow with respect to the shoulder, and the vertical extension of the arm with respect to the shoulder.

In [148], Kleinsmith et al. identified the emotional dimensions that humans use to discriminate between postures. In their work, they aimed to ground emotional gestures in the way that exploits the same emotional dimensions used by human observers. 40 observers were asked to choose a label from an eight-emotion list and rate the intensity of the emotion on a scale from 1 to 5. Three dimensional emotional spaces were obtained by applying multidimensional scaling. Dimension 1, which is called arousal, separates the categories of sad, depressed and upset from the others (angry, fear, happy, joyful, and surprised). These two sets of emotions identify two opposite levels of arousal (low in the first case and high in the second case). The second dimension (valence) separates the

negative emotions (fear/surprise) from the positive emotions (happy/joyful). The third dimension represents action tendency. It separates the fear/surprise emotions from anger. The latter emotional state represents a desire and readiness to react, while the former represents more a sense of protection. The plane defined by dimension 1 (arousal) and dimension 3 (action tendency) approximates the emotion wheel of Plutchik [229]. In order to ground the dimensions of the emotional space onto 24 postural features, they applied the MDA method to each of the three emotional dimensions. MDA builds a model based on a linear combination of the most discriminating features. The rank of discriminative power of the features in each space was also sorted.

The above mentioned work by Bianchi-Berthouze et al. [20, 147, 148, 262], have demonstrated that expressive body gestures can be employed to infer a person's emotional states, but they also point out that existing ambiguity could be potentially reduced by combining other modalities, such as facial expression and voice. Ambady and Rosenthal [8] reported that humans judge behaviours most accurately when they observe both the face and the body. Their experiments proved that joint face and body recognition is 35% more accurate than those based on face alone.

Gunes and Piccardi built the first bimodal face and body gesture database (FABO) [88] for the analysis of non-verbal emotional behaviours. The experiments were recorded by two cameras in a simple setup with uniform background. One camera captured the face region only and the second camera captured the upper body movements. This database is publicly available and widely cited [43, 143, 254]. Based on the recorded face and body gestures, Gunes and Piccardi [85, 86] combined these two modalities to recognize the emotions expressed by the subjects into six categories (i.e. anxiety, fear, anger, happiness, disgust and uncertainty). The feature vectors consisted of displacement measurements between the neutral frame and the expressive frame in which the facial and/or gestural expression was at its apex. To achieve automatic recognition, dedicated vision techniques were adopted to detect face features and segment and track body parts. They compared the recognition results of using unimodal face data and bimodal features of face and body gesture, and concluded that bimodal data achieved better recognition accuracy in general. In [85], two methods of fusing the bimodal data, both feature-level and decision-level, were implemented. Feature-level fusion was performed using the extracted features from each modality and concatenating these features into one large vector. For decision-level integration, each modality was first classified independently and the fusion was based on

the outputs from different modalities. Their experimental results showed that the feature-level fusion method achieved a better recognition accuracy compared to the decision-level fusion. Then they further improved the result by post integration of a whole image sequence. Utilizing the FABO database, Shan et al. [254] used the Canonical Correlation Analysis (CCA) method to fuse facial expressions and body gestures at the feature level and their results proved that the bimodal approach outperformed the unimodal approaches.

7.2 Emotional Gesture Recognition

Figure 7-1 shows the overview of our emotional gesture recognition system. In the training stage, 3D positions of several selected body joints (and hand shapes) can be used to construct the expressive feature vectors. The Linear Discriminant Analysis (LDA) method finds the most discriminant vectors, and the original data are projected to the low-dimension space spanned by these discriminant vectors. The projected vectors are then fed into the Support Vector Machines (SVMs)-based classification tool. The iterative Nelder-Mead nonlinear optimization method is used to search for the optimal parameters. In each iteration, the cross-validation technique is employed so that the limited scale of available data can be utilized to train SVMs models effectively and avoid the overfit problem. In the testing stage, the emotional categories are determined directly when a pre-defined hand gesture is recognized; otherwise the testing samples go through the same pre-processing procedures first and are then classified by the trained SVMs model.

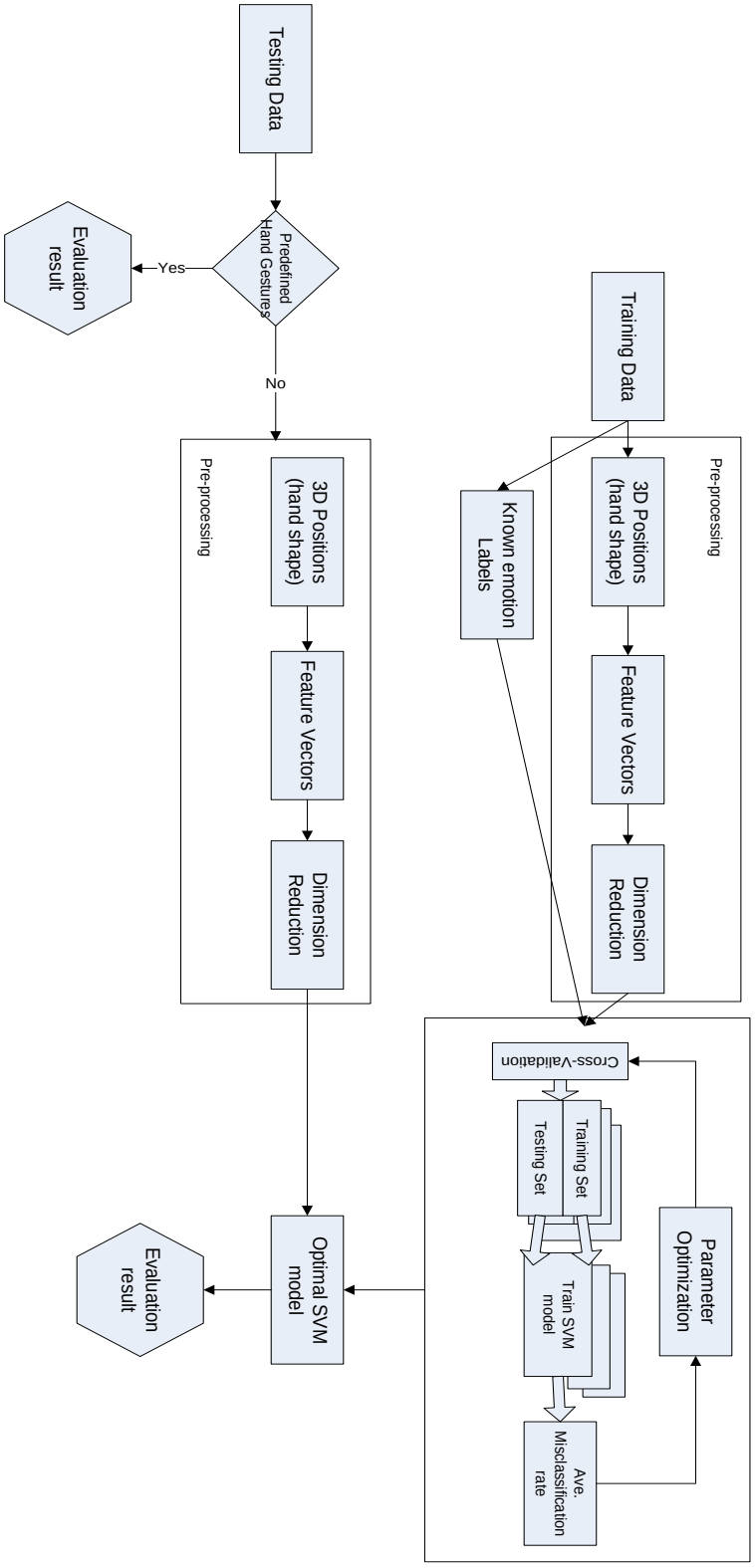


Figure 7-1: Overview of the emotional gesture recognition system.

7.2.1 Feature Vectors

7.2.1.1 UCLIC Emotional Body Posture and Motion Database

The UCLIC emotional body posture and motion database [19] is a well constructed and publicly available database, which contains acted body motions by 13 non-professional actors portraying four kinds of emotions: angry, fearful, happy and sad. 3D position data was collected by the VICON motion capture system [281]. Eight cameras were used to capture the 3D positions of 34 markers which were attached on the major joints and body segments. The most expressive frames are selected and labelled as one of the four emotional categories. Using the recorded data, face-less avatars can be built, as shown in Figure 7-2. These avatars visualize the expressive postures and can also be used to compare the evaluation performance of the recognition system with human observers. Some of the 3D positions of the body joints in the recorded database were selected to construct a 38 dimensional feature vectors as listed in Table 7-1.

Some of these vectors (e.g. arm extension, relative hands-face and hands-waist positions) are constructed based on the observation through the database, while others are based on statistical study results. For example, a statistical study showed that the head inclination and rotation are very important in discriminating between emotions [261] and between nuances of a given emotion [262]. As suggested by [148, 262], the body part extensions are calculated in three axes. Each feature is calculated by the ratio of the actual extension of the body parts and maximal length of his/her limb in the certain directions, so that the calculation is independent of individual differences in limb length. Body rotation is calculated based on the relative positions of the two shoulders.

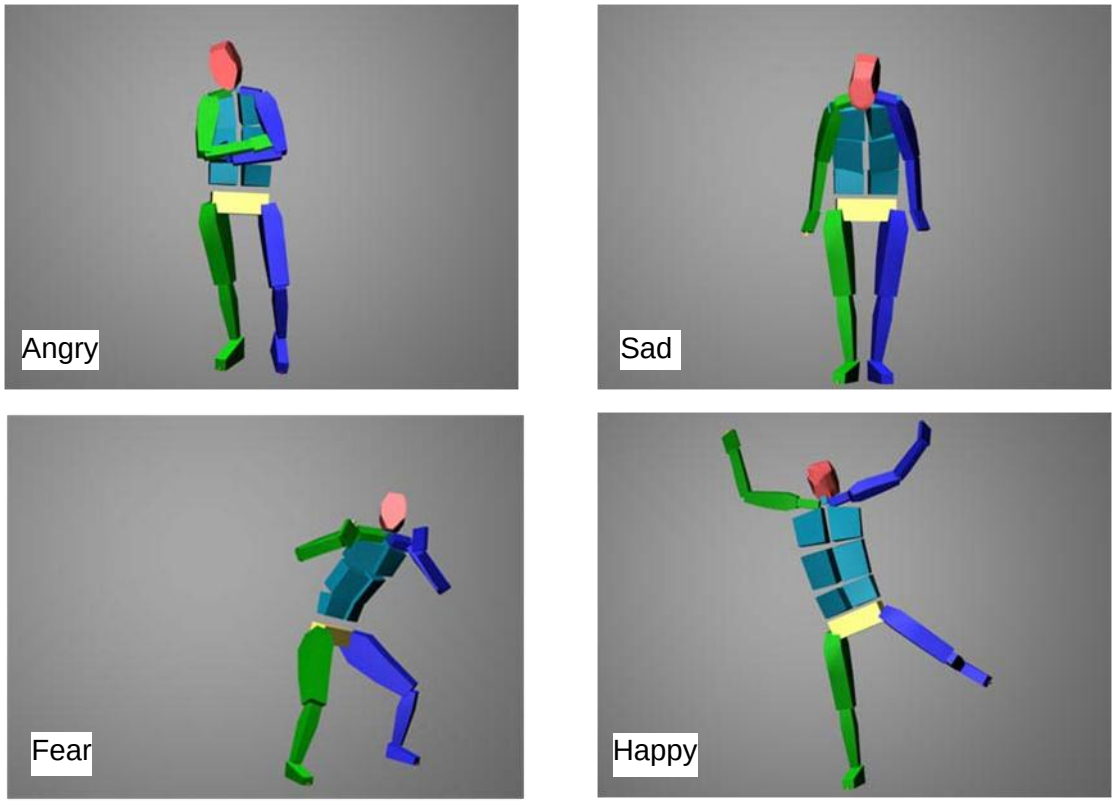


Figure 7-2: Examples of the expressive avatars which are built using the captured 3D data. Pictures are obtained from the UCLIC emotional body posture and motion database [19]

Table 7-1: The feature vector components and their descriptions

Component	Description	Component	Description
V1	Lateral inclination of the head	V2	Forward-backward inclination of the head
V3~V8	Right hand-shoulder and Left hand-shoulder extension along 3 axes	V9~V14	Right hand-elbow and Left hand-elbow extension along 3 axes
V15~V20	Right elbow-shoulder and Left elbow-shoulder extension along 3 axes	V21~V22	Two feet distance along lateral and frontal axes
V23~V24	Feet height	V25~V30	Hands-face distance along 3axes
V31~V36	Hands-waist distance along 3 axes	V37~V38	Body rotations along vertical and frontal axes

7.2.1.2 Joints Positions and Hand Gestures

Kleinsmith et al. mentioned in [147] that the configuration of the hands and fingers might provide potentially useful information for emotional gesture recognition. Bianchi-Berthouze and Kleinsmith [20] also noticed that in their experiments some postures were misclassified due to the lack of features. For example, some 'angry' postures shared the same features with both 'happy' and 'sad' postures with their main differences concerning the shape of the hand-open versus closed.

A hierarchical method is proposed to include hand features into the recognition of emotional states. Utilizing the hand gesture recognition subsystem proposed in Chapter 4, predefined meaningful hand shapes can be recognized. Since these hand gestures are normally performed more deliberately and explicitly, high priorities are assigned to the predefined hand postures which are associated with certain emotions. For example, when the 'victory' finger configuration or the thumb up hand shape appears, the person's emotional state is decided as 'happy' directly. Nevertheless, some other hand shape attributes like 'open' or 'closed' may be expressive but cannot be used alone to determine a particular emotion. Therefore, a new variable v = {0, 1, 2} (indicating no hand shape used (0), open (1), closed (2), respectively) is added into the posture feature vector listed in Table 7-1. Since this kind of open/closed attribute is not easy to detect automatically, the value of the variable v is assigned manually for expressive key frames.

Here, the video and the depth data is captured by the Kinect and 3D positions of the body joints are extracted by its SDK program [192]. A model-based pose estimation method was initially proposed to track the body joints of the user. The method searches for the best match of the observation and the human shape model using silhouette and depth information. The model is built by the superquadrics technique [10], and the joint angles of the model is estimated by the Particle Filter method and a gradient-based method. Unlike the multi-view approaches, the person in the experiments was only observed from the robot's view. The experiments showed that the method could correctly recover the configuration of the upper body, and it was not sensitive to noise and self-occlusion. The work was reported in [164]. The model-based method could also be used for the purpose of hand tracking and forearm direction estimation. However, the method is slow for real time applications for two reasons. First, it had nine DOFs for the upper body which causes a large search space for the parameters. Second, the built-in function used to retrieve the

depth information from the model that was built in OpenGL is slow. Therefore, the model-based method proposed in [164] was not applied because of the real-time requirement of the robotics application.

Following the scenario approach in [88], the actors were provided with situation vignettes or short scenarios describing an emotion eliciting situation. They were asked, for instance, what gesture they would use when "they won a contest or lottery" or "their lovely puppy is missing" and so on. For the moment, we only focus on the four basic emotion categories: happy, sad, fear and angry. Some of the expressive postures are shown in Figure 7-3. The Kinect SDK program extracted the positions of the body joints using captured depth images. These 3D positions are projected onto colour images and plotted as green dots in Figure 7-3. Similarly, the most expressive features are picked out for the recognition. In our experiments, expressive body gestures and hand shapes appeared simultaneously in most cases (if the hand shape is used in the vignettes).

Figure 7-3: Examples of emotional gestures captured by the Kinect. Note that the 'victory' hand gesture and closed fist are expressive in the 'happy' and 'angry' cases.

7.2.1.3 Dimension Reduction

The high-dimensional feature vectors can cause the subsequent training procedures of the SVMs classifier to be time consuming. The dimension of the features is reduced by projecting them to a new low-dimension space which is spanned by the most discriminant vectors. Linear Discriminant Analysis (LDA) [185] and Principal Component Analysis (PCA) [126] are commonly used for the purpose of dimension reduction. LDA attempts to model the difference between the data in classes. PCA, on the other hand, considers the whole set of data and does not distinguish the differences between classes. Therefore, the LDA method is adopted to reduce the dimension of the original feature vectors.

For multi-class LDA, the most commonly used method is to find the vector such that the ratio of the between-class variance to the within-class variance is maximized. It can be expressed as $C = \frac{d^T B d}{d^T W d}$, where B is the between-class covariance matrix and W is the within-class covariance matrix and d is the vector to be determined. The most discriminant vectors are the eignevectors of the matrix $W^{-1}B$ associated with the top eigenvalues. Following [76], the number of the eigenvectors (i.e. the dimension of the projected data) is chosen as four in our implementation. Finally, the original features are projected to the 4D space generating new low-dimension feature vectors.

7.2.2 Classification Approach

7.2.2.1 Support Vector Machines

SVMs deliver state-of-the-art performance in many applications such as text categorisation, image classification, hand-written character recognition and so on., and are established as one of the standard classification tools for machine learning and data mining [257]. SVMs are optimal classification methods based on the Bayesian learning theory [254]. Therefore, the SVMs approach is used to classify the feature vectors into different emotional categories. It is a type of supervised learning approach; a model is trained and tuned before it can be applied to classify test samples. For completeness, the SVMs algorithm is briefly introduced here. More details can be found in [38, 51, 101].

In a two-class classification problem, a SVMs model classifies data by finding the best hyperplane with the maximal margin to separate data points of one class from those of the other class. Given a training set of labelled samples $\{(x_i, y_i), i=1 \ldots N\}$, where $x_i \in R^n$ and $y_i \in \{+1, -1\}$, the best separating hyperplane is solved by searching for w and b which minimize $||w||$, subject to $y_i \cdot (w \cdot x_i - b) \geq 1$. However, the data may not allow for a simple separating hyperplane. In this case, SVMs methods use a soft margin method which means the hyperplane separates as many data as possible under certain criteria, but not all of the data points. One of the formulations of soft margins is [51]

$$\min\left(\frac{1}{2}< w, w > +C\sum_i s_i\right), \text{subject to}$$
$$\begin{cases} y_i \cdot (< w, x_{i>+b}) \geq 1 - s_i \\ s_i \geq 0 \end{cases} \tag{7.1}$$

This equation can be solved by the Lagrange multipliers method. Using Lagrange multipliers a_i, the function to minimize becomes

$$Lp = \frac{1}{2}< w, w > +C\sum_i s_i - \sum_i a_i \left(y_i(< w, x > +b) - (1 - s_i)\right) - \sum_i \mu_i s_i \tag{7.2}$$

Setting the deviation of *Lp* to 0, gives a set of equations

$$\begin{cases} b = \sum_i a_i y_i x_i \\ \sum_i a_i y_i = 0 \\ a_i = C - \mu_i \\ a_i, \mu_i, s_i \geq 0 \end{cases} \tag{7.3}$$

These equations lead to

$$\max_a \sum_i a_i - \frac{1}{2}\sum_i \sum_j a_i a_j y_i y_j < x_i, x_j >, \text{subject to}$$
$$\begin{cases} \sum_i y_i a_i = 0 \\ 0 \leq a_i \leq C \end{cases} \tag{7.4}$$

When the original data are not linearly separable, the data can be mapped into a richer feature space ($x \mapsto \Phi(x)$), then a separating hyperplane can be constructed in that space. However, the dimensionality of $\Phi(x)$ can be very large, making *w* hard to represent explicitly in memory. Kernel functions can be used to retain the simplicity of the SVMs models. There are different types of kernel functions, e.g. polynomials, Radial Basis Function (RBF) and multilayer perceptron. In our implementation, RBF kernel function

$$K(x, y) = \exp(-\frac{\| x - y \|^2}{2\sigma^2}) \tag{7.5}$$

is used, as suggested by [109]. In our implementation, an available MATLAB toolbox is adopted [175].

For the multi-class classification problem, a pairwise classification scheme can be used, which composes all possible two-class comparisons. In the pairwise scheme, with L classes, $L(L\text{-}1)/2$ SVMs need to be trained and in the classifying step for one test sample, L-1 comparison are performed in a bottom-up tree structure as shown in Figure 7-4.

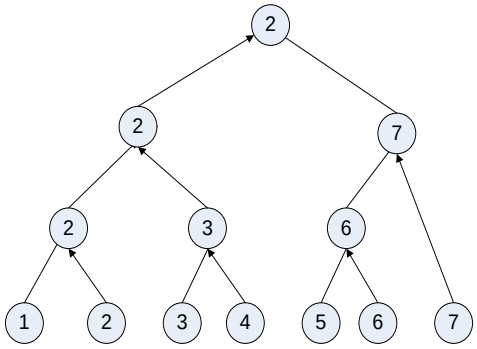


Figure 7-4: Bottom-up multi-class SVMs structure for the pairwise scheme. With 7 classes, 6 comparisons need to be performed to obtain a classification result.

7.2.2.2 Nonlinear Optimization

The most important parameters for training SVMs models are the box-constraint (C in Equation (7.4)) and σ in the RBF kernel function (Equation (7.5)). The Nelder-Mead simplex algorithm [208] is adopted which is a direct searching method for multidimensional unconstrained minimization as described in [156] to search for the best values for the parameters C and σ with minimal misclassification errors. This algorithm uses a simplex of $n + 1$ points for n-dimensional vectors x. The algorithm first makes a simplex around the initial guess x_0 by adding 5% of each component $x_0(i)$ to x_0 and using these n vectors as elements of the simplex in addition to x_0. Let $x(i)$, $i = 1,...,n+1$, denote the list of points in the current simplex, then the implementation of the iterative procedure of the algorithm is summarized as follows based on [156] and [181]. Here, $x = [C, \sigma]$ is a 2D vector and $f(x)$ denotes the average misclassification error by the cross-validation method (see Section 7.3.2.3) applied to the training data.

Algorithm: the implementation of the Nelder-Mead algorithm.

1. ***Order***. Order the points in the simplex from lowest function value $f(x(1))$ to highest $f(x(n+1))$ to satisfy $f(x(1)) \le f(x(2)) < f(x(3)) \le \cdots \le f(x(n+1))$. At each iteration, the algorithm replaces the current worst point $x(n+1)$ with another point into the simplex, or it replace all n points with values above $f(x(1)$ as in the case of step 5.

2. ***Reflect***. Generate the reflection point

$$r = 2m - x(n+1) \tag{7.6}$$

where $m = \sum_{i=1}^{n} x(i)/n$ and calculate $f(r)$. If $f(x(1)) \leqslant f(r) < f(x(n))$, accept r and terminate this iteration.

3. ***Expand***. If $f(r) < f(x(1))$, calculate the expansion point s

$$s = m + 2(m - x(n+1)) \tag{7.7}$$

and calculate $f(s)$. If $f(s) < f(r)$, accept s and terminate the iteration; otherwise, accept r and terminate the iteration.

4. ***Contract***. If $f(r) \geqslant f(x(n))$, perform a *contraction* between m and the better of $x(n+1)$ and r:

   a. Outside. If $f(r) < f(x(n+1))$, calculate

$$c = m + (r - m)/2 \tag{7.8}$$

   and calculate $f(c)$. If $f(c) < f(r)$, accept c and terminate the iteration. Otherwise, go to step 5.

   b. Inside. If $f(r) \geqslant f(x(n+1))$, calculate

$$cc = m + (x(n+1) - m)/2 \tag{7.9}$$

   and evaluate $f(cc)$. If $f(cc) < f(x(n+1))$, accept cc and Otherwise, go to step 5.

5. ***Shrink***. Evaluate f at n points $v(i) = x(1) + (x(i) - x(1))/2$, $i = 2,...,n+1$. The simplex at the next iteration is $x(1), v(2),...,v(n+1)$. Go to step 1.

The function *fminsearch* in MATLAB implements this algorithm and can be used. However, the limitation of this algorithm is that the searching may end at a local minimal, depending on the start point x_0. To increase the chance of obtaining the global minimal, the method used here is to randomly generate several start points, and compare among these local minimal values.

7.2.2.3 Cross-Validation

To make the most use of the limited number of training data, the cross-validation method [99] is used here. Cross-validation is a generally applicable way to predict the performance of a parameterized model on a generalized validation set, especially when the size of training dataset is small. One round of cross-validation involves partitioning the whole set of data into complementary subsets, building the model on the training set, and validating the model on the testing set. Multiple rounds using different partitions are proceeded and the validation results are averaged over the rounds.

The original sample set is randomly partitioned into N subsets. A single subset is retained as the validation data for testing the model, and the remaining N-1 subsets are used as the training data to build a SVMs model. The partition, training and testing steps are repeated N times, with each of the N subsets used once as the validation set. The N results are then averaged as the final estimation result. Considering the size of the dataset, 10-fold cross-validation is used for the UCLIC database and 3-fold for our captured data.

7.3 Experiments

7.3.1 Using UCLIC Database

As stated above, in the UCLIC emotional body posture and motion database, 183 postures are performed by 13 different non-professional actors conveying emotions in four categories including Happy, Angry, Sad and Fearful. The most expressive frames are picked out and labelled by the actors themselves. To evaluate the proposed method, all the samples were randomly divided into training and testing sets for ten iterations, with 90% of the data used for training purposes and 10% for testing and an average value of the ten classification results was calculated.

Using the training data, the Nelder-Mead simplex algorithm with the cross-validation analysis was used to find the optimal parameters in the kernel function, and then the whole training dataset was used to build a SVMs model. The testing data were then classified by the fitted model. The evaluation results were compared with the known labels of the test data. The classification results of the ten runs were recorded and are shown in a confusion matrix in Table 7-2. On average, the correct classification rate was 76.9%.

Table 7-2: The confusion matrix of the emotional gesture classification results using UCLIC database

Emotions Labelled by actors		Emotion evaluated by the system			
		Angry	Fearful	Happy	Sad
	Angry	40 (75%)	2 (4%)	11 (21%)	0 (0%)
	Fearful	1 (2%)	35(81%)	4 (9)	3 (7%)
	Happy	9 (18%)	2 (4%)	38 (77%)	0 (0%)
	Sad	4 (1%)	6 (14%)	1 (2%)	30 (73%)

In the confusion matrix, it can be seen that 'angry' was often classified as 'happy', and vice versa. This can be explained based on the observation of the avatars corresponding to the posture data. Some 'angry' postures with high arousal level (e.g. extended arms) were quite similar to the 'happy' postures in the face-less avatars, which were even difficult for human observers to decide. This also proved that other modalities such as hand gestures, facial expressions or audio clues are critical when the body gestures are ambiguous.

7.3.2 Using Joints Data and Hand Gestures

Using the 3D data of joints positions captured and tracked by the Kinect SDK program and the hand gesture recognition system, we had similar results, but the misclassifications between the 'angry' and 'happy' categories were fewer, as shown in Table 7-3. This supports the hypothesis that the inclusion of the hand gestures, such as the 'victory' finger configuration, and open/closed hand shape attributes have favourable effects.

However, our captured data set is much smaller than the UCLIC database, and the result in Table 7-3 should be interpreted qualitatively rather than quantitatively. More trails need to be done with more participants to increase the variety of the expressions.

Table 7-3: Confusion matrix of the emotional gesture classification results using the motion data captured by the Kinect and the hand gesture recognition system

Emotions Labelled by actors		Emotion evaluated by the system			
		Angry	Fearful	Happy	Sad
	Angry	16 (84%)	2 (10%)	1 (5%)	0 (0%)
	Fearful	2 (9%)	17 (81%)	0 (0%)	2 (9%)
	Happy	1 (5%)	1 (5%)	16 (89%)	0 (0%)
	Sad	0	2 (20%)	0	8 (80%)

7.4 Conclusion and Future Work

In this chapter, an emotional gesture recognition system has been presented. The processing results showed that the proposed method has a good performance in general and the inclusion of hand shape could add an additional positive contribution to the classification results.

However, the performances of non-professional actors, which are being recorded in a laboratory setting, can be often far from natural. As pointed out in [88], it is difficult to produce particular expressions 'artificially' without feeling the emotion that causes the expression. Moreover, it is difficult to change from one expression to another in a short time. Without professional training few people can perform these actions deliberately.

Our current system for recognizing emotional gestures is still at a preliminary stage and runs offline. To achieve a real time online system, in the future, the critical component to be changed is that the expressive frames in the captured video/depth image sequences need to be detected automatically. One possible method could be to use the motion segmentation method based on the motion energy as in [41]. However, false positives may occur, because not all of the gestures at the apex would necessarily express an emotion. A multimodal method combining gestures with facial expressions may be useful to reduce the number of false positives.

Furthermore, in our daily life, people often judge a person's emotional states through long-term observation. A single posture is often not sufficient to make a correct judgement.

Affects such as boredom, tiredness, stress, anxiety and frustration, normally need longer observation before they can be correctly determined.

Currently we have only considered four discrete emotion categories. However, people use a much richer and more powerful vocabulary for defining emotional states. For example, Whissell [287] listed 109 words for describing one's emotional states and Plutchik listed 142 in [230]. Very few papers report experiments about more complex states, such as frustration [138].

In future work, other classification methods, such *k*-Nearest Neighbour, Neural Networks, Decision Trees, Bayesian Networks, Hidden Markov Models and so on, will be applied and their performances will be compared with the SVMs method. Although SVMs have been proved to provide overall good classification result in various applications [120, 171, 294, 301], Colas and Brazdil [49] pointed out that SVMs is not a clear winner in their text classification tasks. In general, the performance of a classification method is affected by many factors including data distribution, feature selection, parameter optimization.

Like other behaviours, emotional gestures can differ between individuals. Users may have different preferences to express their emotions during the communication with a robot. This kind of difference, like other features, can also be included in the user's profile. The robot can recognize the person using the user identification method as mentioned in Chapter 3 and take his/her preference into consideration when it interprets the emotional gestures.

Emotional expressions are also often context-dependent. For example, frowning can be a display of deep thought or anger. To interpret a gestural signal it is important to know the context in which the signal is displayed, including the expresser's personality, where the person is, what the expresser is currently doing, who the receiver is and so on.

In future work, the fusion of facial expression and body gestures will be one of our great interests. These two modes are complementary: because facial expressions are subtle, the features are difficult to be extracted by automatic image processing methods, especially when observed at a distance; body movements are normally on a large scale. However, facial expressions have more discriminating power and are generally shared across individuals of different cultural backgrounds; body gestures tend to vary relatively more

between individuals. Other modalities such as the tone of voice and the speech content can also be integrated in a multimodal recognition structure.

8

Conclusions and Future Work

This thesis has proposed a multimodal interaction system for a domestic assistive robot. It consists of several sub-systems, each of which provides a dedicated functionality, and a fusion agent to integrate all of the input information, so that the robot can understand multiple communication input modes from human users, including speech, head-pose, eye-gaze and gestures. An emotional gesture recognition method has also been proposed, giving the robot the potential of understanding the user's emotional states. This thesis has made significant contributions in designing and realising the multimodal interaction system. The proposed approaches have been described in detail in each chapter. Three principles in our design are emphasized: naturalness, effectiveness and consideration of individual differences. This thesis mainly covers the transactional intelligence for an assistive robot, while some preliminary collaborative work has also been done towards our final goal of a mobile assistive robot to provide help for the elderly in the living room, office or kitchen scenarios. It was originally motivated by the fact that a large number of human-friendly robots are in demand to provide assistance to elderly people with some simple daily tasks, however, an assistive robot has wider potential applications for all humans as an assistant, a social companion or a team member.

This concluding chapter provides a summary of the approaches that have been discussed in the previous chapters, identifies the current limitations of the proposed approaches, and outlines possible future work that can be pursed to improve the current system.

8.1 Conclusions

Face Detection and Tracking

Chapter 3 described a robust face tracking method. The proposed method extended the well-known face detection algorithm [168, 283] by the combination of colour and depth information. The proposed face detection method provides a good initial estimation of the position and the colour distribution of the face for the CAMSHIFT face tracking method. Several criteria are used to detect the loss of tracking, in which case the position and size of the face are re-initialized by applying the detection method, and tracking is then recovered.

Consideration of Individual Differences

Recognition of a person's identification plays an important role which enables the robot to give appropriate responses for different individuals. In our design, the differences between individuals are emphasized. The ability to treat individuals differently has been considered as a desirable human-like feature in the development of robotics. This is also consistent with our principles of naturalness and effectiveness. A profile for each individual user is built, which includes several types of information about a specific user, such as the body height, self-defined gestures, the pointing method preference, a trained speech profile, a lookup table for mapping between certain pronunciations in specific contexts, and temporal relationships between pointing gestures and speech. The user's ID is first recognized when he/she stands in front of the robot and his/her profile is then loaded. This feature allows users to define their own sets of gestures, and improve the estimation of the pointing direction and the recognition of the speech utterance, as demonstrated in the experiments. In particular, the choice of pointing methods has been extensively researched considering the individual preference associated with the location of the target.

Hand Gesture Recognition

A hand gesture recognition system has been proposed in Chapter 4. The proposed system is able to recognize both motion patterns (i.e. dynamic gestures) and hand postures (i.e. static gestures). More specifically, the proposed 3D particle filter-based hand tracking approach combines three kinds of information: colour, motion and depth to track the hands effectively. The robustness of the method is attributed to the multi-hypothesis nature of the

Particle Filter and the combination of three cues. In the experiments, the system did not lose tracking even when the person was wearing a short-sleeve shirt exposing his forearm. This reflects the principle of naturalness in our design. The trajectory of the hands is then used by a Finite State Machine method to model the dynamic motion patterns.

To recognize the meanings conveyed by the hand shapes, the hand images are first segmented from the forearm by a Gaussian Mixture Model (GMM) method. The iterative Expectation-Maximization (EM) algorithm is used to find the parameters of the model, whose initial values are estimated by the PCA method. The segmented images are then rectified to the preferred orientation and resized to the same size. The matching procedure between the incoming enquiry image and the templates adopts the well known distance transformation method. In the experiments, a stability counter was used to determine whether the hand shape gestures were stable. In this way, the transitional state from one shape to another is ignored. Moreover, the user can define his/her unique set of gestures.

Multimodal Interaction and Dialogue Manager

An assistive robot will probably be used to serve an elderly or disabled person. It is thus essential that the robot is able to understand the multiple communication modalities used by humans. The proposed multimodal interaction system combines several input channels that are naturally used in our daily life, including speech, gaze and gestures. Input from each information channel is first recognized by a specialized sub-system. The probabilistic fusion approach integrates available recognition results from all of the sub-systems, and the system takes into account that each recognition sub-system can only generate incomplete and possibly erroneous results. Instances in each class compete with each other and the one with highest Probability Related Score (PRS) is chosen. Mutual interaction is realised by a dialogue manager, which uses a task-orientated command table to determine whether the fusion result is sufficient for execution. If not, possibly because the robot fails to understand correctly or the command itself is ambiguous, the robot requests more information from the user using a synthetic voice. Experiments were designed to test this multimodal interaction system extensively, and the results showed the multimodal approach outperformed the unimodal speech method. Some preliminary collaboration work was conducted to combine this transactional intelligence system with the spatial intelligence systems implemented by colleagues.

Recognizing Emotional Gestures

Unlike the above interaction modalities, which are mainly to express one's requests, emotional gestures deliver complementary information which indicates one's emotional states, such as being happy, angry, fearful and sad. The recognition of a person's emotional states through body gestures has been achieved and presented in Chapter 7. The proposed approach adopts the Support Vector Machines (SVMs) classification method, and optimizes its parameters by a nonlinear searching method. The Cross-Validation technique is used to make the most of the available data and avoid the overfit problem. The approach was successfully tested on a publicly available database and our captured data. Another key finding is that hand gestures can provide distinguishing features, as a complement to body joints.

8.2 Limitations and Future Extensions

This section provides a summary of the main limitations of the current system and provides suggestions for the future extensions.

Currently, it is assumed that only one user communicates with the robot at one time. Although more than one person can be detected and tracked simultaneously, it is difficult for the robot to determine who is actually interacting with it at each instant. A possible method could be to find the person who is talking, but the robot would then need to exclude the cases when the people are talking to each other. Another constraint can be used, such as that the user should look at the robot while talking or calling the robot's name. However, the hands tracking and the forearm direction estimation also need to be modified for the situation of multiple users. Multiple sets of faceLAB, with each set estimating the head pose and gaze direction of one of the people, will also be required.

In future work, the robot should show more initiative. For example, rather than wait for the user to come to it to start a conversation, the robot will be able to detect a person waving a hand at a long distance or localize the sound source when the person calls the robot's name and then approaches him/her. Another example is that the robot may initiate interactions based on the user's emotional states and show more care like a close friend.

The current speech recognition system on the robot is relatively simple. A more sophisticated speech interpretation method appears to be needed to deal with the ASR

errors and the variety of spoken utterances. The feedback from some participants indicated that more features about an object could be included, such as shape, size and usage properties which are actually preferred in natural communication.

Currently, we consider whether the objects are along the user's eye gaze direction. In future work, the relative positions of different objects and the user's perspective could be considered together. As mentioned in [243] and [180], an object cannot be requested by the user if it is occluded by another object from his/her perspective.

The faceLAB device has a narrow field of view and a shallow depth of field, so it loses track of the person's head and eyes when he/she moves around freely. Body movement of the participants were limited when we wanted to evaluate the effect of the gaze direction in the task of object selection. If the faceLAB was equipped with a tilt-pan platform so that its position and view angle can be adjusted online and in real time, it will be possible to reduce the constraints on the user's body movement.

Currently, the robot is designed to serve in an indoor environment. It is attractive but challenging to make the robot work in outdoor environments. Besides the mobility of the robot, the main difficulty lies in the data acquisition. As mentioned in Chapter 2, the depth data captured by the current PMD camera and the Kinect will probably degrade very much under the influence of sunlight. The dramatic changes of the illumination conditions in the outdoor environment will also impose more difficulties on image processing.

Our current system for recognizing emotional gestures is still at a preliminary stage and runs offline. To achieve a real time online system, the expressive frames in the captured colour/depth image sequences need to be detected automatically. One possible method could be to use the motion segmentation method based on motion energy as in [41]. However, many false positives may occur, because not all of the gestures at the apex are necessarily related to an emotion in the pre-defined categories. A multimodal method combining expressive gestures with facial expressions, the content of speech utterances and the tone of voice may be useful to reduce false positives and also improve the recognition results. Furthermore, in our daily life, we often judge a person's emotional states through long-term observation. A single frame of observation is often not sufficient to make a judgement. A long-term observation is usually used to determine a person's emotional states in our daily life, especially for the low-energy states such as boredom,

tiredness, stress, anxiety and frustration. The number of emotional categories to be classified also needs to be enlarged.

The meanings of humans' behaviours are normally context-dependent. Several aspects have already been considered in our current system, which are implemented by the users' profiles. However, more aspects may need to be included, including when and where the interaction occurs, what has just happened to the user, and the expresser's personality.

The experiments were carried out in a laboratory setting. The behaviours of the participants were different from how they behave in their daily life, when they interacted with a robot, being recording by cameras and observed by the others. This was most obvious in the emotional gesture experiments. As pointed out in [88], it is difficult for non-professional actors to produce natural expressions without feeling the emotion that causes the expression, and change the expression of a certain emotion to another in a short time. In addition, in most of the trials in the experiments (except for the collaborative work), the robot only gave visual or audio feedback rather than physically implemented the requested tasks. This might have reduced the interests of the participants for further communication.

Therefore, a large amount of future work still remains to achieve the ultimate goal of a household assistive robot. The integrated system with both spatial intelligence and transactional intelligence needs to be tested extensively in realistic environments.

Appendix A: PMD Camera

Specification of the PMD camera

Table A-1: The PMD Camera Specifications [93]

Name	PMD[VISION]® 19K
Sensor type	CMOS-Matrix camera with 19200 pixels
Detector	½" Global Shutter PMD Sensor
Detector Dimensions	6.4 mm (H) x 4.8 mm (V)
Pixel Dimensions	40 µm (H) x 40 µm (V)
Resolution	160 (H) x 120 (V)
Optical fill factor	30 %
Unambiguous range	7,5 m at 20 MHz1)
z-Resolution	>6 mm
Field of View	40 °@f=12 mm
Illumination Power	Approximately 3W optical
Wavelength	870 nm
Frame Rate (3D)	up to 15 fps
Digital Interfaces	IEEE 802.3u IEEE1394
Power supply	9 V … 18 V
Weight	1400 g

Working Principle of the PMD Depth Camera

A Photonic Mixer Device (PMD) captures depth information using a Time-of-Flight (TOF) technique. By sampling and correlating the received signal with a reference signal, the PMD determines the phase shift of the signal and calculates the distance. A detailed description of its working principle can be found in [131, 158, 170]. Rather than measure the return-trip time of the emitted light pulse, which requires high precision clock circuits, PMD uses the phase shift measurement method. A brief introduction of the principle extracted from [132, 170] is provided as follows.

Given a reference signal $g(t)$ and the incident signal $s(t)$ on a PMD pixel, the pixel samples the correlation function $c(\tau)$ for a given internal phase delay τ:

$$c(\tau) = (s \otimes g)(\tau) = \lim \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) \cdot g(t+\tau) dt \tag{9.1}$$

For a cosinusoidal signal $g(t) = cos(\omega t)$ and the optical response signal $s(t) = k + a \cdot \cos(\omega t + \varphi)$ basic trigonometric calculus yields:

$$c(\tau) = h + \frac{a}{2} \cos(\omega\tau + \varphi) \tag{9.2}$$

where ω is the modulation frequency, a and h is the amplitude and the offset of the correlation function, respectively, and φ is the phase shift relating to the object distance.

The modulation frequency defines the distance unambiguousness of the distance sensing. The demodulation of the correlation function is done using four sequential samples of $c(\tau)$. Each sample is delayed by π/4 as shown in Figure A-1.

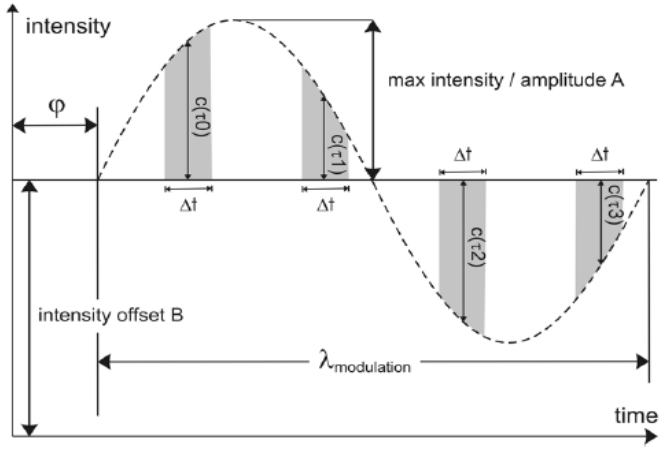


Figure A-1: Phase shift distance measurement principle. Picture from [132].

φ, a and h can be calculated using these four sample values:

$$\varphi = \arctan\left(\frac{A_3 - A_1}{A_0 - A_2}\right) \tag{9.3}$$

$$a = \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2} \tag{9.4}$$

$$h = \frac{A_0 + A_1 + A_2 + A_3}{4} \tag{9.5}$$

The distance between the sensor and the point can be then calculated directly by the phase shift φ:

$$d = \frac{\lambda_{\text{mod}}}{2} \cdot \frac{\varphi}{2\pi} \tag{9.6}$$

where $\lambda_{\text{mod}} = \frac{c}{w}$ is the modulation wavelength and c is the speed of light.

Appendix B: Kinect

Specification of the Kinect

Table B-1: The Kinect Specifications [193, 235]

Name	Kinect Sensor Array
Viewing angle (field of view)	57 °Horizontal , 43 °Vertical , 70 ºDiagonal
Mechanized tilt range (vertical)	±28 °
Frame rate (depth and colour stream)	30 FPS
Max Resolution, depth stream	VGA (640 ×480)
Max Resolution, colour stream	VGA (640 ×480)
Audio format	16-kHz, 16-bit mono pulse code modulation (PCM)
Audio input characteristics	A four-microphone array with 24-bit analog-to-digital converter (ADC) and Kinect-resident signal processing such as echo cancellation and noise suppression
Tilting range	±27 º
Data interface	USB 2.0

Working Principle of the IR Depth Camera in the Kinect

According to the datasheet of PrimeSensor reference design [235], the range camera of the Kinect employs the structured light technique to generate the depth data of the scene. The depth camera consists of an infrared laser projector combined with a monochrome CMOS sensor. Through reverse engineering and the investigation of its patent, a plausible guess of its working principle has been provided by Konolige and Mihelich [151] and summarized as follows.

The IR camera and the IR projector form a stereo pair with a fixed baseline. A set of fixed patterns with different intensities are emitted from the IR projector. The patterns are generated from a set of diffraction gratings with a special method to reduce the effect of zero-order propagation of a bright spot in the centre.

Depth is determined by the triangulation method as in the stereo-vision system. For each pixel in the observed IR image, a small searching window is used to calculate the

correlation between the local pattern and the fixed pattern from the projector. The best correspondence gives a disparity value at each pixel. The device then performs a further interpolation to achieve sub-pixel accuracy of 1/8 pixel.

For a normal stereo-vision system, the relationship between the disparity d and the depth z is:

$$z = \frac{b \cdot f}{d} \tag{10.1}$$

where b is the length of the baseline between two cameras, and f is the common focal length. However, the disparity is not the direct measurement but is calculated as

$$d = \frac{1}{8} \cdot (d_{offset} - kd) \tag{10.2}$$

where kd is the raw measurement in the Kinect disparity, d_{offset} is a specific offset value. Therefore, the resulting equation is:

$$z = \frac{b \cdot f}{\frac{1}{8} \cdot (d_{offset} - kd)} \tag{10.3}$$

References

1. A.R. Abbasi (2007), A Bayesian Network Approach to Interpret Affective States from Human Gestures: An Application to Affective Tutoring. [Online]. *diunibait*, pp. 1-8, Available from: http://www.di.uniba.it/intint/DC-ACII07/Abbasi.pdf.
2. T. Acharya and A.K. Ray, *Image Processing: Principles and Applications*: Wiley-Interscience. 2005
3. B. Achermann, X. Jiang, and H. Bunke, "Face recognition using range images". in *International Conference on Virtual Systems and MultiMedia*, 1997, pp. 129–136.
4. Adept Mobile Robots (2011). PeopleBot. [Online]. Available from: http://www.mobilerobots.com/researchrobots/researchrobots/PeopleBot.aspx.
5. I.C. Albitar, P. Graebling, and C. Doignon, "Robust Structured Light Coding for 3D Reconstruction". in *11th International Conference on Computer Vision (ICCV)*, 2007, pp. 1-6.
6. J. Alon, et al., "Simultaneous Localization and Recognition of Dynamic Hand Gestures". in *IEEE Workshop on Motion and Video Computing* 2005, pp. 254--260.
7. O. Alter, P. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data Processing and modeling". in *Proc Natl Acad Sci USA*, 2000, pp. 10101-10106.
8. N. Ambady and R. Rosenthal, "Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis", *Psychological Bulletin*, vol. 111(2), pp. 256-274, 1992.
9. Australian Bureau of Statistics, "Australian Population Projections 1999-2101", in *Australian Demographic Statistics*, (ABS Cat. no. 3101.0), 2000.
10. A.H. Barr, "Superquadrics and angle-preserving transformations ", *Computer Graphics and Applications*, vol. 1(1), pp. 11-23, 1981.
11. S.S. Beauchemin and J.L. Barron, "The computation of optical flow", *ACM Comput. Surv.*, vol. 27(3), pp. 433-466, 1995.
12. C. Beder, B. Bartczak, and R. Koch, "A Comparison of PMD-Cameras and Stereo-Vision for the Task of Surface Reconstruction using Patchlets". in *Computer Vision and Pattern Recognition* 2007, pp. 1-8.
13. P.J.L.V. Beek, et al., "Semantic segmentation of videophone image sequences". in *Int. Conf. on Visual Communications and Image Processing*, 1992, pp. 1182–1193.
14. P.N. Belhumeur, "A Bayesian Approach to Binocular Stereopsis", *International Journal of Computer Vision*, vol. 19(3), pp. 237-260, 1996.
15. J.L.C.a.F. Berard, "Multi-model tracking of faces for video communications". in *International Conference on Computer Vision and Pattern Recognition*, 1997.
16. C. Beumier and M. Acheroy, "Face verification from 3D and grey level cues", *Pattern Recognition Letters*, vol. 22, pp. 1321–1329, 2001.

17. D. Beymer and M. Flickner, "Eye gaze tracking using an active stereo head". in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, pp. 1-8.

18. N. Bianchi-Berthouze, "Using motion capture to recognize affective states in humans". in *Proceedings of Measuring Behavior*, 2008, pp. 28-35.

19. N. Bianchi-Berthouze (2010). AffectME: Affective Multimodal Engagement. [Online]. Available from: http://web4.cs.ucl.ac.uk/uclic/people/n.berthouze/AffectME.html.

20. N. Bianchi-Berthouze and A. Kleinsmith, "A categorical approach to affective gesture recognition", *Connection Science*, vol. 15(4), pp. 259–269, 2003.

21. L.-S. Bieri and J. Jacot, "Three-dimensional vision using structured light applied to quality control in production line". in *Optical Metrology in Production Engineering*, 2004.

22. F. Blais, "Review of 20 years of range sensor development", *Electronic Imaging*, vol. 13(1), pp. 231-243, 2004.

23. R.A. Bolt, "“Put-that-there”: Voice and gesture at the graphics interface", *SIGGRAPH Comput. Graph.*, vol. 14(3), pp. 262-270, 1980.

24. L. Bonansea, "3D Hand gesture recognition using a ZCam and an SVM-SMO classifier", Iowa State University, Iowa, US, 2009.

25. G. Borgefors, "Distance transformations in digital images", *Computer Vision Graphic Image Process.*, vol. 34(3), pp. 344-371, 1986.

26. G. Borgefors, "Hierarchical Chamfer Matching: a Parametric Edge Matching Algorithm", *IEEE Transactions on SYSTEMS, MAN, AND CYBERNETICS*, vol. 10(6), pp. 17, 1988.

27. J.-y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker: description of the algorithm", Intel Microprocessor Research Labs, 2000.

28. J.-Y. Bouguet (2010). Camera Calibration Toolbox for Matlab. [Online]. Available from: http://www.vision.caltech.edu/bouguetj/calib_doc/.

29. K.W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches to three-dimensional face recognition". in *17th International Conference on Pattern Recognition*, 2004, pp. 358-361 Vol.1.

30. Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision". in *Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2001, pp. 1124-1137.

31. G.R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface", *Intel Technology Journal Q2*, pp. 1-15, 1998.

32. M. Bray, E. Koller-Meier, and L. Van Gool, "Smart particle filtering for 3D hand tracking". in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 675-680.

33. M. Breitenstein, et al., "Head Pose Estimation from Passive Stereo Images", in *Image Analysis*, Springer Berlin, Heidelberg. 2009, pp. 219-228.

34. P. Breuer, C. Eckes, and S. Muller, "Hand gesture recognition with a novel IR time-of-flight range camera: a pilot study". in *Proceedings of the 3rd international conference on Computer vision/computer graphics collaboration techniques*, Rocquencourt, France, Springer-Verlag, 2007, pp. 247-260.

35. A.M. Bronstein, M.M. Bronstein, and R. Kimmel, "Expression-invariant 3D face recognition.". in *Audio- and Video-Based Person Authentication*, 2003, pp. 62-70.

36. M.Z. Brown, D. Burschka, and G.D. Hager, "Advances in computational stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25(8), pp. 993-1008, 2003.

37. R. Bruneli and T. Poggio, "Face recognition: features versus templates", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042-1052, 1993.

38. C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

39. N. Burrus (2011). Kinect Calibration. [Online]. Available from: http://nicolas.burrus.name/index.php/Research/KinectCalibration.

40. S. Caifeng, et al., "Real time hand tracking by combining particle filtering and mean shift". in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 669-674.

41. A. Camurri, et al., "Multimodal analysis of expressive gesture in music and dance performances". in *Gesture-Based Communication in Human-Computer Interaction*, 2004, pp. 357-358.

42. B. Carpenter, *The logic of typed feature structures: with applications to unification grammars, logic programs, and constraint resolution*. Cambrige, England: Cambrige University Press. 1992.

43. G. Castellano, L. Kessous, and G. Caridakis, "Emotion Recognition through Multiple Modalities: Face, Body Gesture, Speech Affect and Emotion in Human-Computer Interaction", Ed. C. Peter and R. Beale, Springer Berlin:Heidelberg. 2008, pp. 92-103.

44. Z. Cernekova, N. Nikolaidis, and I. Pitas, "Single camera pointing gesture recognition using spatial features and support vector machines". in *European Signal Processing Conference*, 2007, pp. 130-134.

45. F. Chen, G.M. Brown, and M. Song, "Overview of 3-d shape measurement using optical methods", *Optical Engineering*, vol. 39(1), pp. 10-22, 2000.

46. Y. Cheng, "Mean shift, mode seeking and clustering", *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 17, pp. 790-799, 1995.

47. R. Cipolla, et al., "Hand Tracking Using a Quadric Surface Model and Bayesian Filtering". in *IMA Conference on the Mathematics of Surfaces* 2003, pp. 129-141.

48. C. Codella, et al., *Interactive simulation in a multi-person virtual world*, in *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1992, ACM: Monterey, California, United States. p. 329-334.

49. F. Colas and P. Brazdil, *Artificial Intelligence in Theory and Practice*. Vol. 217. Boston: Springer. 2006.

50. T.F. Cootes and C.J. Taylor, "Active shape models—'smart snakes'". in *British Machine Vision Conference*, 1992, pp. 266–275.

51. C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, vol. 20(3), pp. 273-297, 1995.

52. M. Coulson, "Attributing Emotion to Static Body Postures: Recognition Accuracy, Confusions, and Viewpoint Dependence", *Nonverbal Behavior*, vol. 28(2), pp. 117-139, 2004.

53. R. Cowie, et al., "'FEELTRACE': An Instrument For Recording Perceived Emotion In Real Time". in *workshop on speech and emotion*, 2000, pp. 19-24.

54. A.R. Damasio, *Descartes' Error: Emotion, Reason, and the Human Brain*. New York: Avon Books. 1994.

55. T. Darrell, et al., "Integrated person tracking using stereo, color, and pattern detection". in *IEEE Conference on Computer Vision and Pattern Recognition*, 1998, pp. 601-608.

56. J. Davis and M. Shah, "Recognizing hand gestures". in *European Conference on Computer Vision, ECCV*, 1994, pp. 331-340.

57. A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm ", *Journal of the Royal Statistical Society*, vol. 39(1), pp. 1-38, 1977.

58. J.M.S. Dias, et al., "OGRE - open gestures recognition engine". in *17th Brazilian Symposium on Computer Graphics and Image Processing*, 2004, pp. 33-40.

59. D. Droeschel, et al., "Using Time-of-Flight Cameras with Active Gaze Control for 3D collision avoidance". in *ICRA*, Alaska, 2010.

60. S. Effendi and R. Jarvis, "Camera Ego-Motion Estimation Using Phase Correlation under Planar Motion Constraint". in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2010, pp. 158-165.

61. S. Effendi, R. Jarvis, and W.H. Li, "3D Shape Recovery by Superquadrics Model Using Object Silhouettes and Stereo Disparity". in *IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, 2010.

62. S. Effendi, R. Jarvis, and D. Suter, "Robot Manipulation Grasping of Recognized Objects for Assistive Technology Support using Stereo Vision". in *Australasian Conference on Robotics and Automation*, 2008.

63. S. Effendi, R. Jarvis, and L. Wai Ho, "3D shape recovery by superquadrics model using object Silhouettes and stereo disparity". in *IEEE Conference on Cybernetics and Intelligent Systems (CIS)*, 2010, pp. 82-89.

64. J. Eisenstein and C.M. Christoudias, "A Salience-Based Approach to Gesture-Speech Alignment". in *Proceedings of the Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting*, 2004.

65. P. Ekman and W.V. Friesen, *Unmasking the Face: A Guide to Recognizing Emotions From Facial Expressions*. N.J.: Prentice Hall. 1975.

66. P. Ekman and W.V. Friesen, *The Face Action Coding System*. San Francisco, CA: Consulting Psychologists Press. 1978.

67. P.C. Ellsworth and K.R. Scherer., "Appraisal processes in emotion", in *Handbook of affective sciences*, Ed. K.R.S. R. J. Davidson, & H. H. Goldsmith, Oxford University Press: New York. 2003, pp. 572–595.

68. A. Erol, et al., "Vision-based hand pose estimation: A review", *Computer Vision Image Understanding*, vol. 108(1-2), pp. 52-73, 2007.

69. A. Ess, Leibe, B., Schindler, K., and Van Gool, L., "Moving obstacle detection in highly dynamic scenes". in *international conference on Robotics and Automation*, 2009, pp. 4451-4458.

70. P. Fechteler, P. Eisert, and J. Rurainsky, "Fast and High Resolution 3D Face Scanning". in *IEEE International Conference on Image Processing*, 2007, pp. 81-84.

71. L. Fletcher (2003). Face Detection and Tracking in Colour Images. [Online]. Available from: http://www.syseng.anu.edu.au/~luke/cvcourse.htm.

72. D.A. Forsyth and J. Ponce, *Computer vision: a modern approach*: Prentice Hall. 2003.

73. Fotonic (2011). Fotonic C70. [Online]. Available from: http://www.fotonic.com/content/Products/Default.aspx.

74. Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Computational Learning Theory: Eurocolt*, pp. 23-27, 1995.

75. S.E. Ghobadi, et al., "Hand Segmentation Using 2D / 3D Images". in *Image and Vision Computing New Zealand*, 2007, pp. 64-69.

76. D. Glowinski, et al., "Towards a Minimal Representation of Affective Gestures", *IEEE Transactions on Affective Computing*, vol. 2(2), pp. 106-118, 2011.

77. G. Gordon, "Face recognition based on depth maps and surface curvature", *Geometric Methods in Computer Vision*, pp. 1-12, 1991.

78. J.C.a.A. Goshtasby, "Detecting human faces in color images", *Image Vision Computer*, vol. 18, pp. 63-75, 1999.

79. V. Govindaraju, "Locating human faces in photographs", *International Journal Computer Vision*, vol. 19, 1996.

80. H.P. Graf, et al., "Locating faces and facial parts". in *Int.Workshop on Automatic Face-and Gesture-Recognition*, 1995, pp. 41-45.

81. H.P. Graf, et al., "Multi-modal system for locating heads and faces". in *2nd Int. Conf. on Automatic Face and Gesture Recognition*, 1996, pp. 277–282.

82. D. Grandjean, D. Sander, and K.R. Scherer, "Conscious emotional experience emerges as a function of multilevel, appraisal-driven response synchronization", *Consciousness and Cognition*, vol. 17(2), pp. 484-495, 2008.

83. M.T.a.E. Grosso, "Active vision-based face authentication", *Image Vision Comput.*, vol. 18, pp. 299–314, 2000.

84. Y.P. Guan, "Non-wearable pointing gesture recognition based on single optimal view camera". in *2nd International Conference on Computer Science and its Applications*, 2009, pp. 1-5.

85. H. Gunes and M. Piccardi, "Fusing Face and Body Display for Bi-modal Emotion Recognition: Single Frame Analysis and Multi-frame Post Integration". in *First International Conference Affective Computing and Intelligent Interaction*, Springer-Verlag, 2005.

86. H. Gunes and M. Piccardi, "Fusing Face and Body Gesture for Machine Recognition of Emotions". in *IEEE International Workshop on Robots and Human Interactive Communication*, 2005.

87. H. Gunes and M. Piccardi, "Bi-modal emotion recognition from expressive face and body gestures", *Network and Computer Applications*, vol. 30(4), pp. 1334-1345, 2006.

88. H. Gunes and M. Piccardi, "A Bimodal Face and Body Gesture Database for Automatic Analysis of Human Nonverbal Affective Behavior". in *18th International Conference on Pattern Recognition (ICPR)*, 2006.

89. H. Gunes, et al., "Emotion representation, analysis and synthesis in continuous space: A survey". in *IEEE International Conference on Automatic Face & Gesture Recognition*, 2011, pp. 827-834.

90. G. Guo, S.Z. Li, and K. Chan, *Face recognition by support vector machines*, in *IEEE International Conference on Automatic Face and Gesture Recognition*. 2000. p. 196-201.

91. O.K. Gupta and R.A. Jarvis, "Optimal Global Path Planning in Time Varying Environments Based on a Cost Evaluation Function". in *21st Australasian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, Auckland, New Zealand, Springer-Verlag, 2008, pp. 150-156.

92. O.K. Gupta and R.A. Jarvis, "Robust pose estimation and tracking system for a mobile robot using a panoramic camera". in *Robotics Automation and Mechatronics (RAM)*, 2010, pp. 533-539.

93. S.R. Hag, "PMD[Vision] 19k Instruction Manual 3D time-of-flight camera", PMDTec GmbH, 2005.

94. M. Haker, et al., "Deictic Gestures with a Time-of-Flight Camera". in *International Gesture Workshop Gesture in Embodied Communication and Human-Computer Interaction*, Springer, 2009, pp. 110-121.

95. M. Haker, et al., "Deictic Gestures with a Time-of-Flight Camera". in *Gesture in Embodied Communication and Human-Computer Interaction - International Gesture Workshop GW 2009*, Springer, 2010, pp. 110-121.

96. D.W. Hansen and Q. Ji, "In the Eye of the Beholder: A Survey of Models for Eyes and Gaze", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 32(3), pp. 478-500, 2010.

97. E. Harte and R. Jarvis, "Multimodal Human-Robot Interaction in an Assistive Technology Context". in *Proceedings of the Second International Conferences on Advances in Computer-Human Interactions (ACHI '09)*, 2009, pp. 212-218.

98. R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision*: Cambridge University Press. 2003.

99. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer. 2001.

100. K. Hayashi, et al., "Multiple-person tracker with a fixed slanting stereo camera". in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 681-686.

101. M.A. Hearst, "Support Vector Machines", *IEEE Intelligent Systems, Trends & Controversies feature*, vol. 13(4), pp. 18-28, 1998.

102. G. Heidemann, I. Bax, and H. Bekel, *Multimodal interaction in an augmented reality scenario*, in *Proceedings of the 6th international conference on Multimodal interfaces*. 2004, ACM: State College, PA, USA. p. 53-60.

103. D. Herrera, J. Kannala, and J. Heikkila, "Accurate and Practical Calibration of a Depth and Color Camera Pair". in *the 14th International Conference on Computer Analysis of Images and Patterns (CAIP)* 2011.

104. E. Hjelmas, "Face Detection: A Survey", *Computer Vision and Image Understanding*, vol. 83(3), pp. 236-274, 2001.

105. G. Holst, "Face detection by facets: Combined bottom-up and top-down search using compound templates". in *the International Conference on Image Processing*, 2000.

106. H. Holzapfel, "A Dialogue Manager for Multimodal Human-Robot Interaction and Learning of a Humanoid Robot ", *Industrial Robots*, vol. 35(6), pp. 528-535, 2008.

107. H. Holzapfel, K. Nickel, and R. Stiefelhagen, "Implementation and evaluation of a constraint-based multimodal fusion system for speech and 3D pointing gestures". in *Proceedings of the 6th international conference on Multimodal interfaces*, State College, PA, USA, ACM, 2004, pp. 175-182.

108. H. Hongo, et al., "Focus of attention for face and hand gesture recognition using multiple cameras". in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 156-161.

109. C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification", *Bioinformatics*, vol. 1(1), pp. 1-16, 2010.

110. C.-L. Huang and W.-Y. Huang, "Sign language recognition using model-based tracking and a 3D Hopfield neural network", *Machine Vision Application*, vol. 10(5-6), pp. 292-307, 1998.

111. C.L. Huang and C.W. Chen, "Human facial feature extraction for face interpretation and recognition", *Pattern Recognition*, vol. 25, pp. 1435–1444, 1992.

112. M. Hunke and A. Waibel, "Face locating and tracking for human-computer interaction". in *28th Asilomar Conference on Signals, Systems and Computers*, 1994, pp. 1277-1281.

113. K. Imagawa, L. Shan, and S. Igi, "Color-based hands tracking system for sign language recognition". in *Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 462-467.

114. A. Jaimes and N. Dimitrova, "Human-centered multimedia: culture, deployment, and access", *Multimedia*, vol. 13(1), pp. 12-19, 2006.

115. R. Jarvis, "A Go Where You Look Tele-autonomous Rough Terrain Mobile Robot", in *Experimental Robotics VIII*, Ed. B. Siciliano and P. Dario, Springer Berlin. 2003, pp. 624-633.

116. R. Jarvis, et al., "An intelligent robotic assistive living system". in *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, Corfu, Greece, ACM, 2009, pp. 1-8.

117. R.A. Jarvis, "A Perspective on Range Finding Techniques for Computer Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5(2), pp. 122-139, 1983.

118. R.A. Jarvis, "Intelligent Robotics: Perception, Reasoning and Action". in *Bicentennial Electrical Engineering Congress: Electro-technology a Springboard for the Future*, 1988, pp. 45-51.

119. S.-H. Jeng, et al., "Facial feature detection using geometrical face model: An efficient approach", *Pattern Recognition*, vol. 31(3), pp. 273-282, 1998.

120. Z. Jian-Pei, L. Zhong-Wei, and Y. Jing, "A parallel SVM training algorithm on large-scale classification problems". in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, 2005, pp. 1637-1641.

121. Y. Jianjun, Y. Hongxun, and J. Feng, "Based on HMM and SVM multilayer architecture classifier for Chinese sign language recognition with large vocabulary". in *Third International Conference on Image and Graphics, 2004*, 2004, pp. 377-380.

122. M.J. Johnson, et al., "Rehabilitation and assistive robotics", *IEEE Robotics & Automation Magazine*, vol. 15(3), pp. 16-110, 2008.

123. M. Johnston, "Unification-based multimodal parsing". in *Proceedings of the 17th international conference on Computational linguistics*, 1998.

124. M. Johnston, et al., "Unification-based multimodal integration". in *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 1997, pp. 281-288.

125. N. Jojicy, et al., "Detection and Estimation of Pointing Gestures in Dense Disparity Maps". in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

126. I.T. Jolliffe, *Principal Component Analysis*. Statistics. New York: Springer. 2002.

127. I.T. Jolliffe, *Principal Component Analysis*. Springer Series in Statistics. NY: Springer. 2002.

128. K. Jonsson, et al., "Support vector machines for face authentication". in *British Machine Vision Conference*, 1999, pp. 543–553.

129. K. Jonsson, et al., "Learning support vectors for face verification and recognition". in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 208-213.

130. G. Junxia, et al., "Action and Gait Recognition From Recovered 3-D Human Joints", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40(4), pp. 1021-1033, 2010.

131. T. Kahlmann and H. Ingensand, "Calibration and improvements of the high-resolution range-imaging camera SwissRanger". in *SPIE-IS&T Electronic Imaging*, 2005, pp. 144-155.

132. T. Kahlmann, F. Remondino, and H. Ingensand, "Calibration for increased accuracy of the range imaging camera swissranger". in *ISPRS*, 2006, pp. 136-144.

133. E. Kaiser, et al., "Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality". in *Proceedings of the 5th international conference on Multimodal interfaces*, Vancouver, British Columbia, Canada, 2003, pp. 12-19.

134. R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *Journal of Basic Engineering*, vol. 82, pp. 35-45, 1960.

135. M. Kamachi, M. Lyons, and J. Gyoba (1998). The Japanese Female Facial Expression (JAFFE) Database [Online]. Available from: http://www.kasrl.org/jaffe.html.

136. T. Kanade, Y. Tian, and J.F. Cohn, "Comprehensive Database for Facial Expression Analysis". in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 46-53.

137. J. Kang-Hyun, Y. Kuno, and Y. Shirai, "Manipulative hand gesture recognition using task knowledge for human computer interaction". in *Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 468-473.

138. A. Kapoor, W. Burnleson, and R.W. Picard, "Automatic Prdiction of Frustration", *International Journal of Human-Computer Studies*, vol. 65(8), pp. 724-736, 2007.

139. A. Kapoor, R.W. Picard, and Y. Ivanov, "Probabilistic Combination of Multiple Modalities to Detect Interest". in *17th International Conference on (ICPR'04) Pattern Recognition*, IEEE Computer Society, 2004, pp. 969-972.

140. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models". in *1st Int Conf. on Computer Vision*, 1987, pp. 321-331.

141. M. Kaur, et al., "Where is "it"? Event Synchronization in Gaze-Speech Input Systems". in *International Conference on Multimodal Interfaces*, 2003, pp. 151-158.

142. A. Kendon, *Gesture and Speech: How they Interact*, in *Nonverbal Interaction*. 1983, Beverley Hills: Sage Publication.

143. L. Kessous, G. Castellano, and G. Caridakis, "Multimodal emotion recognition in speech-based interaction using facial expression, body gesture and acoustic analysis", *Journal on Multimodal User Interfaces*, vol. 3(1), pp. 33-48, 2010.

144. J. Ki and Y.-M. Kwon, "3D Gaze Estimation and Interaction". in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, 2008, pp. 373-376.

145. H.-O. Kim, S. Kim, and S.-K. Park, "Pointing Gesture-based unknown Object Extraction for Learning Objects with Robot". in *International Conference on Control, Automation and Systems*, 2008.

146. A. Kleinsmith and N. Bianchi-Berthouze, "Towards Learning Affective Body Gesture". in *International Workshop on Epigenetic Robotics*, 2003, pp. 169 - 170.

147. A. Kleinsmith, T. Fushimi, and N. Bianchi-Berthouze, "An Incremental and Interactive Affective Posture Recognition System". in *International workshop on Adapting the Interacion Style o Affective Factors*, 2005, pp. 1-13.

148. A. Kleinsmith, P.R.D. Silva, and N. Bianchi-Berthouze, "Grounding Affective Dimensions into Posture Features". in *LNCS*, 2005, pp. 362-270.

149. R. Klose, J. Penlington, and A. Ruckelshausen, "Usability study of 3D Time-of-Flight cameras for automatic plant phenotyping", *Image Analysis for Agricultural Products and Processes*, vol. 69, pp. 93-105, 2009.

150. G. Knight (1999). Umi robot user and programmers manual, for the umi rt100+ win32 library. [Online]. Available from: http://services.eng.uts.edu.au/ carlo/pdf/.

151. K. Konolige and P. Mihelich (2010). Technical description of Kinect calibration. [Online]. Available from: http://www.ros.org/wiki/kinect_calibration/technical.

152. D.B. Koons, C.J. Sparrell, and K.R. Thorisson, "Integrating simultaneous input from speech, gaze, and hand gestures", in *Intelligent multimedia interfaces*, American Association for Artificial Intelligence. 1993, pp. 257-276.

153. V. Laban, *The Mastery of Movement*. London: MacDonald & Evans. 1988.

154. G. Lacey and K.M. Dawson-Howe, "The application of robotics to a mobility aid for the elderly blind", *Robotics and Autonomous Systems*, vol. 23(4), pp. 245-252, 1998.

155. M. Lades, et al., "Distortion Invariant object recognition in the dynamic link architecture", *IEEE Trans. Computers*, vol. 42, pp. 300-311, 1993.

156. J.C. Lagarias, et al., "Convergence Properties of the Nelder--Mead Simplex Method in Low Dimensions", *SIAM J. on Optimization*, vol. 9(1), pp. 112-147, 1998.

157. K.M. Lam and H. Yan, "Locating and extracting the eye in human face images", *Pattern Recognition*, vol. 29, pp. 771–779, 1996.

158. R. Lange, "3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCDtechnology", Ph.D., University Siegen, 2000.

159. A. Lanitis, C.J. Taylor, and T.F. Cootes, "Automatic tracking, coding and reconstruction of human faces, using flexible appearance models", *IEEE Electronic Letter*, vol. 30, pp. 1578–1579, 1994.

160. A. Lanitis, C.J. Taylor, and T.F. Cootes, "Automatic interpretation and coding of face images using flexible models", *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 19, pp. 743-756, 1997.

161. A. Laurentini, "The Visual Hull Concept for Silhouette-Based Image Understanding", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16(2), pp. 150-162, 1994.

162. J.C. Lee and E. Milios, "Matching range images of human faces". in *Int'l Conf. on Comp. Vision*, 1990, pp. 722–726.

163. Y. Lee, et al., "3D face recognition using statistical multiple features for the local depth information". in *16th International Conference on Vision Interface*, 2003, pp. 429-434.

164. Z. Li and R. Jarvis, "A multi-modal gesture recognition system in a Human-Robot Interaction scenario". in *IEEE International Workshop on Robotic and Sensors Environments*, 2009, pp. 41-46.

165. Z. Li and R. Jarvis, "Real time Hand Gesture Recognition using a Range Camera". in *Australasian Conference on Robotics and Automation (ACRA)*, Sydney, Australia, 2009.

166. Z. Li and R. Jarvis, "Visual interpretation of natural pointing gestures in 3D space for human-robot interaction". in *11th International Conference on Control Automation Robotics & Vision (ICARCV)*, 2010, pp. 2513-2518.

167. Z. Li and R. Jarvis, "Multimodal Interaction System for a Household Assistive Robot". in *Australasian Conference on Robotics and Automation (ACRA)*, Melbourne, Australia, 2011.

168. R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection". in *International Conference on Image Processing*, 2002, pp. 900-903.

169. S.H. Lin, S.Y. Kung, and L.J. Lin, "Face recognition/detection by probabilistic decision-based neural network", *IEEE Trans. Neural Networks*, vol. 8, pp. 114-132, 1997.

170. M. Lindner and A. Kolb, "Calibration of the intensityrelated distance error of the PMD FOF-camera". in *SPIE*, 2007.

171. C. Liu, et al., "Application of Adaboost based ensemble SVM on IKONOS image classification". in *18th International Conference on Geoinformatics*, 2010, pp. 1-5.

172. N. Liu, B.C. Lovell, and P.J. Kootsookos, "Evaluation of HMM Training Algorithms for Letter Hand Gesture Recognition". in *IEEE International Symposium on Signal Processing and Information Technology*, Darmstadt, Germany, 2003, pp. 4-7.

173. X. Liu and K. Fujimura, *Hand gesture recognition using depth data*, in *the sixth IEEE international Conference on Automatic face and gesture recognition*. 2004. p. 529-534.

174. M.M. Loper, N.P. Koenig, and S.H. Chernova, "mobile human-robot teaming with environmental tolerance", *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 157-164, 2009.

175. Los Alamos National Laboratory (2003). OSU SVM Version 3.00. [Online]. Available from: http://svm.sourceforge.net/docs/3.00/api/.

176. S. Lu, et al., "Using multiple cues for hand tracking and model refinement". in *Computer Vision and Pattern Recognition*, 2003, pp. 43-50.

177. B.D. Lucas and T. Kanade, *An iterative image registration technique with an application to stereo vision*, in *Proceedings of the 7th international joint conference on Artificial intelligence*. 1981, Morgan Kaufmann Publishers Inc.: Vancouver, BC, Canada. p. 674-679.

178. P.P. Maglio, et al., *Gaze and Speech in Attentive User Interfaces*, in *Proceedings of the Third International Conference on Advances in Multimodal Interfaces*. 2000, Springer-Verlag: Beijing, China. p. 1-7.

179. S. Malassiotis, N. Aifanti, and M.G. Strintzis, "A gesture recognition system using 3D data". in *Proceedings First International Symposium on 3D Data Processing Visualization and Transmission*, IEEE Comput. Soc, 2002, pp. 190-193.

180. L.F. Marin-Urias, et al., "Towards shared attention through geometric reasoning for Human Robot Interaction". in *9th IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 331-336.

181. R.b.D. MATLAB (2011). Optimizing Nonlinear Functions. Available from: http://www.mathworks.com.au/help/techdoc/math/bsotu2d.html#bsgpq6p-11.

182. S. May, et al., "Robust 3D-mapping with time-of-flight cameras". in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1673-1678.

183. S. May, et al., "Three-dimensional mapping with time-of-flight cameras", *Journal of Field Robotics*, vol. 26(11-12), pp. 934–965, 2009.

184. S. McKenna, S. Gong, and J.J. Collins, "Face tracking and pose representation". in *British Machine Vision Conference*, 1996, pp. 755-764.

185. G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*: Wiley Interscience. 2004.

186. A. Mehrabian, *Nonverbal Communication*. Chicago, Illinois Aldine-Atherton. 1972.

187. A. Mehrabian and J. Friar, "Encoding of attitude by a seated communicator via posture and position cues", *Consulting and Clinical Psychology*, vol. 33, pp. 330-336, 1969.

188. M.D. Meijer, "The contribution of general features of body movement to the attribution of emotions", *Nonverbal Behavior*, vol. 13(4), pp. 247-268, 1989.

189. MESA Imaging (2011). SwissRanger 4000. [Online]. Available from: http://www.mesa-imaging.ch.

190. A.B. Michael Isard, "CONDENSATION--conditional density propagation for visual tracking", *International Journal of Computer Vision*, pp. 5-28, 1998.

191. Microsoft (2009). Speech SDK 5.1. [Online]. Available from: http://www.microsoft.com/download/en/details.aspx?id=10121.

192. Microsoft (2011). Kinect for Windows. [Online]. Available from: http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/.

193. Microsoft (2011 June). Kinect Programming Guide. Beta 1 Draft Version 1.0a]. [Online]. Available from: http://kinectforwindows.org/.

194. Z. Mo, J.P. Lewis, and U. Neumann, *SmartCanvas: a gesture-driven intelligent drawing desk system*, in *Proceedings of the 10th international conference on Intelligent user interfaces*. 2005, ACM: San Diego, California, USA. p. 239-243.

195. B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation", *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 19(1), pp. 696 - 710 1997.

196. B. Moghaddam, W. Wahid, and A. Pentland, "Beyond eigenfaces: Probabilistic matching for face recognition". in *International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 30-35.

197. M. Montemerlo, et al., *Experiences with a mobile robotic guide for the elderly*, in *Eighteenth national conference on Artificial intelligence*. 2002, American Association for Artificial Intelligence: Edmonton, Alberta, Canada. p. 587-592.

198. V. Montreuil, et al., "Planning human centered robot activities". in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2007, pp. 2618-2623.

199. C. Moore and P.J. Dunham, *Joint Attention: Its Origins and Role in Development*: Lawrence Erlbaum Associates. 1995.

200. T. Mori, et al., "Hierarchical recognition of daily human actions based on Continuous Hidden Markov Models". in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, 2004, pp. 779-784.

201. D. Morris, et al., *Gesture, their origin and distribution*. London: Jonathan Cape. 1979.

202. R. Morros, et al., "Event Recognition for Meaningful Human-Computer Interaction In a Smart Environment". in *the eNTERFACE'07 Workshop on Multimodal interfaces*, 2007, pp. 71-86.

203. M. Moshier, "Extensions to unification grammar for the description of programming languages", Ph.D., University of Michigan Michigan 1988.

204. E. Murphy-Chutorian and M.M. Trivedi, "Head Pose Estimation and Augmented Reality Tracking: An Integrated System and Evaluation for Monitoring Driver Awareness", *IEEE Transactions on Intelligent Transportation Systems*, vol. 11(2), pp. 300-311, 2010.

205. J.M.J. Murre, R.H. Phaf, and G. Wolters, "Original Contribution: CALM: Categorizing and learning module", *Neural Netw.*, vol. 5(1), pp. 55-82, 1992.

206. J.G. Neal and S.C. Shapiro, "Intelligent multi-media interface technology", in *Intelligent user interfaces*, ACM. 1991, pp. 11-43.

207. G. Nejat and M. Ficocelli, "Can I be of assistance? The intelligence behind an assistive robot". in *IEEE International Conference on Robotics and Automation*, 2008, pp. 3564-3569.

208. J.A. Nelder and R. Mead, "A simplex method for function minimization", *Computer*, vol. 7, pp. 308-313, 1965.

209. NEURO Technology (2008). VeriLook Standard SDK and Extended SDK. Available from: http://www.neurotechnology.com/vl_sdk.html.

210. K. Nickel, E. Scemann, and R. Stiefelhagen, "3D-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario". in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 565-570.

211. K. Nickel and R. Stiefelhagen, "Real-Time Person Tracking and Pointing Gesture Recognition for Human-Robot". in *LNCS*, 2004, pp. 1 - 6

212. K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction", *Image and Vision Computing*, vol. 25(12), pp. 1875-1884 2007.

213. L. Nini, L. Jiwen, and T. Yap-Peng, "Joint Subspace Learning for View-Invariant Gait Recognition", *Signal Processing Letters, IEEE*, vol. 18(7), pp. 431-434, 2011.

214. S.R.G.a.M.S. Nixon, "A dual active contour for head and boundary extraction". in *IEE Colloquium on Image Processing for Biometric Measurement*, 1994, pp. 1-4.

215. N. Oliver, A. Pentland, and F. Berard, "LAFTER: A real-time face and lips tracker with facial expression recognition", *Pattern Recog.*, vol. 33, pp. 1369–1382, 2000.

216. OpenCVWiki (2010). Motion Analysis and Object Tracking. [Online]. Available from: http://opencv.willowgarage.com/documentation/cpp/motion_analysis_and_object_tracking.html?highlight=optical%20flow.

217. OpenCVWiki (2011). Face Detection using OpenCV. [Online]. Available from: http://opencv.willowgarage.com/wiki/FaceDetection.

218. E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection". in *International Conference on Computer Vision and Pattern Recognition*, 1997, pp. 130-136.

219. S. Oviatt, "Multimodal interactive maps: designing for human performance", *Hum.-Comput. Interact.*, vol. 12(1), pp. 93-129, 1997.

220. S. Oviatt, "Ten myths of multimodal interaction", *Commun. ACM*, vol. 42(11), pp. 74-81, 1999.

221. S. Oviatt, A. DeAngeli, and K. Kuhn, "Integration and synchronization of input modes during multimodal human-computer interaction". in *Proceedings of the SIGCHI conference on Human factors in computing systems*, Atlanta, Georgia, United States, ACM, 1997, pp. 415-422.

222. Panasonic (2011). Panasonic D-Imager. [Online]. Available from: http://panasonic-electric-works.net/D-IMager/.

223. C.-B. Park and S.-W. Lee, "Real-time 3D pointing gesture recognition for mobile robots with cascade HMM and particle filter", *Image and Vision Computing*, vol. 29, pp. 51-63, 2010.

224. C.-B. Park, M.-C. Roh, and S.-W. Lee, "Real-Time 3D Pointing Gesture Recognition in Mobile Space". in *8th IEEE International Conference on Automatic Face & Gesture Recognition*, 2008, pp. 1-6.

225. H.M. Paterson, F.E. Pollick, and A.J. Sanford, "The Role of Velocity in Affect Discrimination". in *the Twenty-Third Annual Conference of the Cognitive Science Society*, 2001, pp. 756-761.

226. J. Penne, et al., "Robust real-time 3D respiratory motion detection using time of flight cameras", *International journal of computer assisted radiology and surgery*, vol. vol. 3(no5), pp. 427-431 2008.

227. A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition". in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 84-91.

228. R.W. Picard, *Affective Computing* Cambrige MA: MIT Press. 1997.

229. R. Plutchik, "Emotions: A general psychoevolutionary theory", in *Approaches to Emotion*, Ed. K. Scherer and P. Ekman, Lawrence Erlbaum Associates: NJ. 1984.

230. R. Plutchik, *Emotion: a psychoevolutionary synthesis*. New York: Harper and Row. 1989.

231. PMD Technologies (2009). PMD CamCube3.0. [online]. Available from: http://www.pmdtec.com/products-services/pmdvisionr-cameras/pmdvisionr-camcube-30/.

232. PMD Technologies (2009). PMD Technologies. [Online]. Available from: http://www.pmdtec.com/.

233. PMDTechnologies GmbH, *PMD[Vision] CamCube 3.0*. 2010.

234. Point Grey Research Inc (2010). Bumblebee camera [Online]. Available from: http://www.ptgrey.com/.

235. PrimeSensor (2010). The PrimeSensor™ Reference Design 1.08. [Online]. Available from: http://www.primesense.com/?p=514.

236. Z. Qian and D. Xu, "Research Advances in Face Recognition". in *Chinese Conference on Pattern Recognition*, 2009, pp. 1-5.

237. L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *Proceedings of the IEEE*, vol. 77(2), pp. 257-286, 1989.

238. M. Reale, T. Hung, and Y. Lijun, "Pointing with the eyes: Gaze estimation using a static/active camera system and 3D iris disk model". in *IEEE International Conference on Multimedia and Expo (ICME)*, 2010, pp. 280-285.

239. R. Reulke, "Combination of distance data with high resolution images", *Image Rochester NY*, vol. 2, pp. 1-6, 2006.

240. J. Richarz, et al., "There You Go! - Estimating Pointing Gestures In Monocular Images For Mobile Robot Instruction". in *15th IEEE International Symposium on Robot and Human Interactive Communication*, 2006, pp. 546-551.

241. J. Richarz, et al., "A monocular pointing pose estimator for gestural instruction of a mobile robot", *International Journal of Advanced Robotic Systems*, pp. 139–150, 2007.

242. T. Ringbeck and B. Hagebeuker, "A 3d time of flight camera for object detection". in *Optical 3-D Measurement Techniques* ETH Zürich, 2007.

243. R. Ros, et al., "Solving ambiguities with perspective taking". in *5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010, pp. 181-182.

244. G. Russ, B. Sallans, and H. Hareter, "Semantic based information fusion in a multimodal interface". in *International Conference on human-computer interaction*, Las Vegas, Nevada, USA, 2005, pp. 94-102.

245. J.A. Russell, " Reading emotions from and into faces: Resurrecting a dimensional-contextual perspective", in *The Psychology of Facial Expression* Cambridge University Press: Cambridge. 1997.

246. T. Sakai, M. Nagao, and T. Kanade, "Computer Analysis and Classification of Photographs of Human Faces". in *First USA-JAPAN Computer Conference*, 1972, pp. 55-62.

247. J. Salvi, J. Pages, and J. Batlle, "Pattern codification strategies in structured light systems", *Pattern Recognition*, pp. 827-849, 2004.

248. E. Sato, S. Sakurai, and A. Nakajima, "Context-based interaction using pointing movements recognition for an intelligent home service robot". in *16th IEEE*

International Conference on Robot & Human Interactive Communication, 2007, pp. 854-859.

249. E. Sato, T. Yamaguchi, and F. Harashima, "Natural Interface Using Pointing Behavior for Human–Robot Gestural Interaction", *IEEE Transactions on Industrial Electronics*, vol. 54(2), pp. 1105-1112, 2007.

250. Y. Sato, Y. Kobayashi, and H. Koike, "Fast Tracking of Hands and Fingertips in Infrared Images for Augmented Desk Interface". in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE Computer Society, 2000, pp. 462-467.

251. B. Schauerte, J. Richarz, and G.A. Fink, "Saliency-based Identification and Recognition of Pointed-at Objects". in *International Conference on Intelligent Robots and Systems (IROS)*, Taipei, TaiWan, 2010, pp. 4638-4643.

252. S.R. Schmidt-Rohr, M. Losch, and R. Dillmann, "Human and robot behavior modeling for probabilistic cognition of an autonomous service robot". in *17th IEEE International Symposium on Robot and Human Interactive Communication*, 2008, pp. 635-640.

253. seeingmachines (2000). faceLAB. [Online]. Available from: http://www.seeingmachines.com/product/facelab/.

254. C. Shan, S. Gong, and P.W. McOwan, "Beyond Facial Expressions: Learning Human Emotion from Body Gestures". in *Proceedings of BMVC*, 2007, pp. 1-8.

255. K. Shanmukh and A.K. Pujari, "Volume intersection with optimal set of directions", *Pattern Recognition Letters*, vol. 12(3), pp. 165-170, 1991.

256. R. Sharma, V.I. Pavlovic, and T.S. Huang, "Toward Multimodal Human-Computer Interface". in *Proceedings of the IEEE special issue on Multimedia Signal Processing*, 1998, pp. 853-869.

257. J. Shawe-Taylor and N. Cristianini, *Support Vector Machines and other kernel-based learning methods*: Cambridge University Press. 2000.

258. B. Shneiderman, *Leonardo's Laptop: Human Needs and the New Computing Technologies*. Cambridge: MIT Press. 2002.

259. C.-F. Shu, et al., "a open and extensible framework for event based surveillance". in *Advanced Video and Signal Based Surveillance, 2005.*, 2005, pp. 318-323.

260. L.C.D. Silva, K. Aizawa, and M. Hatori, "Detection and tracking of facial features by using a facial feature model and deformable circular template", *IEICE Trans. Inform. Systems*, pp. 1195–1207, 1995.

261. P.R.D. Silva and N. Bianchi-Berthouze, "Modeling human affective postures: an information theoretic characterization of posture features", *Computer animation and virtual worlds*, vol. 15, pp. 269–276, 2004.

262. P.R.D. Silva, A. Kleinsmith, and N. Bianchi-Berthouze, "Towards Unsupervised Detection of Affective Body Posture Nuances". in *Lecture Notes in Computer Science*, 2005, pp. 32-39.

263. P.R.D. Silva, M. Osano, and A. Marasinghe, "Towards recognizing emotion with affective dimensions through body gestures". in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition* 2006, pp. 269-274.

264. L. Snidaro, C. Micheloni, and C. Chiavedale, "Video security for ambient intelligence", *IEEE transactions on systems, man, and cybernetics*, vol. 35, pp. 133-144, 2005.

265. S. Soutschek, et al., "3-D Gesture-Based Scene Navigation in Medical Imaging Applications Using Time-Of-Flight Cameras". in *EEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1-6.

266. T. Starner and A. Pentland, "Real-time American sign language recognition from video using hidden markov models". in *the International Symposium on Computer Vision*, Washington DC, USA, 1995, pp. 265-270.

267. B. Stenger, P.R.S. Mendonca, and R. Cipolla, "Model-based 3D tracking of an articulated hand". in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 310-315.

268. Stiefelhagen and Fugen, "Natural human-robot interaction using speech, head pose and gestures", *Proceedings of 2004 IEEE/RSJ international conference on Intelligent Robots and Systems*, pp. 2422-2427, 2004.

269. R. Stiefelhagen, et al., "Enabling Multimodal Human-Robot Interaction for the Karlsruhe Humanoid Robot", *IEEE Transactions on Robotics*, vol. 23(5), pp. 840-851, 2007.

270. D.J. Sturman and D. Zeltzer, "A survey of glove-based input", *Computer Graphics and Applications*, vol. 14(1), pp. 30-39, 1994.

271. O. Sugiyamal, et al., "Three-Layer Model for Generation and Recognition of Attention-Drawing Behavior". in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, beijing, China, 2006, pp. 5843 - 5850.

272. Y. Sumi and Y. Ohta, "Detection of face orientation and facial components using distributed appearance modelling". in *Int. Workshop on Automatic Face- and Gesture-Recognition*, 1995, pp. 254-255.

273. J. Sun, H.-Y. Shum, and N.-N. Zheng, "Stereo Matching Using Belief Propagation". in *European Conference of Computer Vision*, 2002, pp. 510-524.

274. S. Tamura and S. Kawasaki, "Recognition of sign language motion images", *Pattern Recogn.*, vol. 21(4), pp. 343-353, 1988.

275. H.T. Tanaka, M. Ikeda, and H. Chiaki, "Curvature-based face surface recognition using spherical correlation. Principal directions for curved object recognition". in *Third International Conference on Automated Face and Gesture Recognition*, 1998, pp. 372-377.

276. R. Tanawongsuwan and A. Bobick, "Gait recognition from time-normalized joint-angle trajectories in the walking plane". in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 726-731.

277. T. Teixeira, G. Dublon, and A. Savvides (2010), A Survey of Human-Sensing: Methods for Detecting Presence, Count, Location, Track, and Identity. ACM computing surveys, [Online]. Available from: http://www.eng.yale.edu/enalab/publications/human_sensing_enalabWIP.pdf.

278. Y.-L. Tian, T. Kanade, and J. Cohn, "Robust Lip Tracking by Combining Shape, Color and Motion". in *the 4th Asian Conference on Computer Vision*, 2000, pp. 1040-1045.

279. A.S. Tolba, A.H. El-Baz, and A.A. El-Harby, "Face Recognition: A Literature Review", *International Journal of Information and Communication Engineering*, pp. 88-103, 2006.

280. M. Turk and A. Pentland, "Face recognition using eigenfaces.". in *Computer Vision and Pattern Recognition.*, 1991, pp. 586-591.

281. VICON (2008). Motion Capture System from Vicon. [Online]. Available from: www.vicon.com/.

282. V. Vinayagamoorthy, M. Slater, and A. Steed, "Emotional Personification of Humanoids in Immersive Virtual Environments", Department of Computer Science, University College London., 2002.

283. P. Viola and M.J. Jones, "Robust Real-Time Face Detection", *Int. J. Comput. Vision*, vol. 57(2), pp. 137-154, 2004.

284. H. Wang, H. Prendinger, and T. Igarashi, "Communicating emotions in online chat using physiological sensors and animated text". in *CHI '04 extended abstracts on Human factors in computing systems*, Vienna, Austria, ACM, 2004, pp. 1171-1174.

285. J.J. Wang, et al., *3D Landmarks Extraction from a Range Imager Data for SLAM*, in *ACRA* 2009: Sydney.

286. Z. Wang, et al., "Camshift Guided Particle Filter for Visual Tracking". in *IEEE Workshop on Signal Processing Systems*, 2007, pp. 301-306.

287. C.M. Whissell, *The dictionary of affect in language*. Emotion: Theory, Research, and Experience, ed. R.P.a.H. Kellerman. New York: Academic Press. 1989.

288. Willow Garage (2011). PR2. [Online]. Available from: http://www.willowgarage.com/pages/pr2/overview.

289. A.D. Wilson and A.F. Bobick, "Recognition and interpretation of parametric gesture". in *Computer Vision, 1998. Sixth International Conference on*, 1998, pp. 329-336.

290. W. Woo, J. Park, and Y. Iwadate, "Emotion analysis from dance performance using time-delay neural networks". in *JCIS-CVPRIP* 2000, pp. 374-377.

291. A. Wu, M. Shah, and N.d.V. Lobo, "A Virtual 3D Blackboard: 3D Finger Tracking Using a Single Camera". in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE Computer Society, 2000, pp. 536-543.

292. Y. Yamamoto, I. Yoda, and K. Sakaue, "Arm-pointing Gesture Interface Using Surrounded Stereo Cameras System". in *Proceedings of the 17th International Conference on Pattern Recognition*, 2004, pp. 965-970.

293. K.M.L.a.H. Yan, "Facial feature location and extraction for computerised human face recognition". in *Int. Symposium on information Theory and Its Applications*, Sydney, Australia, 1994.

294. G. Yang, "License plate character recognition based on wavelet kernel LS-SVM". in *Computer Research and Development (ICCRD)*, 2011, pp. 222-226.

295. J. Yang and A. Waibel, "A real-time face tracker". in *the 3rd Workshop on Applications of Computer Vision*, 1996, pp. 142-147.

296. T.S.H. Ying Wu, "Vision-Based Gesture Recognition: A Review", *Lecture Notes in Computer Science* vol. 1739, pp. 103-115 1999.

297. B. Yoo, et al., "3D user interface combining gaze and hand gestures for large-scale display". in *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, Atlanta, Georgia, USA, ACM, 2010, pp. 3709-3714.

298. K.C. Yow and R. Cipolla, "Feature-based human face detection". in *Proceedings of the Twenty-First National Radio Science Conference*, 1997, pp. 1-10.

299. A.L. Yuille, P.W. Hallinan, and D.S. Cohen, "Feature extraction from faces using deformable templates", *Int. J. Comput. Vision*, vol. 8, pp. 99–111, 1992.

300. A.L. Yuille, P.W. Hallinan, and D.S. Cohen, "Feature extraction from faces using deformable templates", *Int. J. Comput. Vision*, vol. 8, pp. 99-111, 1992.

301. N.A. Zaidi and D.M. Squire, "Local Adaptive SVM for Object Recognition". in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2010, pp. 196-201.

302. Q. Zhang, et al., "A gaze and speech multimodal interface". in *24th International Conference on Distributed Computing Systems Workshops*, 2004, pp. 208-213.

303. J. Zhu, et al., "Reliability Fusion of Time-of-Flight Depth and Stereo for High Quality Depth Maps", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 33(7), pp. 1400-1415, 2011.

304. Y. Zhu, B. Dariush, and K. Fujimura, "Controlled human pose estimation from depth image streams". in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1-8.