

USER MODELLING FOR PERSONALISED DRESSING
ASSISTANCE BY HUMANOID ROBOTS USING
MULTI-MODAL INFORMATION

YIXING GAO

Thesis submitted for the degree of Doctor of Philosophy

Supervised by PROFESSOR YIANNIS DEMIRIS

Personal Robotics Lab

Department of Electrical and Electronic Engineering

Imperial College London

September 2016

COPYRIGHT DECLARATION

The copyright of this thesis rests with the author and is made available under a **Creative Commons Attribution Non-Commercial No Derivatives** licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

ORIGINALITY DECLARATION

I hereby declare that this thesis and the work herein detailed, was composed and originated by myself, except where appropriately referenced and credited.

London, September 2016

Yixing Gao

To my family.

ABSTRACT

To enable personalised assistance, assistive robots benefit from building a user-specific model, so that the assistance is customised to the particular set of user abilities. Among various tasks in home environments, assistive dressing, which is greatly beneficial to people with upper-body movement limitations, remains a challenging task for humanoid robots. In this thesis, we aim to design, implement, and evaluate user modelling methods which can enable humanoid robots to provide personalised dressing assistance.

We begin by proposing a user modelling method using vision information. We use Gaussian mixture models (GMMs) to model the movement space of the human upper-body joints to learn the reachable area of each joint. We enabled a Baxter humanoid robot to plan its dressing motion using the GMMs of the human joints and real-time pose estimation. The dressing assistance is personalised by fulfilling a reachability criterion.

To compensate for the disadvantages of using vision information only, we proposed an online iterative path optimisation method based on adaptive moment estimation. We enabled the Baxter robot to search for the optimal personalised dressing path for human users using force information. The dressing assistance is personalised by fulfilling a comfort criterion.

Finally, to enable personalised dressing assistance fulfilling both the reachability and the comfort criteria, we proposed a user modelling method using multi-modal information by combining the GMMs of the human upper-body joints with the online iterative path optimisation. Experiments on both the synthetic dataset and the real-world assistive dressing data showed that the proposed method can achieve a balance between the two criteria when searching for the optimal path.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Yiannis Demiris. He is the one who inspires me to work on personalised assistive dressing, the research topic which brings a lot of inspiration to me. Although there are ups and downs during my PhD study, Prof. Demiris always gives me positive feedback when I make some achievement. It would have been impossible to finish this thesis without him. I am forever grateful to him for all the help.

I would like to thank my parents, Yuntian Gao and Xuehui Wang, for their unconditional love and support throughout my life. It was my father who insisted on letting me go aboard four years ago. Although there was hesitation, now I realise this is the correct decision I have made in my life. It is my mother who always listens to me, shares my happiness and sadness. I feel so lucky and proud to be their daughter.

I would like to thank my husband Yun Fu for his love, care, and support. The luckiest thing in my life is to meet him when we visited Buckingham Palace together three years ago. I would like to thank him for understanding and encouraging me whenever I have a difficult time in study, accompanying me to work on weekends or stay late at the college, cooking various kinds of delicious meal, and bringing all the happiness time. This thesis would have been impossible without his love.

I would like to thank my friends and colleagues in the Personal Robotics Lab. I would like to especially thank Hyung Jin Chang for his patient instructions and valuable suggestions during my PhD study. I would also like to thank Yan Wu for his help at the beginning of my PhD study. I would like to thank Miguel Sarabia for teaching me various programming skills and Di-

mitrios Korkinof for being an interesting lab neighbour. Also, I would like to thank Yanyu Su, Ayse Kucukyilmaz, Kyuhwa Lee, Harold Soh, Tobias Fischer, Maxime Petit, Antoine Cully, Martina Zambelli, Theodosios Georgiou, Fan Zhang, Arturo Ribes, Joshua Elsdon, Eris Chinellato, Dimitri Ognibene, Raquel Ros, Alexandre Coninx, Ahmed Al-Hindawi, Mark Zolotas, Ruohan Wang, and Caterina Buizza. I enjoy all the time that we spent together to work, discuss, and celebrate inside and outside the lab. I would also like to thank my friends in the EEE department and the college.

CONTENTS

1	INTRODUCTION	15
1.1	Motivation	16
1.2	Contributions	18
1.3	Thesis Outline	19
1.4	Publications	22
2	BACKGROUND	23
2.1	Socially Assistive Robots	24
2.1.1	Assistive Robots in General	24
2.1.2	Dressing Robots	26
2.2	User Modelling	28
2.2.1	Personalisation and User Modelling in General	28
2.2.2	User Modelling in Robotics	32
2.3	Vision in Robotics	35
2.3.1	Human-centred Vision	35
2.3.2	Vision in Human-robot Interaction	38
2.4	Robot Learning using Multi-modal Information	41
2.4.1	General Sensors for Human	41
2.4.2	Human-robot Interaction using Multi-modal Information	43
2.5	Robot Path Planning	45
2.6	Conclusion	46
3	USER MODELLING USING VISION INFORMATION	49
3.1	System Overview	49
3.2	Upper-body Pose Recognition	53
3.2.1	Randomised Decision Forests	53

3.2.2	Labelling Training Images	55
3.3	Upper-body Movement Space Modelling	57
3.4	Personalised Assistive Dressing using Vision Information	59
3.5	Experiments	61
3.5.1	Evaluation of Upper-body Pose Recognition and Movement Space Modelling	62
3.5.2	Assisting Users with Dressing	66
3.6	Conclusion	69
4	USER MODELLING USING FORCE INFORMATION	71
4.1	Overview	74
4.2	Robot Motion Adjustment Based on Force Information	75
4.3	Proposed Method	79
4.3.1	Adaptive Moment Estimation	79
4.3.2	Online Iterative Path Optimisation	79
4.4	Personalised Assistive Dressing	83
4.4.1	Movement Space Modelling of Human Hands	83
4.4.2	Detecting External Force Resistance	84
4.4.3	Learning Optimal Personalised Dressing Path	86
4.5	Experiments	87
4.5.1	Synthetic Dataset	87
4.5.2	Real-world Personalised Assistive Dressing	92
4.6	Conclusion	111
5	USER MODELLING USING VISION AND FORCE INFORMATION	113
5.1	Proposed Method	114
5.1.1	Online Iterative Path Optimisation using Multi-modal Information	114
5.1.2	Stick Model for Upper-body Joints	119
5.2	Experiments	120
5.2.1	Synthetic Dataset	121

5.2.2	Real-world Personalised Assistive Dressing	132
5.3	Conclusion	138
6	CONCLUSIONS AND FUTURE WORK	141
6.1	Summary of Contributions	141
6.2	Limitations	142
6.3	Future Work	147
A	APPENDIX	151
A.1	Robot Platform	151
A.2	MoveIt!	152
	BIBLIOGRAPHY	153

LIST OF FIGURES

Figure 1.1	Roadmap of the thesis.	21
Figure 3.1	System overview in Chapter 3.	50
Figure 3.2	System illustration in Chapter 3.	51
Figure 3.3	Colour-based segmentation for labelling training data.	56
Figure 3.4	An illustration of the robot dressing motion from a top-view.	61
Figure 3.5	Extracting the 3D joint coordinates from a testing depth image.	64
Figure 3.6	Visualisation of 4 representatives of the GMMs of the human upper-body joints.	65
Figure 3.7	Sequential shots 1 of personalised assistive dressing with the user modelling method in Chapter 3.	67
Figure 3.8	Sequential shots 2 of personalised assistive dressing with the user modelling method in Chapter 3.	68
Figure 4.1	Organisation of the sections in Chapter 4.	72
Figure 4.2	An illustration of the user modelling method using force information in Chapter 4.	73
Figure 4.3	An example of the local motion adjustment of the robot's right gripper path using force information.	77
Figure 4.4	The force coordinates of robot's grippers and the robot's coordinates.	85
Figure 4.5	8 examples of iteration process with the synthetic dataset using the proposed iterative path optimisation method in Chapter 4.	89
Figure 4.6	Changes in energy against iterations.	96

Figure 4.7	Changes in execution time against iterations.	96
Figure 4.8	The legend used in Figure 4.9, 4.10, 4.11, 4.12, 4.13, and 4.14.	97
Figure 4.9	A complete path update. 1 st iteration: part A.	100
Figure 4.10	A complete path update. 1 st iteration: part B.	101
Figure 4.11	A complete path update. 2 nd iteration: part A.	102
Figure 4.12	A complete path update. 2 nd iteration: part B.	103
Figure 4.13	A complete path update. 3 rd iteration: part A.	104
Figure 4.14	A complete path update. 3 rd iteration: part B.	105
Figure 4.15	The energy of each iteration.	105
Figure 4.16	An example of the path iteration with real assistive dressing data.	106
Figure 4.17	Sequential shots of personalised assistive dressing with the user modelling method in Chapter 4.	107
Figure 5.1	3D illustration of a stick model in Chapter 5.	120
Figure 5.2	Experimental result 1 with synthetic dataset using the proposed method in Chapter 5.	125
Figure 5.3	Experimental result 2 with synthetic dataset using the proposed method in Chapter 5.	126
Figure 5.4	Comparison result 1 with synthetic dataset among the proposed methods in Chapter 3, 4, and 5.	128
Figure 5.5	Comparison result 2 with synthetic dataset among the proposed methods in Chapter 3, 4, and 5.	129
Figure 5.6	Comparison result 3 with synthetic dataset among the proposed methods in Chapter 3, 4, and 5.	130
Figure 5.7	Comparison result 4 with synthetic dataset among the proposed methods in Chapter 3, 4, and 5.	131

Figure 5.8	Comparison result 1 with real assistive dressing data among the proposed user modelling methods in Chapter 3, 4, and 5. 135
Figure 5.9	Comparison result 2 with real assistive dressing data among the proposed user modelling methods in Chapter 3, 4, and 5. 136
Figure 5.10	Comparison result 3 with real assistive dressing data among the proposed user modelling methods in Chapter 3, 4, and 5. 137
Figure A.1	The Baxter humanoid robot we use for the assistive dressing experiments in this thesis. 152

LIST OF TABLES

Table 3.1	Robot motion planning procedures. 60
Table 3.2	Average classification accuracy of testing images with different tree models. 63
Table 4.1	Evaluation of synthetic dataset with the proposed iterative path optimisation method in Chapter 4. 90
Table 4.2	Evaluation of real assistive dressing data with the user modelling method in Chapter 4. 94

INTRODUCTION

Due to the rapid development in the field of service robotics, assistive humanoid robots have been increasingly used in various applications in human lives, such as companion robots for elderly people, cooking robots, shopping assistants, and robots that can clean and tidy rooms. Although some tasks can be pre-programmed by software engineers, a real intelligent humanoid robot should not only possess the ability to learn from human-robot interactions, but also be able to adapt its behaviour to different human users and changing environments. There has been interesting prior research that enables robots to learn motor skills and generalise the learned skills to different environments, such as imitation learning, also termed as learning from human demonstrations, and research that enables robots to learn human preferences during human-robot interactions and adapt their behaviours to the preferences of the users. This thesis investigates the problems involved in building user models using multi-modal information during human-robot interactions, and enabling humanoid robots to provide personalised assistance to individual users by using and updating user models online.

Undoubtedly, there is not a universal user modelling method for all kinds of robotic applications. How to build user models is closely related to the aims of the applications. It is challenging because building user models is not the final goal, the user models should be utilised by the robots to provide personalised assistance in real-world human-robot interactions. Interesting prior research on user modelling is addressing robotic applications, such as robots handing over objects to humans and robots approaching humans from different directions. To build user models, sensor information about

the human state is required. Due to environment noise, multiple sensors can provide more robust information compared to a single sensor. However, how to make use of multi-sensor information in building user models becomes another challenging problem. This thesis presents a user modelling method using multi-modal information and enables humanoid robots to provide personalised dressing assistance to real human users.

1.1 MOTIVATION

Assistive robots in home environments have gained significant popularity, not only because of the increasingly sophisticated manufacturing of robots and the rapid development of artificial intelligence, but also due to a huge potential to reduce the need for human labour in daily care, especially considering the ageing problem (Broekens et al., 2009; Fasola and Mataric, 2012; Flandorfer, 2012; Schroeter et al., 2013; Tapus, Maja and Scassellatti, 2007). However, people vary significantly in their skill sets, culture, habits, behaviours, *etc.*, and these factors affect their choices and preferences in human-robot interactions. For a widespread use of home-environment assistive robots in the future, the ability to provide personalised assistance has become one of the key issues.

Dressing is one of the most common daily activities for humans. Elderly people or people with upper-body movement limitations can greatly benefit from assistive dressing. Although recent studies have enabled assistive robots to perform some daily tasks in home environments, assistive dressing remains a challenging task for robots and very little effort has been put on this particular problem. To provide personalised dressing assistance, the first idea that this thesis puts forward is to enable the robot to build a model of the movement space of the human's arms. By recognising human upper-body pose in real time, human motion information can be recorded and

modelled. By learning the movement space of the human's arms, the robot can avoid placing clothes in the areas which are not reachable to the human users. Different people may have different reachable areas for the upper body, due to the difference in the physical characteristics, such as the human's heights and the lengths of the human arms, and the difference in the upper-body movement limitations, such as difficulties in lifting arms, extending arms or turning towards a certain direction.

However, using vision information alone can suffer from environment noise, such as occlusions. Additionally, vision information can not reflect another significant factor during assistive dressing, which is the human's comfort. Thus the second idea for personalised dressing assistance is to use force sensor information for the robot to locally adjust its motion by detecting external resistance. Due to the consideration that humans tend to follow certain behaviour patterns for their daily activities, we take a further step to enable the robot to iteratively learn the optimal personalised dressing path for a user using force information.

Assistive dressing can be personalised based on either a reachability criterion using vision information or a comfort criterion using force information. We hypothesise that by combining the vision information with the force information, the robot can achieve a balance between the two criteria. We expect that the movement space information of human arms can help with the process of searching for the optimal personalised dressing path, by making sure that the search happens within the reachable area of the human arms.

To realise the above ideas for personalised assistive dressing, the following goals should be achieved:

- The robot should be able to use vision information to recognise human upper-body pose in real time, model the movement space of human arms, and provide personalised dressing assistance.

INTRODUCTION

- The robot should be able to use force sensor information to detect resistance during human-robot interactions, locally adjust its motion, and iteratively search for the optimal personalised dressing path.
- The robot should be able to provide personalised dressing assistance by fulfilling both the reachability and comfort criteria.

To address these challenges, we outline our contributions in section [1.2](#).

1.2 CONTRIBUTIONS

This thesis contributes to the research domains of user modelling, assistive dressing, and human-robot interaction.

- User modelling plays a significant role in human-robot interaction, in which it enables robots to adapt their behaviours to the preferences of different users. Existing user modelling methods in robotics mainly focus on scenarios such as objects hand-over by robots, robots approaching human users, and companion robots in the home environment. However, there is much less work on user modelling in the domain of assistive dressing. In this thesis, we proposed user modelling methods using different sensor information to enable humanoid robots to provide personalised dressing assistance to different users.
- Among various daily tasks in the home environment, dressing is one of the most common daily activities for the human. Elderly people or people with upper-body movement limitations can greatly benefit from assistive dressing. Existing research work on assistive dressing focus on either a sub-part of dressing assistance, such as estimation of the human-clothes spatial relationships, outcome classification of the dressing process or enabling robots to learn the motor skills of assistive dressing. However, experiments were mainly conducted with human

mannequins or in the simulation. In this thesis, we enabled humanoid robots to provide dressing assistance to real human users.

- Most of the time, the human pose is recognised with a front-view camera. In our work, we recognised the real-time human upper-body pose with a top-view depth sensor. This idea can be applied to pose estimation for self-feeding people or wheelchair users. Besides, we proposed to model the movement space of the human upper-body joints in a probabilistic way, which can be applied to evaluate the improvement of the human upper-body mobility in rehabilitation robotics or medical robotics. Although current research work has enabled humanoid robots to perform various tasks, less attention has been paid to tasks such as assistive dressing. In this thesis, we present an end-to-end approach to enable humanoid robots to provide dressing assistance.

1.3 THESIS OUTLINE

In this section, we present an outline of the rest of this thesis along with a roadmap shown in Figure 1.1.

- **Chapter 2 - Background** presents related work and background information in socially assistive robots, user modelling, vision in robotics, robot learning using multi-modal information, and robot path planning. We also discuss the relevance of the reviewed work to this thesis.
- **Chapter 3 - User Modelling using Vision Information** presents a movement space modelling method of the human upper-body joints using Gaussian mixture models (GMMs). We also present the utility of randomised decision forests to recognise human upper-body pose in real time with a top-view depth sensor. We enable assistive robots to provide personalised dressing assistance according to the GMMs

of the human upper-body joints and real-time upper-body pose estimation. We evaluated the proposed system with ten healthy participants. We first evaluated the average classification accuracy of the human upper-body pose estimation with different tree models. We also demonstrated how to model the movement space of the human upper-body joints by visualising the GMMs of the human body. Experimental results showed that the Baxter robot successfully assisted all the participants with their dressing.

- **Chapter 4 - User Modelling using Force Information** presents an online iterative path optimisation method based on adaptive moment estimation. We compared the proposed method with methods using vanilla SGD update, momentum update, Adagrad, and RMSProp on the synthetic dataset. Experimental results showed that the performance of the proposed method achieved the smallest error with fewer iterations and less computation time. We also evaluated the proposed method on the real-world assistive dressing data by enabling the Baxter robot to dress human users using force information. For all the twelve participants, the robot found the optimal dressing paths within a maximum of five iterations.
- **Chapter 5 - User Modelling using Vision and Force Information** presents an online iterative path optimisation method using multi-modal information, by combining the GMMs of the human upper-body joints in Chapter 3 with the online iterative path optimisation in Chapter 4. During iteration, a stick model is used for calculating the joint Gaussian probability of two arm joints connected by the stick. With both the synthetic dataset and the real-world assistive dressing data, we compared the experimental results among the proposed user modelling methods in Chapter 3, 4, and 5. The comparison results showed that while the user modelling method in Chapter 3 can mainly

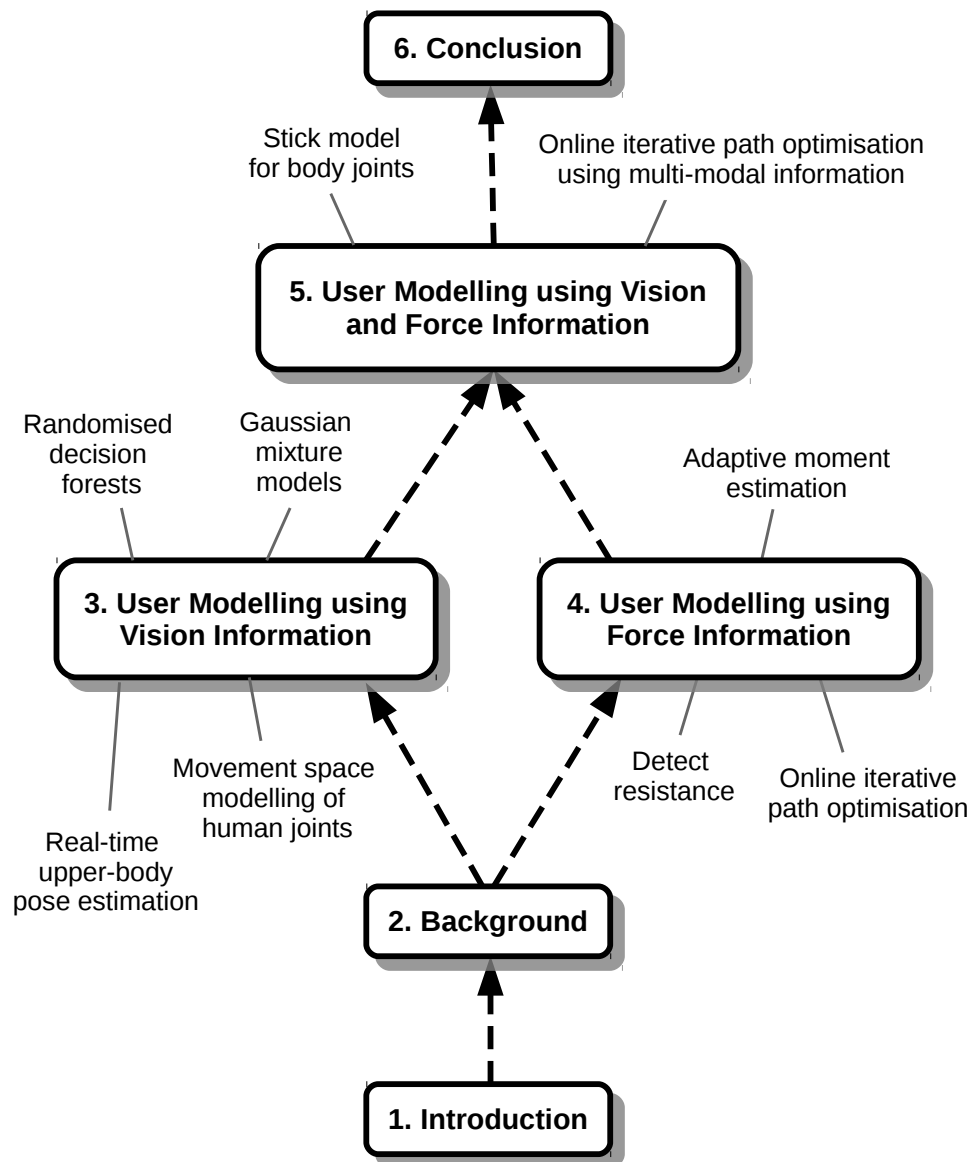


Figure 1.1: This roadmap shows the overall organisation of the thesis. Each box indicates a main chapter along with a few keywords.

fulfil a reachability criterion and the user modelling method in Chapter 4 can mainly fulfil a comfort criterion, the proposed method in Chapter 5 can achieve a balance between the reachability and the comfort criteria.

- **Chapter 6 - Conclusions and Future Work** concludes this thesis, discusses about the limitations and possible future work.

1.4 PUBLICATIONS

The publications resulting from this thesis are listed as follows.

- **Yixing Gao**, Hyung Jin Chang, and Yiannis Demiris: "User Modelling for Personalised Dressing Assistance by Humanoid Robots", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015 (oral). (Chapter 3)
- **Yixing Gao**, Hyung Jin Chang, and Yiannis Demiris: "Personalised Assistive Dressing by Humanoid Robots using Multi-modal Information", in IEEE ICRA Workshop on Human-Robot Interfaces for Enhanced Physical Interactions, 2016. (Chapter 4)
- **Yixing Gao**, Hyung Jin Chang, and Yiannis Demiris: "Iterative Path Optimisation for Personalised Dressing Assistance using Vision and Force Information", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016 (oral). (Chapter 4)

BACKGROUND

This thesis mainly presents a user modelling method using multi-modal information for humanoid robots to provide personalised assistance in home environments. We draw inspiration not only from the research field of human-robot interaction, but also computer vision and machine learning. Robot learning plays a significant role because it enables robots to adapt to changes in tasks or environments. Nowadays, more and more machine learning methods have been applied in robotics field to enable robots to learn how to perform tasks. Robots can either learn motions from human demonstrations (Wu et al., 2014), or learn their kinematic mechanism from motor babbling (Zambelli and Demiris, 2016). In this thesis, we are focused on assistive robot learning from human preferences, which we term as user modelling. The aim for user modelling is for assistive robots to provide personalised assistance to each user.

In this chapter, we review the related work and background knowledge of this thesis. We start in section 2.1 by generally looking at various kinds of socially assistive robots. In particular, we review recent research on assistive robots that can help human users to dress. Section 2.2 looks at work in user modelling methods in the fields of general computer science and robotics. In section 2.3, we look at recently proposed methods in human-centred vision and the vision methods applied in human-robot interaction. In section 2.4, an overview of robot learning using multi-modal information is presented, along with a quick look at different kinds of sensors that can be used in human-robot interaction. Section 2.5 looks at research work on robot motion path planning for humanoid robots and stochastic optimisation. Finally, sec-

tion 2.6 presents a conclusion of this chapter by discussing the relevance of the reviewed work to this thesis.

2.1 SOCIALLY ASSISTIVE ROBOTS

In this section, we review the research work on socially assistive robots. Section 2.1.1 first looks at various forms of assistive robots in human lives. Since the main robot application involved in this thesis is assistive dressing, we then review research work that enables humanoid robots to dress humans in section 2.1.2.

2.1.1 *Assistive Robots in General*

There have been different forms of assistive robots for various purposes. Assistive robots are mainly designed to assist or interact with humans physically or socially. Some researchers study small and soft robots, for instance, seal robots, to accompany elderly people (Kidd et al., 2006; Robinson et al., 2013; Wada and Shibata, 2007). Such robots mainly interact with humans through voice, language, and simple motions. Humanoid robots are also used as companion robots in home environments (Amirabdollahian et al., 2013; Lee et al., 2009; Torta et al., 2012). Such robots are usually equipped with multiple sensors so that they can possess various abilities, for instance, navigation in a smart home, human and object recognition, and complicated human-robot interactions using motions and languages. For all these different types of companion robots, the robots' social presences have been studied to analyse their psychological and physiological influences on the elderly (Bemelmans et al., 2012; Broadbent et al., 2009; Heerink et al., 2008).

Another research area in assistive robots is for post-stroke rehabilitation. Robotic manipulators are most commonly used to provide hands-on assist-

ance (Lum et al., 2002; Masiero et al., 2007). During the goal-directed movements, a robot manipulator applies forces to enable the affected human arm to follow the correct moving path. Instead of providing hands-on assistance, assistive robots mainly use voices or gestures to interact with humans by enabling them to accomplish different tasks using their affected limbs (Eriksson et al., 2005; Lo et al., 2010; Matarić et al., 2007; Tapus et al., 2008). Examples of rehabilitation tasks include limbs exercises, the Towers of Hanoi puzzle, and turning pages of a newspaper. Motion capture sensors or other vision sensors are often used to detect human motion, and the robot provides feedback to human users based on their performances. For instance, if the user fails the task, the robot would encourage the user to try again. If the user accomplishes the task, the robot would suggest a more complicated task for the user.

There have also been wearable robots for people with different disabilities, such as exoskeleton robots for human lower limbs (Banala et al., 2007, 2009; Ferris et al., 2005; Veneman et al., 2007), for human upper limbs (Carignan and Liszka, 2005; Cavallaro et al., 2006, 2005; Perry et al., 2007), for human hands (Ho et al., 2011; Iqbal et al., 2011; Kawasaki et al., 2007; Schabowsky et al., 2010) and for human whole-body (Marcheschi et al., 2011). These exoskeleton robots are used to provide rehabilitation aids or to assist humans with their affected limbs in daily living.

Smart wheelchair robots are widely studied to provide mobility assistance for children, adults, and elderly people (Simpson, 2005). Sensor information from humans are used for the shared control of wheelchairs movement, such as brain signals (Carlson and Millan, 2013; Galán et al., 2008; Rebsamen et al., 2010) and human gaze (Barea et al., 2002; Carlson and Demiris, 2012; Matsumoto et al., 2001). An interesting aspect of the shared control is that wheelchairs only provide assistance when needed. The issue of how to assist can be learned through imitation learning via a pair of haptic devices, where one

haptic device is controlled by a teacher to teach the wheelchair how to assist and another haptic device is controlled by a wheelchair user (Kucukyilmaz and Demiris, 2015; Soh and Demiris, 2013). For wheelchair users, especially children, to understand the wheelchair behaviours more clearly and directly, Sarabia and Demiris proposed a humanoid robot companion for wheelchair users (Sarabia and Demiris, 2013). A NAO robot which was attached in front of the smart wheelchair used voice and gesture information to inform the user about the behaviours of the wheelchair.

Considering the ageing population, more and more humanoid robots have been studied to provide assistance for elderly people in their daily living (Broadbent et al., 2009; Broekens et al., 2009; Tapus, Mataric and Scassellati, 2007). Yamazaki *et al.* enabled a humanoid robot to perform tidying and cleaning rooms tasks such as carrying a tray from one table to another, picking up clothes and putting it into a washing machine, and cleaning floors with a broom (Yamazaki et al., 2010). Humanoid robots are also used for providing cooking support in a kitchen (Gravot et al., 2006), providing massage for the human backs (Luo et al., 2014), acting as shopping assistants (Iwamura et al., 2011), or providing bathing care (King et al., 2010; Satoh et al., 2009).

2.1.2 Dressing Robots

Dressing is one of the most common daily activities for humans and providing dressing assistance remains a challenging problem for robots. There has been interesting prior research on assistive dressing by humanoid robots in home environments. There are 3 core components in assistive dressing, which are assistive robots, human users, and clothes.

Tamei *et al.* proposed to use the reinforcement learning method to teach a dual-arm robot to learn the dressing motions (Tamei et al., 2011). For a

human mannequin with a different head and shoulder inclination, the robot learned how to dress it with a T-shirt. At the beginning of the experiments, the mannequin's arms were inside the sleeves of the T-shirt while the robot held the hem. The dressing task for the robot was to pull the T-shirt over the mannequin's head. The topological relationships between the cloth and the mannequin were observed using the motion capture system for the robot to optimise its joints trajectories. Although the motion capture can provide accurate data, the system is complicated to use. To successfully detect the markers in the system, the markers should be visible and not occluded by the obstacles. To improve real-time estimation of the human-cloth relationships, Koganti *et al.* proposed the offline learning of a cloth dynamics model with different sensor data, and applied this learned model to track the human-cloth relationships online using a depth sensor (Koganti *et al.*, 2015). A shared Gaussian Process Latent Variable Model (GP-LVM) was used to learn a joint low-dimensional latent model for the motion capture data and depth sensor data. It was assumed that the cloth followed consistent dynamics during the dressing process. Their experiments were also conducted with a human mannequin.

Klee *et al.* used a Baxter robot to assist humans to put on a hat (Klee *et al.*, 2015). The human pose was recognised and tracked with a vision sensor for the robot to check if the user could reach certain goal pose. During human-robot interactions, limitations of the human body movement were gradually learned, for instance, how far the user could move towards the robot, or to the left/right side of the robot. The final goal for the robot was to choose a reachable area for the user to move and then assist the user in putting on the hat. Colome *et al.* enabled a WAM robot to wrap a scarf around a human mannequin's neck through reinforcement learning (Colomé *et al.*, 2015). Since friction became a factor which can no longer be ignored when wrapping a scarf, an analytical model of friction was proposed in order to improve

the robot’s Inverse Dynamic Model (IDM) for compliant control. There has also been interesting research work on humanoid robots manipulating non-rigid material, for instance, folding cloth (Bersch et al., 2011; Maitin-Shepard et al., 2010; Van Den Berg et al., 2010; Yamakawa et al., 2011) and unfolding cloth (Doumanoglou et al., 2014; Kita et al., 2011; Ramisa et al., 2012; Willimon et al., 2011).

2.2 USER MODELLING

In this section, we review the research work on user modelling. We first discuss why personalisation and user modelling are important in our lives, and we look at various kinds of research work that builds user models and their applications in section 2.2.1. Then we review some user modelling methods in robotics in section 2.2.2.

2.2.1 *Personalisation and User Modelling in General*

We are living in the computer age and we are surrounded by all kinds of electrical devices such as laptops, smart tablets, smart phones, and smart watches. With the development of software engineering, our user data can be collected when using these devices and we can be provided with more personalised information. The technologies behind these are data mining (Fayyad et al., 1996; Han et al., 2011; Hand et al., 2001; Witten and Frank, 2005) and machine learning (Bengio, 2009; Bishop, 2006; Michalski et al., 2013; Murphy, 2012).

Nowadays, we usually use a search engine to search for the information we are interested in the Internet. Our searching history and preferred searching information can be recorded and analysed by the search engine. Each time when we search, the search engine provides other news, music, videos,

etc. which it thinks we might be also interested. Another common example is the online social network platform. Based on our own profiles, the profiles of our friends, and the friend's profiles of our friends, the online social network platform could suggest friends we might know to us. This helps us save our time to search for different friends by ourselves.

One of the most notable features of the mobile apps used on smart electrical devices is personalisation. Mobile apps for exercises can record the total hours of exercise, the calories burned, the total amount of running, *etc.*, and use a week/month record to analyse the personal achievement to provide suggestions for the exercise goal in the next week/month. This principle is the same for the other mobile apps, such as balancing diet, brushing teeth, and listening to music and songs.

Personalisation is everywhere in our daily living. To provide personalised information or interaction, user data should be collected, modelled, and analysed. This whole process can be viewed as building user models or user modelling. The definition of user modelling according to (Fischer, 2001) is that it describes the process of building up and modifying a conceptual understanding of the user, where the main goal of user modelling is customisation and adaptation of systems to the user's specific needs. Depending on the different purposes of applications, various kinds of personal information can be collected to build user models. According to the user modelling studies in (Hothi and Hall, 1998; Kobsa, 2001; Rich, 1979; Webb et al., 2001), there are usually 4 types of user models, which are static user models, dynamic user models, stereotype-based user models, and highly adaptive user models. Static user models are the most basic type of user models. Usually, personal information, such as names, ages, skills, behaviours, is collected before any interaction with the systems to build static user models, which are no longer changed during the interaction with the systems. However, such models lack the adaptation ability to the personal changes

of users. If the user's preference changes later, the static user models need to be rebuilt. Dynamic user models allow changes of personal preferences during the interaction with the systems, thus they can be updated based on new goals, latest individual information, *etc.*. Stereotype-based user models are commonly used in user modelling research. Building such models is mainly based on demographic statistics. Users are classified into different stereotypes after collecting their relevant information. For a new user with little-known information, the computer/application can still infer the relative characteristics of this user after classifying him/her to a stereotyped group. However, sometimes a user's personal attributes may not match any of the existing stereotypes and such situations cannot be dealt with by the user models flexibly. Highly adaptive user models aim to build a representation for a particular user, therefore it allows a high flexibility and adaptivity of the system. Since a unique model for each user is built, it can avoid the disadvantages of stereotype-based user models.

In order to build user models, users' information should be collected. Usually there are 3 methods (Benyon and Murray, 1993; Montaner et al., 2003; Salvendy, 2012; Vu and Proctor, 2011; Yin et al., 2015). The 1st method is to directly ask the users who interact with the systems, such as questionnaires or the registrations processes for the social websites. For instance, when somebody registers for a social platform, personal information such as ages, occupations, and locations is provided by the user. Initially, the user's information might not be complete and the user can add new information or modify previous information later. Such method is intuitive, direct, and easy to implement. However, depending on the purpose of the application, some kinds of personal information can not be directly given by the user. Thus the 2nd method is to learn users' preferences by observing and interpreting their interactions with the system. For instance, based on our previous search history, the search engine can categorise this information into differ-

ent types, find the most frequent search type, and provide relevant information that we might be interested potentially. As we mentioned earlier, data mining and machine learning methods are usually applied when building such user models. Since the user's preference is mainly interpreted by the system, some users might feel that the user models lack the interpretations of their behaviours, skills, *etc.* by themselves. However, we should be aware that this is not absolute because users cannot always interpret their preferences by themselves depending on the purposes of applications. The 3rd method is a hybrid approach which asks for explicit feedback and alters the user model by adaptive learning. This method combines the first two methods and takes the advantages of both. It not only allows objective opinions from users but also updates user models dynamically.

There has been large amount of research work on user modelling and personalisation in different fields, such as personalised network updates for increasing social interactions and contributions in social networks (Berkovsky et al., 2012), personalised theme and tour recommendations for museums visitors (Bohnert et al., 2012), inferring personality of online gamers by fusing multiple-view predictions (Shen et al., 2012), a multi-faceted user model for twitter (Hannon et al., 2012), property-based interest propagation in ontology-based user model (Cena et al., 2012), preference relation based matrix factorization for recommender systems (Desarkar et al., 2012), modelling multiple distributions of student performances to improve predictive accuracy (Gong et al., 2012), studies to determine user requirements regarding in-home monitoring systems (Larizza et al., 2012), enhanced semantic TV-show representation for personalised electronic program guides (Musto et al., 2012), and automating the modelling of learners' erroneous behaviours in model-tracing tutors (Paquette et al., 2012).

2.2.2 User Modelling in Robotics

Personalisation and user modelling play significant roles not only in our daily living, but also in human-robot interactions. When we discuss the general user modelling methods in the previous section, the systems that interact with users mainly refer to computers, electronic devices, *etc.*. While in this section, we discuss interactions between users and robots. Fong *et al.* point out that user modelling is useful for socially interactive robots to not only understand human behaviours but also adapt their behaviours to different users (Fong, Nourbakhsh and Dautenhahn, 2003).

There have been various kinds of methods to build user models in human-robot interaction. For instance, user stereotypes can be pre-defined with different attributes (Flandorfer, 2012; Fong, Thorpe and Baur, 2003). During human-robot interactions, a robot can match the current user's attributes with existing user stereotypes and interact with the user based on the stereotype attributes. The user model in the stereotyped approach could be for a single user or a group of users.

With a hands-off assistive robot during post-stroke rehabilitation therapy, Tapus *et al.* investigated the extroversion-introversion personality matching between the robot and users (Tapus *et al.*, 2008). Since there is a strong relationship between the human personality and behaviour (Ewen, 2014), they believed that robots should act in accordance with the user's personality during human-robot interactions. Experimental results showed that human users preferred the robot behaviours which matched with their own personalities.

There has been research work on studying user preferences in an object handover scenario. For a companion robot approaching a seated person in a helping context, Dautenhahn *et al.* studied user preferences for the comfortable approach directions by considering factors such as gender differences,

age, and handedness (Dautenhahn *et al.*, 2006). Apart from robot approaching, Cakmak *et al.* presented a user study on human preferences of robot hand-over configurations, using a simulated kinematics model of human to collect information on user preferences (Cakmak *et al.*, 2011). Besides, the spatial reasoning of users, such as user visibility and arm comfort, was considered as an important factor in (Sisbot *et al.*, 2007) for object hand-over tasks.

For an assistive robot to provide personalised assistance in (Klee *et al.*, 2015), the task for the robot was initially viewed as a template which consisted of a sequence of robot motions. The first form of personalisation was that the robot's goal positions were instantiated with the user's physical features, such as the human height and pose. The second form of personalisation was that the robot learned the user's physical constraints during interactions. The robot finally finished the task by preventing the user reaching the constrained area.

To recognise local navigation plans of the wheelchair users, a probabilistic approach was proposed in (Hüntemann *et al.*, 2013) which fused past driving information with a personalised user model. The user model learned the transformation relationships from the user's mental plans to the robot's input using Gaussian Process Regression. The navigation plan process was modelled with a Dynamic Bayesian Network.

For human-robot collaboration tasks, Nikolaidis *et al.* presented a framework to learn different user models (Nikolaidis *et al.*, 2015). From demonstrated human-robot joint actions, different types of user motion were modelled using an unsupervised learning algorithm and the joint actions were modelled with a mixed-observability Markov decision process (MOMDP). Given a new user, the proposed framework could infer the user's type and compute a user-preferred policy for the robot to cooperate with the user.

Mason and Lopes developed a robotic system which could gradually adapt its behaviours to different users through repeated interactions (Mason and Lopes, 2011). During interactions, users sent verbal commands to the robot and gave positive or negative feedback to the robot's goal states. The user's preferences were modelled as a function which could judge if a given robot state was satisfied or not. For a new task, the robot first generalised from the learned user's profile and then planned its motions without new user's requests. The proposed system was applied to a robot to tidy up a table with randomly located objects.

Lee *et al.* designed a snack delivery service with a humanoid robot and explored the advantages of applying a personalisation strategy on the robot (Lee, Forlizzi, Kiesler, Rybski, Antanitis and Savetsila, 2012). The snack service lasted for 2 months for each participant. For each user, the history of each time interaction was recorded, the user's snack choice patterns and the service usage patterns were analysed to personalise the robot's social behaviours. Experimental results showed that adding the personalisation feature to the snack robot could improve the users' engagement and relationships with the robot.

To evaluate if personalisation could yield benefits in assistive human-robot interactions, Leyzberg *et al.* designed experiments to use a robot tutor to assist participants to solve grid-based logic puzzles (Leyzberg *et al.*, 2014). During interactions, the robot tutor had 4 different choices to assist users by providing different lessons, which are no lessons, randomised-but-relevant lessons, personalised lessons chosen by an additive skill assessment algorithm, and personalised lessons chosen by a Bayesian network skill assessment algorithm. Experimental results showed that participants who received personalised lessons performed much better comparing with participants who received non-personalised lessons.

It can be seen that user modelling is closely related to the aims of applications, which decides the types of users' information to be collected and the ways to build user models. Besides, how to enable robots to make use of the user models also needs to be concerned. In this thesis, our interest of the application domain is assistive dressing by humanoid robots. However, there is no prior user modelling work on assistive dressing. As we discussed in section 2.1.2, although there has been some research work on dressing robots, the focus is either on a sub-part of a dressing task or the learning of robot dressing motor skills, and the experiments were mainly conducted with human mannequins. One of the main reasons that existing research work cannot enable robots to assist real human users to dress is that states of real human users are usually not considered. Through building user models, we collect information about users' states and model individual user's preference in order to enable humanoid robots to provide not only dressing assistance, but also personalised dressing assistance.

2.3 VISION IN ROBOTICS

In this section, we review the research work in human-centred vision. We first mainly look at the computer vision methods in human pose recognition, human hand pose recognition, and recent advances of utilising deep learning in computer vision in section 2.3.1. Then we review some vision methods or techniques applied in human-robot interaction in section 2.3.2.

2.3.1 *Human-centred Vision*

Vision is one of the most critical components in robotic applications. With vision sensors, robots could recognise the environment, behaviours of other robots or human partners to better interact with the world. There have been

significant advances in the research field of human pose recognition and hand pose recognition.

Researchers have been studying on human pose recognition for a long period (Aggarwal and Ryoo, 2011; Moeslund et al., 2006; Poppe, 2007, 2010; Turaga et al., 2008; Wu and Huang, 1999). With challenging video dataset, Ferrari *et al.* proposed an approach which could progressively reduce the search space for the body parts (Ferrari et al., 2008). The body parts were firstly inferred in a single frame and multiple frames were used to further refine the single-frame estimation results. Detected human pose was represented with stick models. Rogez *et al.* presented an exemplar-based approach for human pose detection and recognition using randomised trees (Rogez et al., 2008). With a bottom-up approach, different human motions for training were recursively clustered and merged at each level when building a decision tree. Given a new image, the human's pose can be classified with the learned pose classifier forest. To recognise the 3D human pose, Shotton *et al.* proposed to use randomised decision forests to classify every pixel of a single depth image into a body label (Shotton et al., 2013). A local mode-finding approach based on mean shift (Comaniciu and Meer, 2002) was then used to search for the 3D joint positions. Girshick *et al.* proposed to use Hough forests to recognise 3D human pose from the depth images (Girshick et al., 2011). Each pixel of the depth image voted for the positions of different joints and the final voted joint positions were found using mean shift. With the method of Hough forests, even if there existed self-occlusions, all the body joints could still be inferred.

Apart from human pose recognition, researchers have also been working on hand pose recognition (de La Gorce et al., 2011; Erol et al., 2007; Khamis et al., 2015; Tang et al., 2015; Wang and Popović, 2009; Yub Jung et al., 2015). Comparing with the human body, there are more articulations and self-occlusions in hand pose recognition. Tang *et al.* proposed a semi-

supervised transductive algorithm for 3D hand pose estimation in real-time (Tang et al., 2013). Labelled realistic dataset and a large synthetic dataset were used to train the Semi-supervised Transductive Regression (STR) forest. In testing phases, joint patches were first densely extracted from the depth image and each patch voted for the potential locations of all joints. The final voted joint locations were acquired using the kinematic joint refinement method. In contrast to this method where joint patches needed to be extracted from a depth image, Tang *et al.* proposed another Latent Regression Forest (LRF) method, where the whole point cloud starting from the centre of mass propagated down the latent regression tree until all the skeletal joints were located (Tang et al., 2014). The method could be viewed as a structured coarse-to-fine search process. Oikonomidis *et al.* proposed a model-based method for 3D tracking of hand articulations (Oikonomidis et al., 2011). The problem was viewed as an optimisation process by seeking the hand model parameters which minimised the errors between the hypothesised model and actual observations. Particle Swarm Optimisation (PSO) (Kennedy, 2011) was applied to solve the optimisation. By combining motion and skeleton information, Chang and Demiris presented an unsupervised method to learn the kinematic structure for complex articulated objects such as human hands (Jin Chang and Demiris, 2015). Each rigid motion segment was extracted with randomised voting (Jung et al., 2014). To estimate the skeleton information, the object silhouette was first generated with the support vector data description method (Tax and Duin, 2004) and then the skeletons were found by utilising a distance function.

In recent years, deep learning has become more and more popular in research fields such as machine learning, computer vision, and speech recognition (Bengio et al., 2013; LeCun et al., 2015; Ngiam et al., 2011; Schmidhuber, 2015). There are different forms of deep learning architectures, for instance, deep neural networks, deep belief networks, recurrent neural networks, and

convolution deep neural networks (Arel et al., 2010; Bengio et al., 2015). One of the significant differences between deep learning and other machine learning methods is that most machine learning methods require identifying features in the raw input data, such as the most commonly used scale-invariant feature transform (SIFT) (Lowe, 1999; Nixon, 2008). This feature extraction process is not needed in deep learning. The raw data can be directly used as the input and the training process can automatically discover the most useful patterns. Deep learning originated from artificial neural networks (Cheng and Titterton, 1994; Egmont-Petersen et al., 2002), which could only be trained with a few layers about 40 years ago, due to the limit of computation power. Nowadays, with the rapid development of computer hardware, software, GPUs, *etc.*, more and more complex and deep neural networks can be trained to solve big data problems. Deep learning has been successfully applied to solve various computer vision problems, such as human pose estimation (Jain et al., 2014; Ouyang et al., 2014; Thompson et al., 2014; Toshev and Szegedy, 2014), object recognition (Donahue et al., 2014; Krizhevsky et al., 2012; Nair and Hinton, 2009; Socher et al., 2012), face recognition (Hu et al., 2014; Huang et al., 2012; Sun et al., 2013, 2014; Taigman et al., 2014), and image classification (Akata et al., 2014; Ciregan et al., 2012; Karpathy et al., 2014; Simonyan and Zisserman, 2014).

2.3.2 *Vision in Human-robot Interaction*

Vision plays a significant role in human-robot interactions. According to the purposes of different applications, either simple or complicated vision methods can be applied.

One of the simplest ways to detect the positions of objects or humans is by using markers. Wu and Demiris proposed a template-based approach for robots to learn arm trajectories through imitation learning (Wu and De-

miris, 2010). A few markers were attached to the human teacher's arm so that the demonstrated arm trajectories could be recorded. For a humanoid robot iCub to learn the dancing actions from human demonstrations (Lee et al., 2013), the human teacher wore clothes covered with markers and the human joint motions were recognised with a motion capture system. For the assistive dressing applications in (Tamei et al., 2011), markers were attached on both the human mannequin and the T-shirt. The positions of these markers were detected by a motion capture system so that the topological relationships between the mannequin and the clothes could be analysed. To analyse the body movement of human-robot interaction in (Kanda et al., 2003), markers were attached to both the robot and human users. To determine the smart assistive wheelchair's location in the environment, Carlson and Demiris placed fiducial markers at regular intervals on the ceiling (Carlson and Demiris, 2008). A camera was positioned on the wheelchair which faced towards the ceiling to detect the markers. Structured visual markers were also used for indoor pathfinding in (Kalkusch et al., 2002).

The human's head orientation could indicate the human's attention during human-robot interaction. With this information, robots could understand better the human's interests. By wearing a helmet or a pair of glasses attached with markers, the human head pose was detected using the motion capture system in (Lemaignan et al., 2014; Pandey and Alami, 2010; Pandey et al., 2013). To estimate human head pose without markers, depth information was used in (Seemann et al., 2004) which were acquired from a stereo camera to track the human face with the colour-based face detection technique (Nickel and Stiefelhagen, 2003). Then the human head pose was estimated with a three-layer feedforward neural network. Stiefelhagen *et al.* also used two neural networks to estimate respectively the rotation angles for the pan and tilt of the head pose (Stiefelhagen et al., 2004). Fischer and Demiris proposed a markerless perspective taking method for humanoid

robots to see the world from the human's viewpoint (Fischer and Demiris, 2016). With the normalised depth data, the random forest algorithm (Fanelli et al., 2011) was applied to estimate the human head pose.

Human body pose estimation plays a significant role in human-robot interactions. With the mature development of depth sensors such as Kinect (Zhang, 2012) and Xtion (Gonzalez-Jorge et al., 2013), the OpenNI skeleton tracking software becomes a popular tool to recognise the human body pose in human-robot interactions. For instance, a humanoid robot learned human motions through imitations where the human pose was recognised with the skeleton tracker (Lee et al., 2012), complicated human gestures could be recognised based on the lower-level skeleton tracking during human-robot interactions (Fanello et al., 2013), and a dancing robot tutor could evaluate children's dancing performance with a skeleton tracker (Ros and Demiris, 2013). Pose recognition methods could also be designed heuristically based on the experimental set-ups. For instance, Fasola and Mataric presented a coach robot to engage elderly people in the physical exercises (Fasola and Mataric, 2013). With a black background, a user was first segmented from the whole image and the user's face was detected with the OpenCV frontal face detector. The locations of the human hands were determined by examining the extrema points of different arms. Luo *et al.* presented an adaptive massage trajectory generator for a humanoid robot to provide a massage service to humans (Luo et al., 2014). Human back pose was recognised with an adaptable method proposed by (Buys et al., 2014) which made use of the RGB-D sensor data, a human kinematic model, and the random decision forest classification method.

2.4 ROBOT LEARNING USING MULTI-MODAL INFORMATION

In this section, we review the research work in robot learning using multi-modal information. We first look at different kinds of sensors that can provide human user information in section 2.4.1, then we move to human-robot interactive research using multi-modal information in section 2.4.2.

2.4.1 *General Sensors for Human*

For robots to interact more intelligently with humans, more knowledge about the humans should be acquired. There are various kinds of sensors that can provide human information from different perspectives.

In our everyday life, we usually use verbal language to communicate with other people, express our emotions or needs. It would be useful for robots if they could hear and understand humans. There has been research work on human speech recognition using machine learning methods, such as neural networks (Chan et al., 2016; Dahl et al., 2012; Graves et al., 2013; Hinton et al., 2012; Sak et al., 2015; Seltzer et al., 2013). Human language is also used in robot imitation learning, where the human teacher can provide positive or negative verbal feedback, or corrections, to the robots (Cakmak et al., 2010; Cakmak and Thomaz, 2012; Cantrell et al., 2011).

There are sensors which can detect the electroencephalography (EEG) signal from the human brains. Choi and Jo proposed an EEG system-based hybrid brain-computer interface for the human users to control humanoid robot navigation (Choi and Jo, 2013). Human brain signals were interpreted into commands, such as turning left or right, and these commands were sent to the robot. Cinar and Sahin applied Particle Swarm Optimisation and Radial Basis Function Networks (PSO-RBFN) for the EEG signal classification and controlled the robot movement with the brain signal in real time (Cinar

and Sahin, 2013). There are also studies in shared control for wheelchair users using EEG signal (Carlson and Millan, 2013; Leeb et al., 2015). Apart from the EEG signal, there are also sensors which can detect the electrocardiography (ECG) signal from a human body (Maglaveras et al., 1998; Sameni and Clifford, 2010).

Apart from the sensors that can reveal the human's physiological information, sensors can be attached to the robot's body to detect the interaction information between the robot and humans. For instance, Mukai *et al.* developed the tactile sensor system for a humanoid robot RI-MAN (Mukai et al., 2008). When the robot held a human mannequin, the magnitude and position of the mannequin could be detected by the tactile sensor system. Ohmura and Kuniyoshi developed a humanoid robot attached with the whole-body tactile sensors and enabled the robot to lift a 30kg box using the tactile feedback (Ohmura and Kuniyoshi, 2007). Such abilities like lifting and landing humans are crucial for rescue robots in the future. Soh and Demiris enabled a humanoid robot iCub to classify non-rigid objects with tactile sensors using sparse online infinite Echo-State Gaussian process (OIESGP) (Soh and Demiris, 2014). A tactile array sensor was used to recognise objects through multiple touches in (Luo et al., 2015).

Force sensors are also widely used in human-robot collaborations. Mohareri *et al.* proposed a control framework for two pairs of master and slave da Vinci surgical robots in a teleoperated robot-assisted surgery (RAS) environment (Mohareri et al., 2013). While the human action hand performed the surgical task, the other human fixed hand would feel the force feedback from the environment and adjust its motions. A pair of haptic devices was used in (Kucukyilmaz and Demiris, 2015) to assist wheelchair users to drive. One of the haptic devices was controlled by a human teacher while the other was controlled by the users. When facing difficulties in turning the wheelchair, the human teacher assisted the user in driving with the haptic device. The

force information was sensed by the user's haptic device and then turned into the control command to the wheelchair.

2.4.2 Human-robot Interaction using Multi-modal Information

For robots to interact more naturally with humans, robots should be able to utilise multi-sensor or multi-modality information. Using multi-modal information could compensate the disadvantages of using unimodal information when noise or ambiguity occurs in the unstructured environment.

Stiefelbogen *et al.* studied multi-modal human-robot interaction by enabling a humanoid robot to recognise speech, track humans, identify human faces, detect pointing gesture, and estimate human head pose (Stiefelbogen *et al.*, 2007). For multi-modality information fusion, a constraint-based rule system was used to determine which events could be merged, based on constraints such as time correlation and semantic content. Petit *et al.* proposed a dynamic autobiographical memory system for humanoid robots to receive continuous multi-modal data from the robot sensors and external devices (Petit *et al.*, 2016). Multi-modal data includes RGB data, proprioceptive data, and tactile data from the robots, and depth data and human skeleton information from the depth sensor. With the long-term memory, the robots could not only remember, but also relive the previous interactive event with humans.

Lang *et al.* presented a multi-modal attention system for a mobile robot to track humans (Lang *et al.*, 2003). The system consisted of three modules, which were the vision module to recognise the human faces, stereo microphones to locate the human sound sources, and a laser range finder to detect the human legs. Martin *et al.* also enabled robots to detect and track humans with the proposed multi-modal fusion method using a probabilistic aggregation scheme (Martin *et al.*, 2006). Data from an omnidirectional camera,

laser, and sonar sensors was concurrently processed to infer nearby human positions.

Force and vision information could be used together in human-robot collaborative tasks. Rozo *et al.* proposed an approach incorporating dynamical systems, probabilistic learning, and stiffness estimation. They enabled a robotic arm to learn the physical collaborative behaviours from human demonstrations by satisfying both the position and force constraints (Rozo *et al.*, 2016). The robot was able to learn the behaviours such as lifting, moving, and landing an object with a human partner, and the assembling task by holding a wooden table while a human partner screwed the four legs to the table. Kruse *et al.* presented a feedback controller using both force and vision information, and enabled a Baxter robot to collaboratively unfold a piece of cloth with humans by responding to force and vision changes (Kruse *et al.*, 2015).

There are human-robot interactive tasks which intrinsically require multi-modal information. Zambelli and Demiris proposed an imitation learning method to enable a humanoid robot iCub to learn how to play with a piano keyboard from a human teacher (Zambelli and Demiris, 2016). Through self-exploration, the robot learned sensorimotor representations on multi-modal task spaces including vision, touch, and proprioception. Given a new task, the robot inferred its own motion by fulfilling multi-modal constraints. Schmidts *et al.* combined Gaussian Mixture Regression and Hidden Markov Models to teach a robot hand to imitate human grasping skills from motion and force data (Schmidts *et al.*, 2011). They showed that, regarding the generalisation capability for grasping other similar objects, learning from motion and force data outperforms learning from motion data only.

2.5 ROBOT PATH PLANNING

There is a large number of research problems that requires humanoid robots to plan paths for their arm motion. Most of the time, path planning becomes an optimisation problem. In this section, we mainly review research work in robot motion path planning and stochastic optimisation.

Imitation learning termed as learning by demonstration aims to teach robots different actions or behaviours through human demonstrations (Argall *et al.*, 2009; Billard *et al.*, 2008; Schaal, 1999). For a robot to imitate a given task, the robot usually needs to plan its end-effector path to accomplish the task. Wu and Demiris proposed a template-based approach for the robot to imitate the human arm movement (Wu and Demiris, 2010). A human demonstration was recorded as a template which consisted of a set of spatial feature points. For the robot to reach a new goal position, the end-effector path was planned by warping the feature points and minimising the translation energy. Calinon *et al.* proposed a task-parametrised mixture model to enable the robots to learn a set of demonstrated movements which were observed from different candidate frames (Calinon *et al.*, 2014). Given a new task, a temporary Gaussian Mixture Model was first estimated by the learned model and the tracking reference was then retrieved by Gaussian Mixture Regression.

There are a number of robotic research in stochastic path optimisation, such as robot motion planning (Latombe, 2012; Masehian and Sedighizadeh, 2007) and path planning (Raja and Pugazhenth, 2012). Hegels *et al.* proposed an approach to post-optimisation using the non-linear conjugate gradient method. They applied the iterative path optimisation algorithm in the real-world application of robot-guided thermal spraying (Hegels *et al.*, 2015). The optimal path was planned offline with a simulation tool and real-time feedback was not taken into consideration.

Stochastic optimisation recently became an active area of research because of the popularity of deep networks. To perform a parameter update with the gradients, there has been some well-established methods. Vanilla update which uses stochastic gradient descent (SGD) (Rumelhart, Hinton and Williams, 1986) is the simplest form, but it may not converge or it converges slowly when the learning rate is small enough. With momentum update (Rumelhart, Hinton and Williams, 1986), the parameter vector builds up velocity in any direction which has the consistent gradient, thus the convergence rate is faster. Duchi *et al.* proposed an adaptive learning rate method AdaGrad, which performed well with sparse gradients (Duchi *et al.*, 2011). Tieleman & Hinton presented RMSProp which adjusted the AdaGrad method by using a moving average of squared gradients (Tieleman and Hinton, 2012). The recently proposed Adam method (Kingma and Ba, 2015) combines the advantages of the two popular methods AdaGrad and RMSProp, which has shown its robustness to a variety of non-convex optimisation problems in machine learning.

2.6 CONCLUSION

In this chapter, research work in socially assistive robots, user modelling, vision in robotics, robot learning using multi-modal information, and robot path planning have been reviewed. Although assistive robots can perform various kinds of tasks in home environments, assistive dressing remains a challenging problem. Existing methods focus on enabling robots to learn the motor skills for dressing and experiments are usually conducted with a human mannequin.

We have shown that user modelling and personalisation play significant roles in our lives. Although there has been research on modelling user behaviours or preferences in human-robot interaction, interacting scenarios

mainly include robots handing over objects to humans or robots approaching humans from different directions.

Various sensor information has been used to collect the human user data and there has been research in human-robot interaction using multi-modal information. However, existing methods mainly enable robots to interact with humans using multiple sensors and less attention has been paid to model the human preferences. It is obvious that multiple sensors could provide more robust information compared to a single sensor, but how to make use of multiple sensor information in user modelling brings new challenges. For instance, one may think to build each layer of the user model using different sensor information and then combine all the layers of user models hierarchically. One may think to build one layer of the user model using one type of sensor information and use other sensor information to update the existing layer of the user model. The other may think to build a combined user model using multi-modal information directly. Another challenge is that user models should also be able to adapt to the changes from the users. For instance, the movement flexibility of the human body would gradually decrease as humans grow older. For long-term human-robot interactions, the ability for adaptation should be taken into consideration when building the user models.

BACKGROUND

USER MODELLING USING VISION INFORMATION

There are two key criteria that we use to evaluate the performance of our proposed methods throughout this thesis, which are the reachability criterion and the comfort criterion. In this thesis, (1) the definition of the reachability criterion is that assistive robots provide dressing assistance by placing the clothes within reach of the human upper-body joints, (2) the definition of the comfort criterion is that there is no external force resistance detected during assistive dressing.

In this chapter, we present a user modelling method using vision information and enable a Baxter robot to provide personalised dressing assistance to real human users by fulfilling the reachability criterion. With vision information, one of the most distinguishable features among different people is the reachable area of the upper body. Not only for assistive dressing, but also for object handover, assistive robots should avoid placing clothes or objects to the areas which are difficult to reach for the users. Thus learning the reachable area of the human upper body is beneficial to assistive robots to provide personalised assistance. To recognise each body part, the robots should also be able to estimate human upper-body pose in real time.

3.1 SYSTEM OVERVIEW

The overview of the proposed system in this chapter is shown in Figure 3.1, along with the system illustration shown in Figure 3.2.

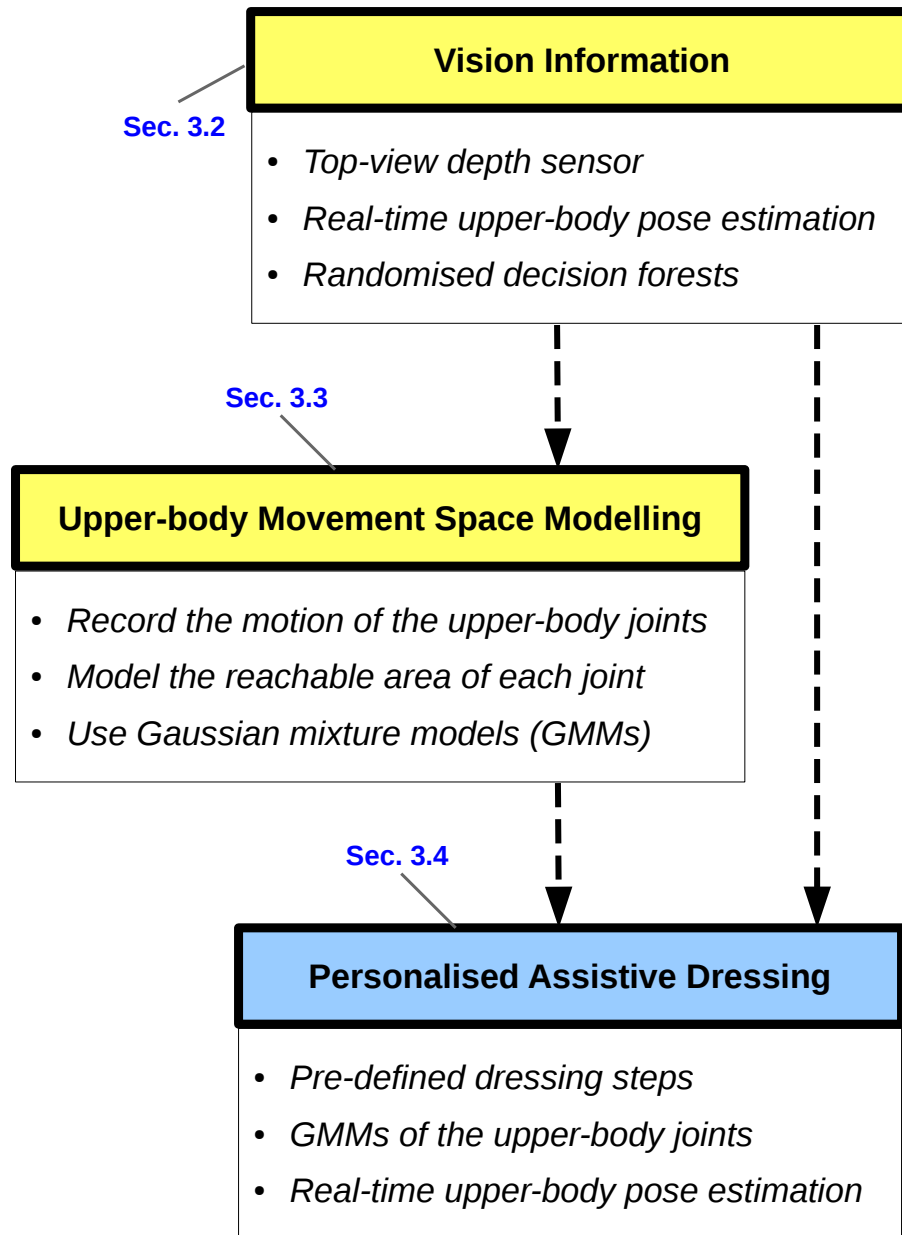


Figure 3.1: System overview in Chapter 3. Each box indicates a main section along with a few key words. A top-view depth sensor is used to provide depth images of a user. We adopt the randomised decision forest method (Shotton et al., 2013) to recognise the human upper-body pose in real time from a single depth image. Then we record the human motion, which consists of a sequence of poses of the upper-body joints. With this information, we propose to model the movement space of the human upper-body joints using GMMs to learn the reachable area of each joint. When the probability of a position given the GMMs is higher, it means that this position is more frequently visited by the corresponding upper-body joint. For assistive robots to provide personalised dressing assistance, a set of goals is sent to the robot’s grippers to execute in each step, where the exact values for the goal positions are calculated according to the GMMs of the human upper-body joints and real-time upper-body pose estimation.

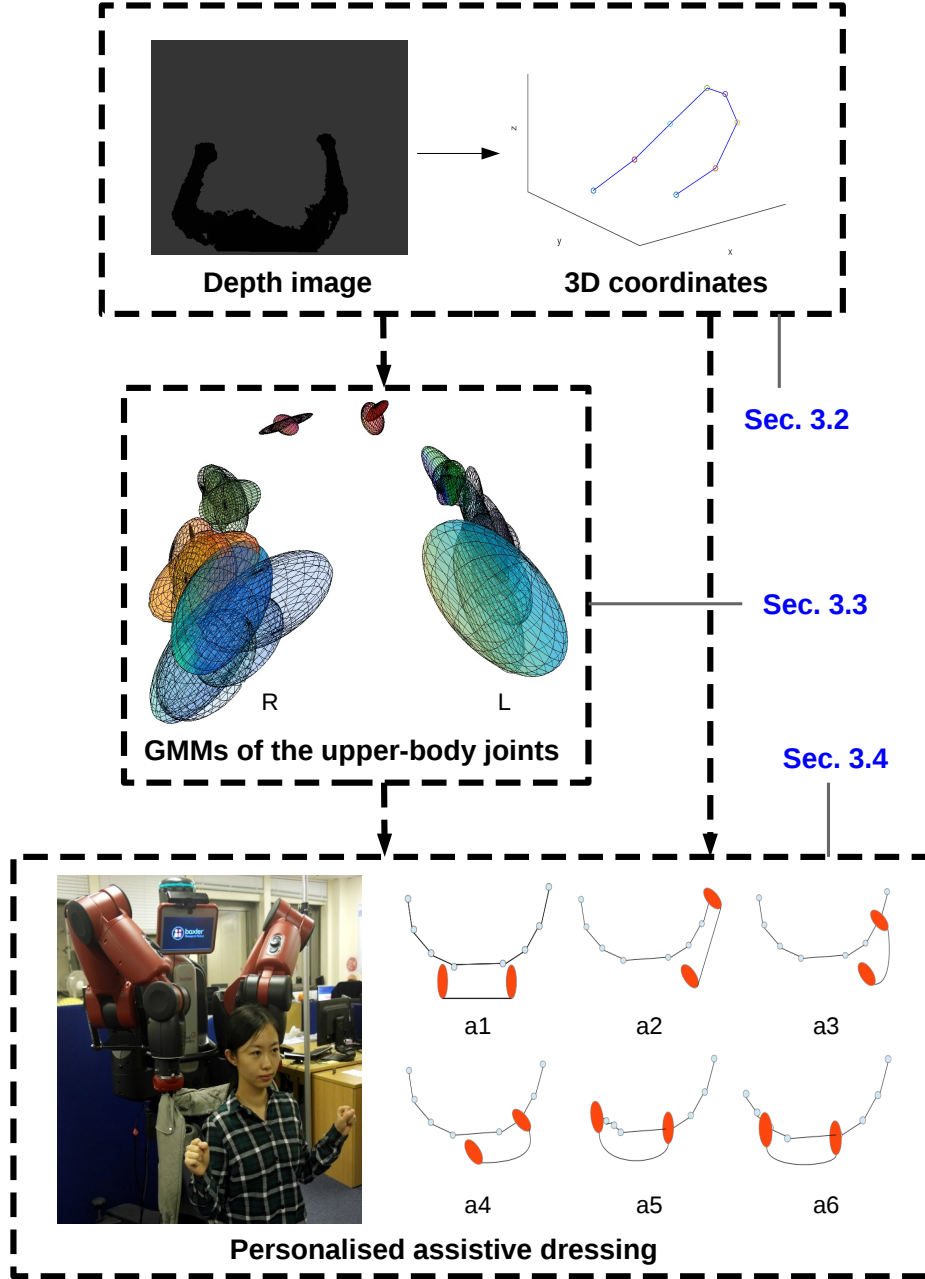


Figure 3.2: System illustration in Chapter 3. Each box indicates a main section. The structure of the system is the same as in Figure 3.1. The box of section 3.2 shows that the 3D coordinates of the upper-body joints are extracted from a single depth image. The box of section 3.3 shows the 3D visualisation of the GMMs of the human upper-body joints, where L and R indicate the left and right arm respectively. The box of section 3.4 shows that a Baxter humanoid robot assists a user to dress with a sleeveless jacket. (a1)-(a6) show the dressing steps for the robot's grippers from a top-view. The 2 orange circles represent the positions of the robot grippers, which are also the positions of the jacket shoulders. The robot first dresses the user's right arm and then the left arm, following the order of hand, forearm, upper arm, and shoulder. The exact values of the goal positions in each step are personalised according to the GMMs of the human upper-body joints and real-time upper-body pose estimation.

In Figure 3.1, each box indicates a main section along with a few keywords. A top-view depth sensor is used to provide depth images of a user. We adopt the randomised decision forests method (Shotton et al., 2013) to recognise the human upper-body pose in real time from a single depth image. Then we record the human motion, which consists of a sequence of poses of the upper-body joints. With this information, we propose to model the movement space of the human upper-body joints using GMMs to learn the reachable area of each joint. When the probability of a position given the GMMs is higher, it means that this position is more frequently visited by the corresponding upper-body joint. For assistive robots to provide personalised dressing assistance, a set of goals is sent to the robot's grippers to execute in each step, where the exact values for the goal positions are calculated according to the GMMs of the human upper-body joints and real-time upper-body pose estimation.

The structure of the system in Figure 3.2 is the same as in Figure 3.1, where each box indicates the main section. The box of section 3.2 shows that the 3D coordinates of the upper-body joints are extracted from a single depth image. The box of section 3.3 shows the 3D visualisation of the GMMs of the human upper-body joints, where L and R indicate the left and right arm respectively. The box of section 3.4 shows that a Baxter humanoid robot assists a user to dress with a sleeveless jacket. (a1)-(a6) show the dressing steps for the robot grippers from a top-view. The 2 orange circles represent the positions of the robot's grippers, which are also the positions of the jacket shoulders. The robot first dresses the user's right arm and then the left arm, following the order of hand, forearm, upper arm, and shoulder. The exact values of the goal positions in each step are personalised according to the GMMs of the human upper-body joints and real-time upper-body pose estimation.

3.2 UPPER-BODY POSE RECOGNITION

In this section, we mainly describe how we use a top-view depth sensor to recognise human upper-body pose by adopting the randomised decision forests method proposed in (Shotton et al., 2013). Besides, we also describe how we generate labels for training images using colour information in section 3.2.2.

3.2.1 *Randomised Decision Forests*

Considering the dressing need, only the human upper-body pose is of interest. In this chapter, we define 8 body parts for an upper body which are left/right (L/R) shoulder, L/R upper arm, L/R forearm, and L/R hand. We omit the human head since we are more concerned with the arm movements.

A randomised decision forest (Shotton et al., 2013) is a multitude of decision trees which consist of split and leaf nodes. A split node can be seen as a weak classifier which contains the information of a selected feature and a threshold while a leaf node contains the information of the probabilities that this leaf node belongs to a certain body part. Considering the amount of our training data, we train a randomised decision tree instead of a forest.

For a given pixel n in a depth image, we choose the same depth comparison features as in (Shotton et al., 2013):

$$f_{\theta}(I, n) = d_I(n + \frac{u}{d_I(n)}) - d_I(n + \frac{v}{d_I(n)}) \quad (3.1)$$

where $d_I(n)$ is the depth value of pixel n , I indicates the specific image that pixel n comes from, and $\theta = (u, v)$, where u and v are the offsets. The offsets u and v are normalised by the depth of pixel n to ensure that the features are depth invariant. This feature calculates the depth difference between

two neighbour pixels of n . If a neighbour pixel lies outside the bounds of the image or lies on the background, the depth of this neighbour pixel is set to a large positive constant value.

We follow the same training steps as (Shotton et al., 2013) for the randomised decision tree. The training steps are:

1. Generate a set of random splitting candidates $\phi = (\theta, \tau)$, where θ is the feature parameter and τ is the threshold.
2. Partition the training dataset reaching the current node $Q = \{(I, n)\}$ into left and right subsets by ϕ :

$$\begin{aligned} Q_l(\phi) &= \{(I, n) | f_\theta(I, n) < \tau\} \\ Q_r(\phi) &= Q \setminus Q_l(\phi) \end{aligned} \quad (3.2)$$

3. Compute the ϕ^* :

$$\begin{aligned} \phi^* &= \arg \max_{\phi} G(\phi) \\ G(\phi) &= H(Q) - \sum_{s \in \{l, r\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi)) \end{aligned} \quad (3.3)$$

where $H(Q)$ is the Shannon entropy.

4. Go to step 2 until the terminating conditions (e.g. reaching the maximum depth of the tree) are satisfied.

Through training each tree model, a pair of offsets u and v , and a threshold τ are learned for every split node. A distribution over body part labels is also stored in every leaf node. In the testing phase, each pixel from a filtered new depth image is classified by the learned tree model. Specifically, every pixel traverses the tree starting at the root and finally reaches a leaf node after repeatedly evaluating equation (3.1) and branching left or right by the weak classifiers.

After classifying the foreground pixels of a depth image in the testing phase, we extract the mean point of each body part to represent the position of its corresponding joint. We calculate the mean values of the 2D coordinates for each upper-body joint after filtering outliers, and then convert them to the 3D coordinates using the depth information in the camera coordinates. For the assistive robot to work with the user's upper-body pose, we convert every joint position from the camera coordinates to the robot coordinates according to the spatial relationships of the two coordinates.

3.2.2 Labelling Training Images

First of all, labelled training images for a user should be collected. A sitting user is allowed to move both arms freely without self-occlusions within the working range of the camera while pair-wise pixel-aligned RGB and depth images are recorded. The user wears clothing with 8 different colours on the upper body during the training data collection, which facilitates the following segmentation process using colour information.

After collecting RGB and depth images for training, we extract foreground pixels by filtering the head and the background. One reason for filtering the human's head is because the head movement is not of interest. Another reason is that with our experimental set-up, by mounting the depth sensor on top of Baxter's facescreen, a user's head is usually within the minimum distance, which ends up with no information for the head from the depth sensor. For each user, we measured the distance from the depth sensor to the user's neck before collecting data. Then we used this distance information as a threshold to filter the head. To generate labels for every pixel remaining in the RGB images, we first calculate the sample colours in the $L^*a^*b^*$ colour space for each piece of cloth, where L^* is the luminosity layer, a^* and b^* are the chromaticity layers. Then, we classify each pixel using the nearest

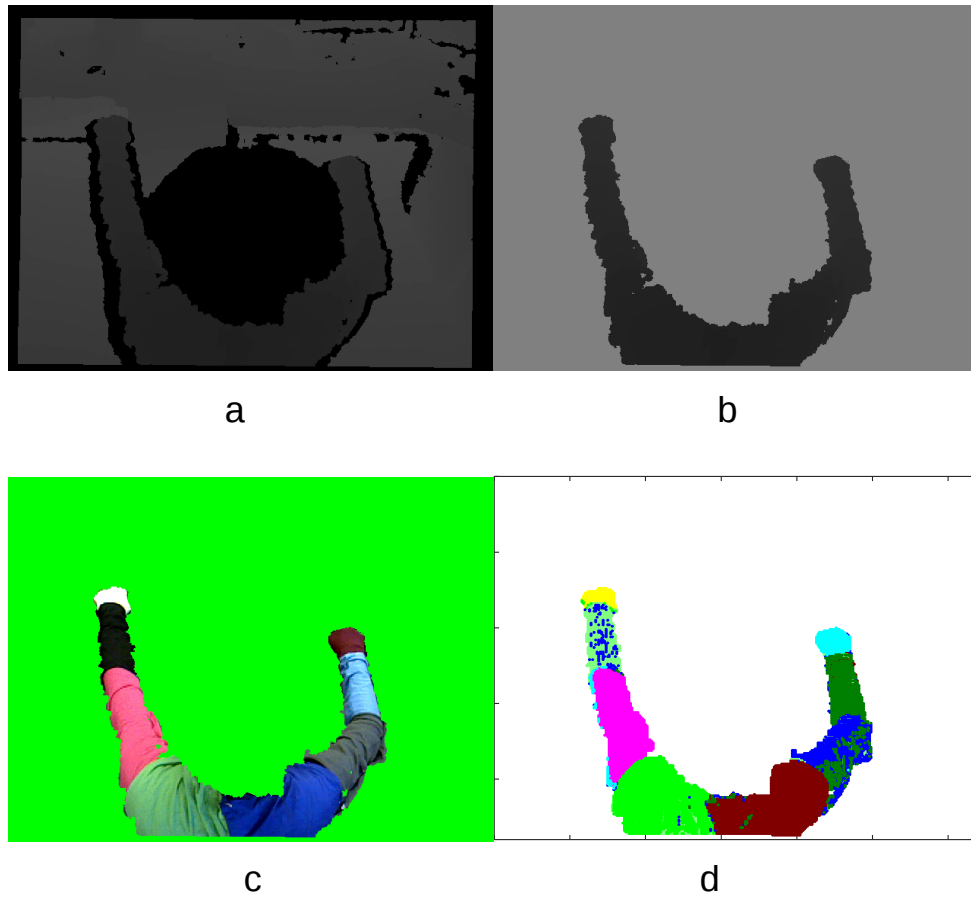


Figure 3.3: Colour-based segmentation for labelling training data. (a) shows an original depth image. (b) and (c) show the depth and RGB image after filtering the head and background. The user wears clothing with 8 different colours and (d) shows the generated body labels for the training depth image using a colour-based segmentation method.

neighbour rule. Due to the noise caused by light, the classification result of every image is further improved by image erosion and image dilation. Because pixels in a RGB image are aligned to the pixels in its corresponding depth image, the labels generated from the RGB images can be used for labelling the depth images.

The complete image pre-processing is shown in Figure 3.3. During data collection, the user's left arm stays in the left side of the body while the right arm stays in the right side. Poses such as crossing arms are not allowed during data collection since such human poses are not helpful to the dressing. The colours of the cloth covered on the human's left hand, left forearm,

left upper arm, left shoulder, right shoulder, and right hand are quite distinguishable in the $L^*a^*b^*$ colour space. Thus when extracting the 2D joint positions for these joints, we directly calculate the mean values of the corresponding body parts. These joints' positions are then used as a reference when calculating the joint positions which requires filtering the outliers. It can be seen from Figure 3.3(d) that some pixels of the left forearm are classified as the right upper arm. When we extract the 2D joint position of the right upper arm, we filter the outliers which locate in the area of the left forearm. Specifically, the x values of the outliers are smaller than the x value of the 2D joint position of the left shoulder while the x values of the pixels which belong to the real right upper arm are larger than the x value of the 2D joint position of the right shoulder. It can also be seen from Figure 3.3(d) that some pixels of the right upper arm are classified as the right forearm. When we extract the 2D joint position of the right forearm, we filter the outliers which locate in the area of the right upper arm. Specifically, the y values of most outliers are smaller than the y value of the 2D joint position of the right upper arm while the y values of the pixels which belong to the real right forearm are larger than the y value of the 2D joint position of the right upper arm.

3.3 UPPER-BODY MOVEMENT SPACE MODELLING

In this section, we propose a user modelling method using vision information to model the movement space of the human upper-body joints.

As we discussed earlier, from a single depth image, we can get the 3D joints positions of a user in the robot coordinates. For the user modelling purpose, a sequence of N depth images is used. We define the joint set of a single depth image as $J_i = \{J_i^1, J_i^2, \dots, J_i^M\}$ where M is the total number of the upper-body joints, and i indicates the depth image. For each joint,

$J_i^m = [x_i^m, y_i^m, z_i^m]$ where $m \in \{1, \dots, M\}$, is the 3D coordinates of the joint. The user's joints' information are the 3D points in the space and are not informative enough for an assistive robot to know the limitations of the upper body. Considering that the working space of each joint is quite different, we use GMMs to model the movement space of each upper-body joint. We define $\mathcal{J}^m = \{J_1^m, J_2^m, \dots, J_N^m\}$, where $\mathcal{J}^m$ represents the set of joint m from N depth images. The Gaussian mixture distribution of $\mathcal{J}^m$ is in the form

$$p(\mathcal{J}^m) = \sum_{k=1}^{K^m} \pi_k^m \mathcal{N}(\mathcal{J}^m | \mu_k^m, \Sigma_k^m) \quad (3.4)$$

We adopt the unsupervised Expectation-Minimisation (EM) learning algorithm in (Figueiredo and Jain, 2002) to estimate the parameters of each Gaussian model. Given $\mathcal{J}^m$, the minimum and maximum number of mixture components, this algorithm outputs the best-selected number of components K^m , the obtained mixture probabilities π_k^m , the estimates of the means μ_k^m , and covariance Σ_k^m of the components.

To fulfil the reachability criterion, the robot should place the clothes to the reachable area of each upper-body joint. For each joint, we take the means μ_k^m of the GMMs as different candidate positions for placing the clothes. This is because the mean of a Gaussian model is the position which maximises the Gaussian probability. When the Gaussian probability is higher, this position is more frequently visited by the corresponding joint. In assistive dressing, we choose the mean of the GMMs of the joint which is the closest to the current joint position and enable the robot to place the clothes to this position for the joint. More details will be discussed in section 3.4.

3.4 PERSONALISED ASSISTIVE DRESSING USING VISION INFORMATION

In this section, we describe how the GMMs of the human upper-body joints are used by assistive robots to provide personalised dressing assistance. A sleeveless jacket is used for the dressing experiments.

We propose an intuitive motion planning method for assistive robots to plan the dressing motion. This is inspired by real scenarios where a human assistant helps another person to wear a sleeveless jacket. A set of goals are sent successively to the robot to execute and the inverse kinematics problems (Nakamura and Hanafusa, 1986) are solved by MoveIt!, the Robot operating system (ROS) motion planning library (Chitta et al., 2012). Table 3.1 shows to which goals the robot’s grippers should move at each step, along with an illustration of the spatial relationships between the robot’s grippers and the human body shown in Figure 3.4.

During assistive dressing, the robot uses two grippers to hold the shoulder areas of a sleeveless jacket respectively, thus the positions of the robot grippers also represent the positions of the jacket shoulders. We enable the robot to assist the user to wear the right part of the jacket first, and then the left part. In the 1st step in Table 3.1, each of the robot’s grippers moves to the position which is behind the corresponding shoulder of the user. In the 2nd step, the robot moves its left gripper to the position which is behind the right shoulder of the user and moves its right gripper to dress the right hand of the user. From the 3rd step to the 5th step, the robot keeps moving its left gripper towards the direction of the left shoulder of the user, while the robot moves its right gripper to dress the user’s right forearm, upper arm, and shoulder. The exact values for the goal positions of the robot’s right gripper are calculated according to the GMMs of the joints and real-time positions of the joints. In the robot coordinates, we represent the current position of the user’s upper-body joint as $\mathbf{p}_{\text{cur}}^m$ and the means of the GMMs of the joint

Table 3.1: Robot motion planning procedures. This table shows to which goals the robot's grippers should move at each step. In the 1st step, each of the robot's gripper moves to the position which is behind the corresponding shoulder of the user. In the 2nd step, the robot moves its left gripper to the position which is behind the right shoulder of the user and moves its right gripper to dress the right hand of the user. From the 3rd step to the 5th step, the robot keeps moving its left gripper towards the direction of the left shoulder of the user, while the robot moves its right gripper to dress the user's right forearm, upper arm, and shoulder. The exact values for the goal positions of the robot's right gripper are calculated according to the GMMs of the joints and real-time positions of the joints. We choose the mean of the GMMs of the joint which is the closest to the current joint position and enable the robot to place the clothes to this position for the joint. After the right part of the jacket is worn on the user body, the jacket adds some constraints on the movement of the robot's left gripper. Thus in the 6th step, the robot moves its left gripper to the left shoulder of the user and the user needs to pull back the left arm to wear the left part of the jacket.

STEP	ROBOT LEFT GRIPPER	ROBOT RIGHT GRIPPER
1	Behind left shoulder	Behind right shoulder
2	Behind right shoulder	Right hand
3	Towards left shoulder	Right forearm
4	Towards left shoulder	Right upper arm
5	Towards left shoulder	Right shoulder
6	Left shoulder	No movement

as μ_k^m , where m indicates the index for the specific joint. We use $\mathbf{d}_k^m$ to represent the vector between $\mathbf{p}_{\text{cur}}^m$ and μ_k^m . For each joint, we enable the robot to place the jacket to μ_l^m which is the closet to $\mathbf{p}_{\text{cur}}^m$, where

$$l = \arg \min_k (||\mathbf{d}_k^m||) \quad (3.5)$$

After the right part of the jacket is worn on the user body, the jacket adds some constraints on the movement of the robot's left gripper. Thus in the final step, the robot moves its left gripper to the left shoulder of the user and the user needs to pull back the left arm to wear the left part of the jacket.

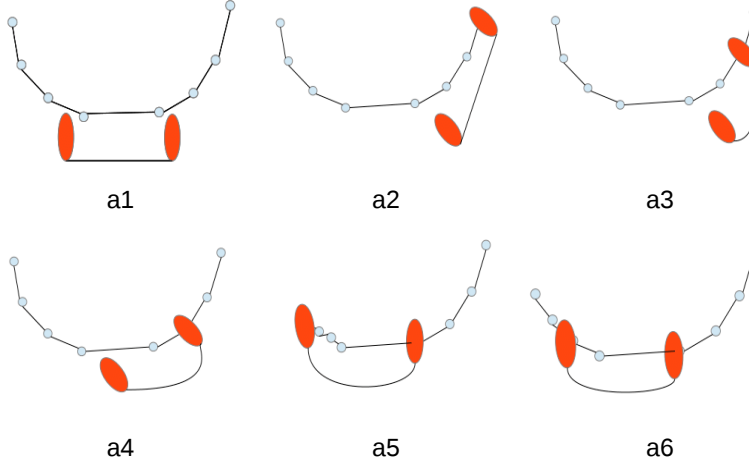


Figure 3.4: An illustration of the robot dressing motion from a top-view. This figure corresponds to the dressing steps shown in Table 3.1. The 2 orange circles represent the positions of the robot’s grippers, which are also the positions of the jacket shoulders. The dressing order is from a1 to a6.

3.5 EXPERIMENTS

We used a Xtion PRO camera which provided RGB and depth images at the frame rate of 30Hz and frame resolutions of 640×480 pixels. The assistive humanoid robot we used in the dressing application was a Baxter robot built by Rethink Robotics. To observe the upper body behaviours of the users from a top view, we mounted the Xtion PRO on top of Baxter’s face screen. We first evaluated the performance of the upper-body pose estimation and movement space modelling. Then we tested the whole system by demonstrating how the Baxter robot made use of the GMMs of the human upper-body joints and real-time upper-body pose estimation to assist a human in wearing a sleeveless jacket. Ten healthy participants (six female) ages 25-33 (mean: 28.2, std: 2.70) participated in the experiments. All computation was conducted on a standard desktop computer with quad-core Intel i7 processor.

3.5.1 *Evaluation of Upper-body Pose Recognition and Movement Space Modelling*

For each participant, we recorded 5,000 pairs of pixel-aligned RGB and depth images of human arm motion without self-occlusions. We instructed the participants to move their arms in one of the following ways by pretending mobility limitations in their right arms: (1) human arms mainly move in the left and right directions (2 participants), (2) human arms mainly move in the forward and backwards directions (2 participants), (3) human arms mainly move in the up and down directions (3 participants), (4) human arms draw circles vertically (3 participants).

For each user, we used 3,000 depth images as training data and the other 2,000 images as testing data. Among the 3,000 training images, we used 30, 300, 1,000, and 3,000 training images to train different tree models with depth 10 and depth 20 separately. A depth 20 tree is deep enough for the number of our training images. For each depth image, we randomly selected 504 pixels where every 63 pixels came from one body part. For each node in the tree model during training, we randomly generated 500 candidate features and 10 candidate thresholds per feature. The colour based segmentation method was used to generate the ground-truth body labels for both the training and testing images. We calculated the average accuracy of classification results from different trained tree models of all the users, which are shown in Table 3.2.

It can be seen that while the depth of the tree model remains the same, the average classification accuracy increases as the number of training images increases. The average classification accuracy is the highest, at 90.82%, when the number of training images is 3,000 and the depth of tree model is 20. When the number of training images is the same, the average classification accuracy is smaller with the depth 10 trees than with the depth 20 trees.

Table 3.2: Average classification accuracy of testing images with different tree models. Ten healthy participants (six female) ages 25-33 (mean: 28.2, std: 2.70) participated in the experiments. For each participant, we recorded 5,000 pairs of pixel-aligned RGB and depth images of the human arm motion without self-occlusions. The colour based segmentation method was used to generate the ground-truth body labels for both the training and testing images. For each user, we used 3,000 depth images as training data and the other 2,000 images as testing data. Among the 3,000 training images, we used 30, 300, 1,000, and 3,000 training images to train different tree models with depth 10 and depth 20 separately. It can be seen that while the depth of the tree model remains the same, the average classification accuracy increases as the number of training images increases. The average classification accuracy is the highest, at 90.82%, when the number of training images is 3,000 and the depth of tree model is 20. When the number of training images is the same, the average classification accuracy is smaller with the depth 10 trees than with the depth 20 trees.

NUMBER OF TRAINING IMAGES	DEPTH OF TREE MODEL	AVERAGE CLASSIFICATION ACCURACY
30	20	44.37%
300	20	66.71%
1000	20	82.44%
3000	20	90.82%
30	10	39.94%
300	10	61.56%
1000	10	78.25%
3000	10	85.63%

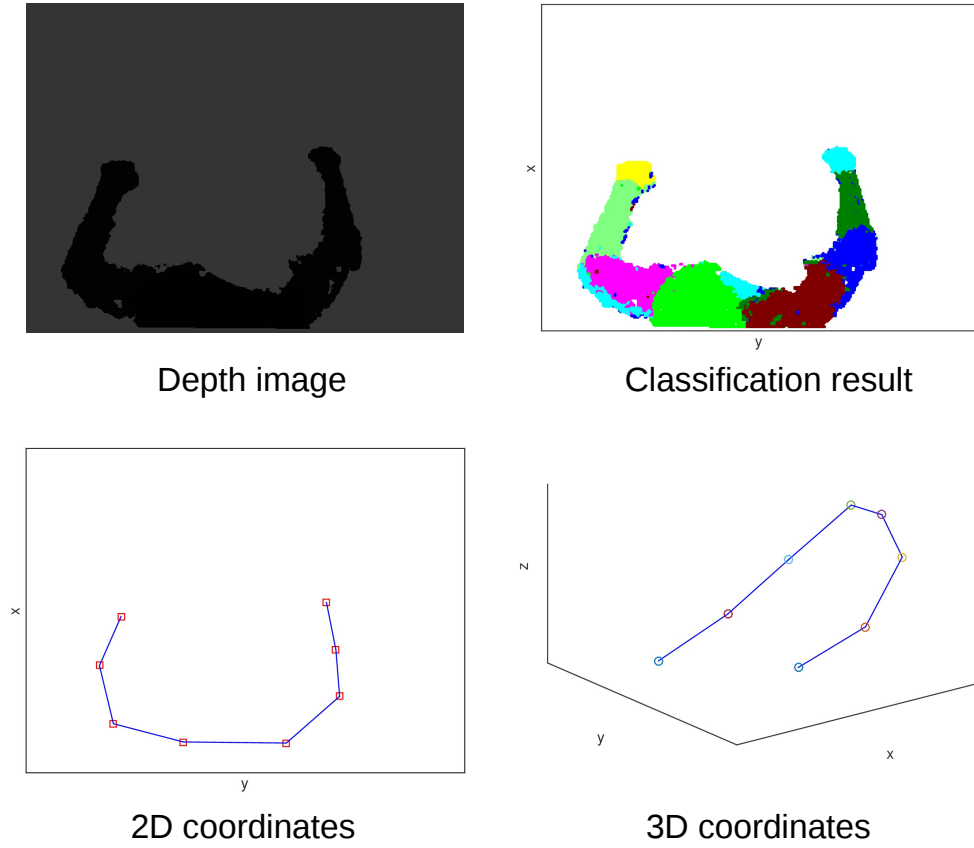


Figure 3.5: Extracting the 3D joint coordinates. From a testing depth image, we first classify every pixel to a joint class with a trained randomised decision tree model. Then, we calculate the 2D coordinates of each body joint after filtering outliers. Finally, we convert the 2D joint coordinates to the 3D joint coordinates using depth information.

For any testing image, we first classify each pixel to a joint class and find the 2D coordinates of each joint. We calculate the mean position of each body part after filtering the outliers and use this position as the joint position. Then, the 2D joint coordinates are converted to the 3D joint coordinates using the depth information. The whole process is shown in Figure 3.5.

For each participant, we modelled the movement space of the upper-body joints in the robot coordinates. In Figure 3.6, we visualise 4 representatives of the GMMs of the upper-body joints. The transparency of individual Gaussian model depends on its mixture probability obtained using the unsupervised EM learning algorithm (Figueiredo and Jain, 2002). When the probability of a position given by the GMMs is higher, it means that this position

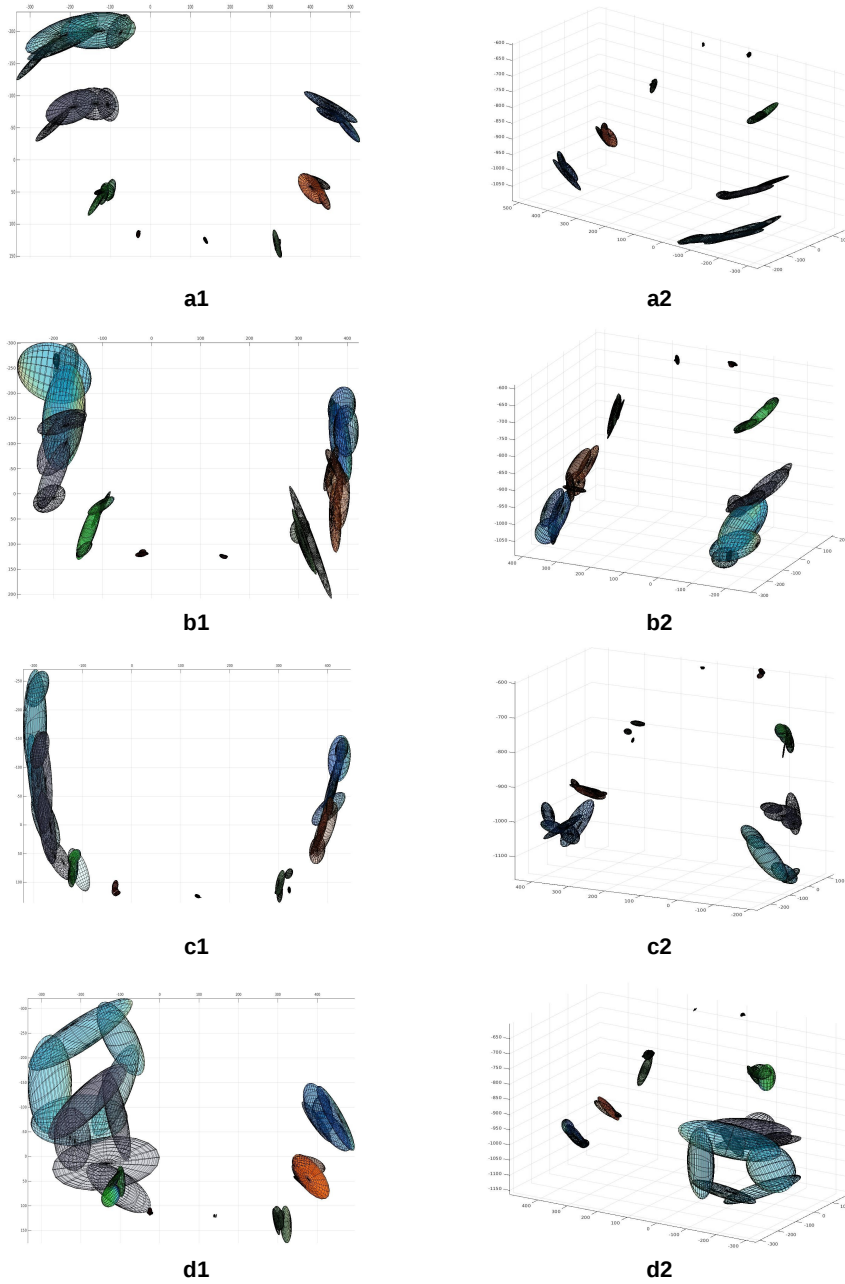


Figure 3.6: Visualisation of 4 representatives of the GMMs of the human upper-body joints. We instructed the participants to move their arms by pretending mobility limitations in their right arms. The movement space of each upper-body joint is modelled with GMMs. The transparency of individual Gaussian model depends on its mixture probability obtained using an unsupervised EM learning algorithm (Figueiredo and Jain, 2002). When the probability of a position given the GMMs is higher, it means that this position is more frequently visited by the corresponding joint. When an area is frequently visited by a joint, the colour of this area becomes denser. Images on the left and right show the visualisation of the GMMs from a top view and a front view respectively. In (a1) and (a2), the human arms mainly move in the left and right directions. In (b1) and (b2), the human arms mainly move in the forward and backwards directions. In (c1) and (c2), the human arms mainly move in the up and down directions. In (d1) and (d2), the human arms mainly draw circles vertically.

is more frequently visited by the corresponding joint. When an area is frequently visited by a joint, the colour of this area becomes denser. Images on the left and right show the visualisation of the GMMs from a top view and a front view respectively. In (a1) and (a2), the human arms mainly move in the left and right directions. In (b1) and (b2), the human arms mainly move in the forward and backwards directions. In (c1) and (c2), the human arms mainly move in the up and down directions. In (d1) and (d2), the human arms mainly draw circles vertically.

3.5.2 *Assisting Users with Dressing*

We tested the whole system on the ten participants by enabling the Baxter robot to provide personalised assistance to help each user wear a sleeveless jacket, where the robot made use of the GMMs of the upper-body joints and real-time upper-body pose estimation. For each user, the robot planned its motion according to the dressing steps shown in Table 3.1. The interval of Baxter moving its grippers from current positions to new positions was set to 3 seconds. The robot moved two grippers slowly while dressing the users. Orientation constraints were added to the robot's grippers during motion planning and a minimum distance was always kept between the robot's grippers to avoid any potential self-collisions. To further guarantee the safety of users, the whole dressing process was under careful supervision by the researchers. Experimental results showed that the robot successfully assisted all the participants to dress with the sleeveless jacket. Figure 3.7 and 3.8 show some sequential shots of the Baxter robot assisting two users to wear a sleeveless jacket individually¹. The dressing order in each figure is from a to f.

¹ The video results can be found at <http://www3.imperial.ac.uk/personalrobotics/videos>

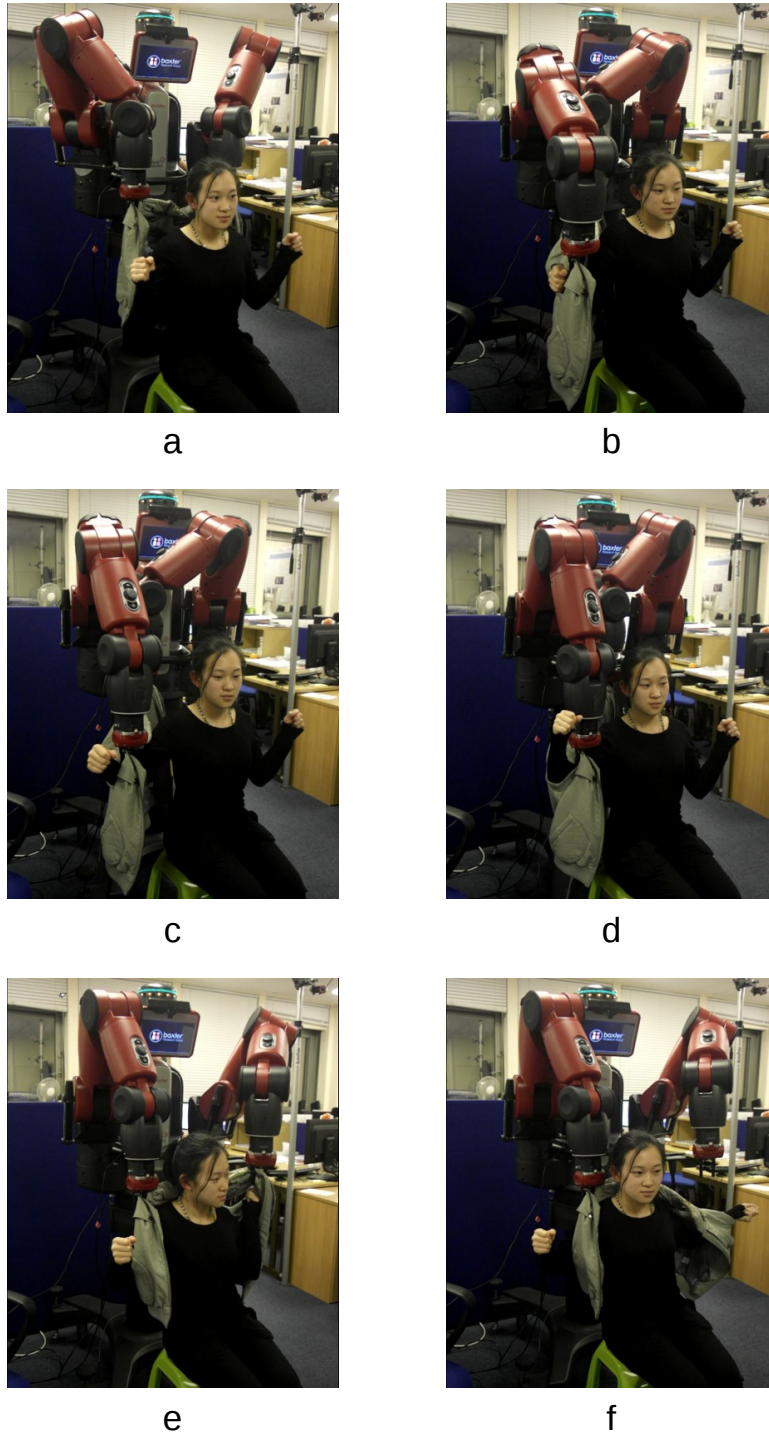


Figure 3.7: Sequential shots 1 of personalised assistive dressing with the user modelling method in this chapter. The Baxter robot planned the dressing motion according to the GMMs of the human upper-body joints and real-time upper-body pose estimation. (a) The robot used two grippers to grasp the shoulder parts of the sleeveless jacket behind the user. (b) The robot assisted the user to dress the right hand. (c) The robot assisted the user to dress the right forearm. (d) The robot assisted the user to dress the right upper arm. (e) The robot assisted the user to dress the right shoulder while the robot's left gripper moved towards the user's left shoulder. (f) The robot assisted the user to dress the left arm.

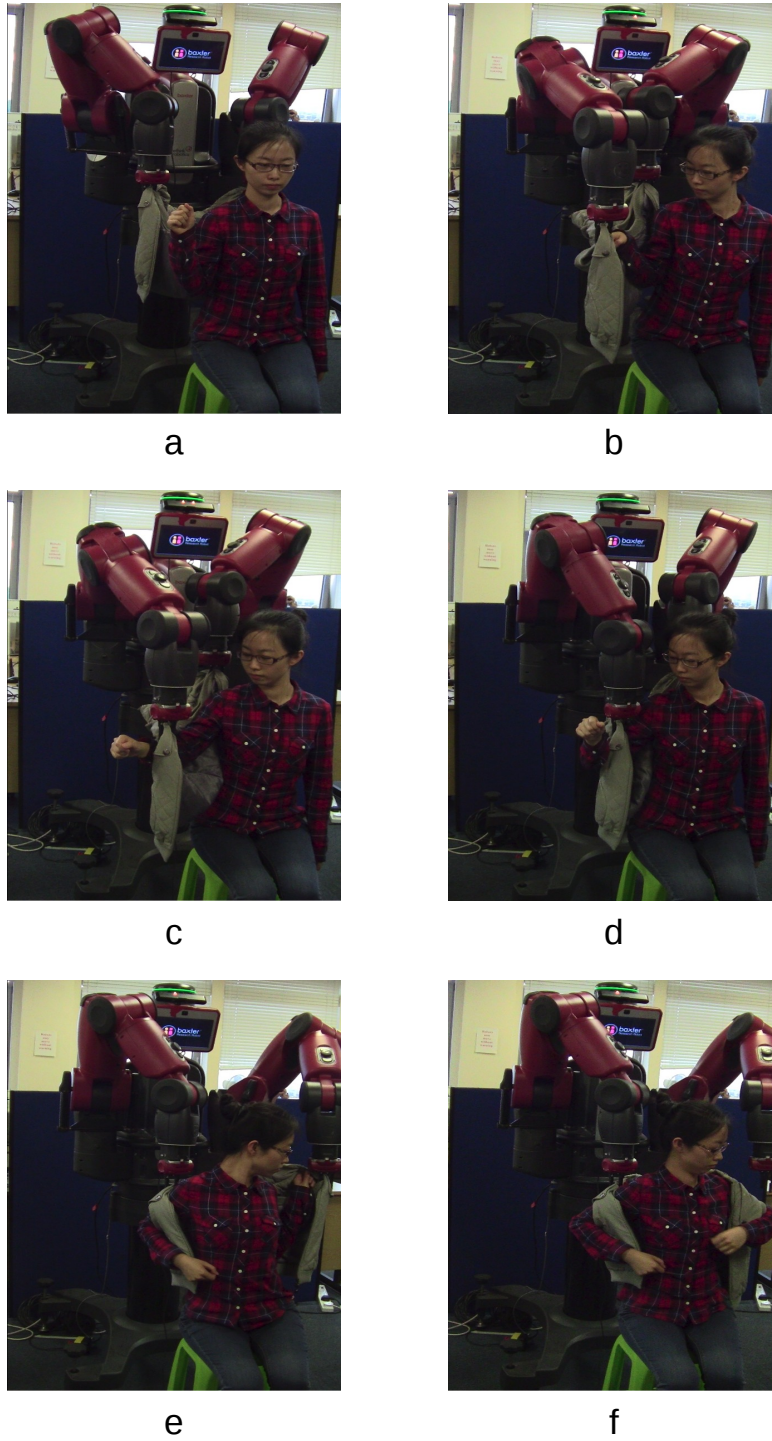


Figure 3.8: Sequential shots 2 of personalised assistive dressing with the user modelling method in this chapter. The Baxter robot planned the dressing motion according to the GMMs of the human upper-body joints and real-time upper-body pose estimation. (a) The robot used two grippers to grasp the shoulder parts of the sleeveless jacket behind the user. (b) The robot assisted the user to dress the right hand. (c) The robot assisted the user to dress the right forearm. (d) The robot assisted the user to dress the right upper arm. (e) The robot assisted the user to dress the right shoulder while the robot's left gripper moved towards the user's left shoulder. (f) The robot assisted the user to dress the left arm.

When we collected the motion data from the participants to model the movement space of the upper-body joints, we instructed the participants to move their arms by pretending one kind of mobility limitations in their right arms. During assistive dressing, we instructed the participants to follow the same mobility limitations when they moved the arms. However, although we took measures to request compliance with the trained behaviours, we had no measure of how compliant they were.

3.6 CONCLUSION

In this chapter, we have presented a user modelling method using vision information. We applied the proposed method to enable a Baxter humanoid robot to provide personalised dressing assistance to real human users by fulfilling the reachability criterion. By mounting a depth camera on top of the robot's face screen, the human upper-body pose is recognised in real time from a single depth image using randomised decision forests. By collecting the motion data of the human upper body, the movement space of each upper-body joint is modelled with GMMs so that we can learn the reachable area of each upper-body joint. To enable the robot to provide personalised dressing assistance, (1) we define a set of dressing steps for the robot's grippers, (2) the exact values for the goal positions in each step are determined according to the GMMs of the human upper-body joints and real-time pose estimation.

We evaluated the proposed system on ten healthy participants. Experimental results on the upper-body pose estimation showed that the highest average classification accuracy was 90.82% when the number of training images was 3,000 and the depth of tree model was 20 for a user. We demonstrated how to model the movement space of the human upper-body joints by visualising 4 representatives of the GMMs of the human body among the

participants. We also evaluated the whole system by enabling the Baxter robot to make use of the GMMs of the human upper-body joints and real-time upper-body pose estimation to plan the dressing motions. Experimental results show that the robot successfully assisted all the participants to dress with the sleeveless jacket.

USER MODELLING USING FORCE INFORMATION

Assistive robots can use the vision information of a human body when dressing a user (Colomé et al., 2015; Gao et al., 2015; Klee et al., 2015). However, occlusions could occur when the robot's arms, the clothes, and the human body are in close contact, which leads to failures in human pose recognition. Thus, other sensor information about humans should be introduced to compensate for the disadvantages of using vision information only. Additionally, due to varying dressing habits, different people may have different preferred paths for the movement of their arms.

In this chapter, we present a user modelling method using force information and enable a Baxter robot to provide personalised dressing assistance to real human users by fulfilling the comfort criterion. As we have mentioned in Chapter 3, the definition of the comfort criterion in this thesis is that there is no external force resistance detected during assistive dressing. To satisfy the comfort criterion, assistive robots should be able to find the optimal personalised dressing path for a user so that the user feels comfortable during the whole dressing process.

Although some vision information is used to decide the starting dressing positions and the initial dressing path for a human user, the focus of this chapter is a user modelling method using force information.

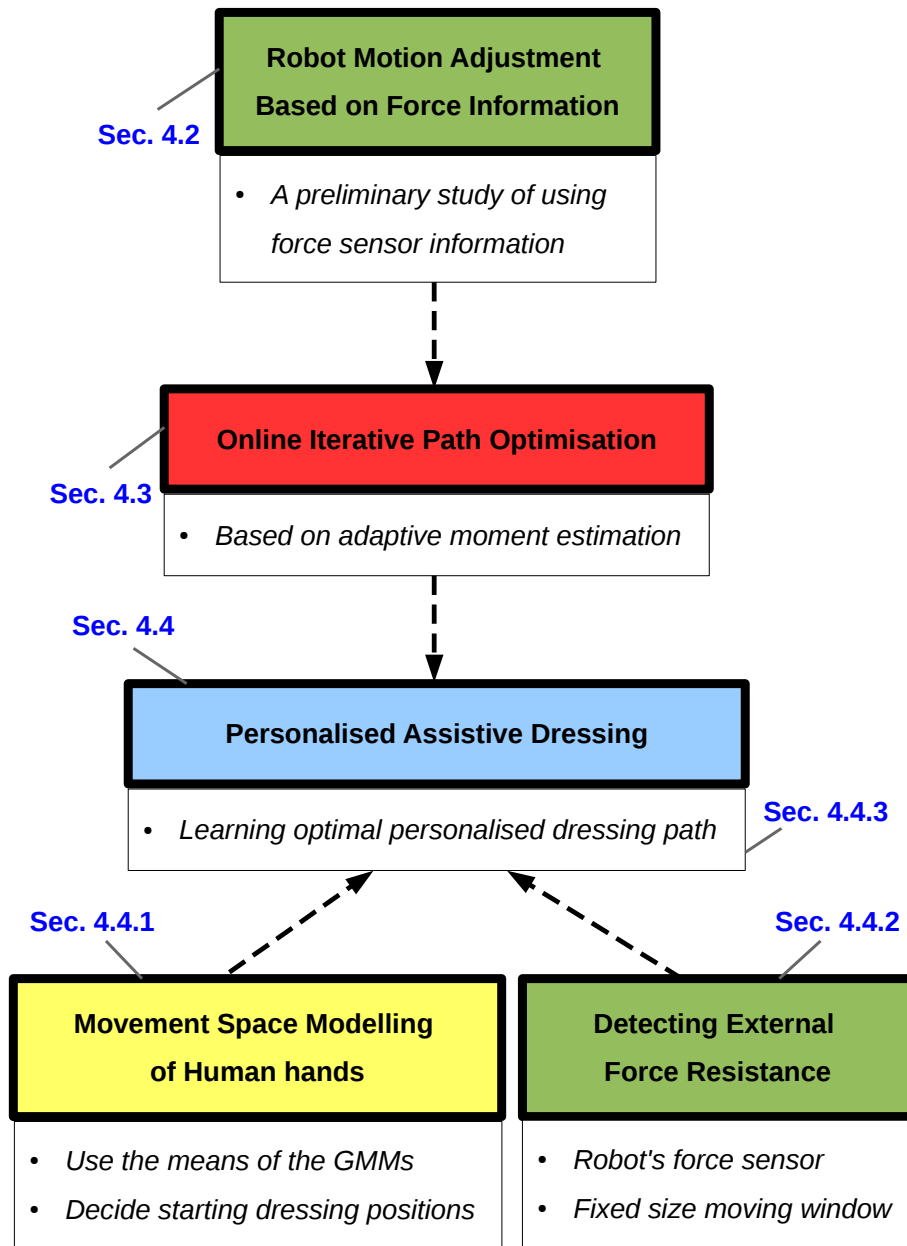


Figure 4.1: This Figure shows the organisation of the sections in this chapter. In section 4.2, we start with a preliminary study of using force sensor information to enable the robot to detect external resistance and locally adjust its motion during assistive dressing. To enable the robot to memorise the updated path and avoid the same resistance in the next round human-robot interaction, we further propose an online iterative path optimisation method to search for an optimal path in the space using force information. The proposed new stochastic path optimisation method in section 4.3 is based on adaptive moment estimation (Kingma and Ba, 2015). We apply the proposed method on personalised assistive dressing in section 4.4 by enabling the robot to iteratively search for the optimal personalised dressing path for a human user. Section 4.4.1 presents how we make use of the GMMs of the human hands to determine the starting dressing positions. Section 4.4.2 presents how we make use of a fixed size moving window to enable the robot to detect the external force resistance during robot's motion execution.

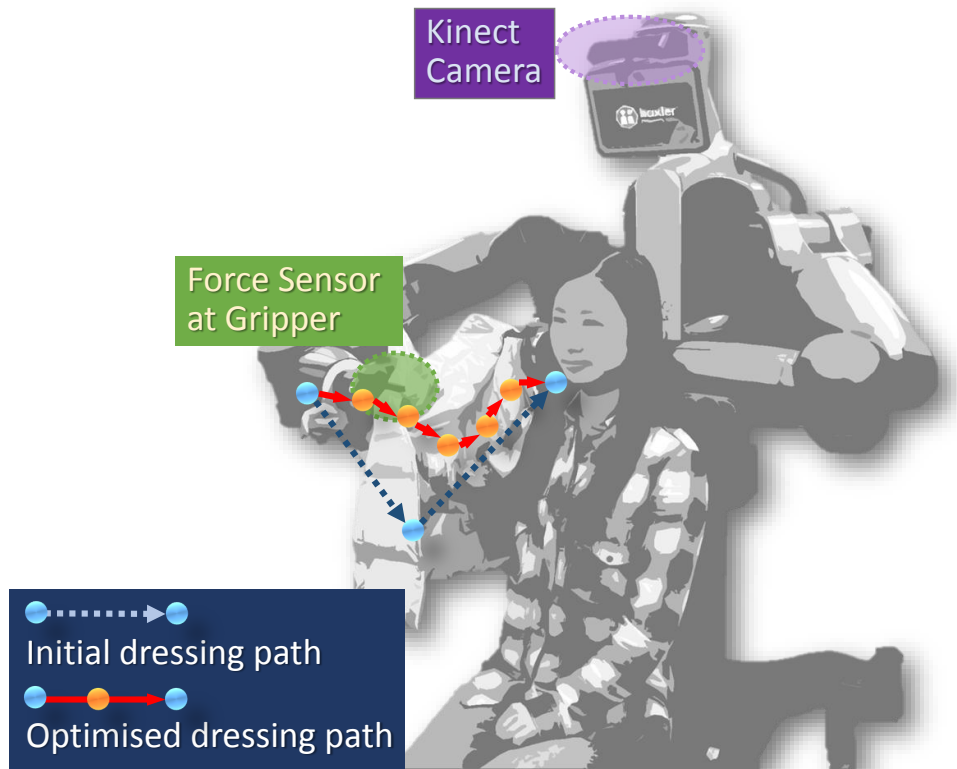


Figure 4.2: The proposed user modelling method in this chapter enables a Baxter humanoid robot to search for the optimal personalised dressing path for a real human user. Force sensor information (green) is used by the robot to detect external force resistance in order to locally adjust its motion. The robot iteratively updates the current dressing path until finding the optimised one (red path connecting orange circles). Vision information (purple) is used to (1) decide the starting dressing positions according to the GMMs of the human hands, (2) decide the initial dressing path (blue circles).

4.1 OVERVIEW

The organisation of the sections in this chapter is shown in Figure 4.1. Each box indicates a section with a few keywords. In section 4.2, we start with a preliminary study using force sensor information to enable the robot to detect external resistance and locally adjust its motion during assistive dressing. To enable the robot to memorise the updated path and avoid the same resistance in the next round of human-robot interaction, we further propose an online iterative path optimisation method to search for an optimal path in the space using force information. The proposed new stochastic path optimisation method in section 4.3 is based on adaptive moment estimation (Kingma and Ba, 2015). We apply the proposed method on personalised assistive dressing in section 4.4 by enabling the robot to iteratively search for the optimal personalised dressing path for a human user. Section 4.4.1 presents how we make use of the GMMs of the human hands to determine the starting dressing positions. Section 4.4.2 presents how we make use of a fixed-size moving window to enable the robot to detect the external force resistance during the robot's motion execution.

Figure 4.2 shows an illustration of the user modelling method using force information in this chapter. The proposed user modelling method enables a Baxter humanoid robot to search for the optimal personalised dressing path for a real human user. Force sensor information (green) is used by the robot to detect external force resistance in order to locally adjust its motion. The robot iteratively updates the current dressing path until finding the optimised one (red path connecting orange circles). Vision information (purple) is used to (1) decide the starting dressing positions according to the GMMs of the human hands, (2) decide the initial dressing path (blue circles).

4.2 ROBOT MOTION ADJUSTMENT BASED ON FORCE INFORMATION

In this section, we study the preliminary use of force sensor information to enable the robot to detect external resistance and locally adjust its motion during assistive dressing.

At the endpoint of each limb, the Baxter robot has force sensors which can detect the current applied force in 3 dimensions. The force coordinates are with respect to the robot's grippers. The spatial relationships between the gripper force coordinates and the robot coordinates can always be acquired using the standard ROS techniques of coordinate transformation. During assistive dressing, we fix the orientations of the robot's grippers, and the spatial relationships between the force coordinates and the robot coordinates are $x_{\text{force}} = -x_{\text{robot}}$, $y_{\text{force}} = -y_{\text{robot}}$, $z_{\text{force}} = -z_{\text{robot}}$. A positive axis in the force coordinates is the corresponding negative axis in the robot coordinates.

To detect external force resistance, for each axis we calculate the force difference at every time step and sum them up within a fixed-size moving window. In the force coordinates, we use F_x^t , F_y^t and F_z^t to represent the sum of force difference in each axis at time step t . When there is external force resistance, there will be a continuous increase or decrease in the force difference, resulting in an increase in $|F_i^t|$, where $i \in \{x, y, z\}$.

According to the dressing order, different goal positions are sent to the robot's grippers to execute. Given a goal position for the robot's gripper, a path is planned using the motion planning library (Chitta et al., 2012). While the robot's gripper executes the planned path, $|F_i^t|$ is checked at each time step. We use τ_f to represent the force threshold. If $|F_i^t| > \tau_f$, then the gripper stops moving and adjusts its gripper from the current posi-

tion $P_{\text{cur}}^t = \{x_{\text{cur}}^t, y_{\text{cur}}^t, z_{\text{cur}}^t\}$ to a new position $P_{\text{new}}^t = \{x_{\text{new}}^t, y_{\text{new}}^t, z_{\text{new}}^t\}$ where

$$\begin{aligned} x_{\text{new}}^t &= x_{\text{cur}}^t - \alpha * F_x^t \\ y_{\text{new}}^t &= y_{\text{cur}}^t - \alpha * F_y^t \\ z_{\text{new}}^t &= z_{\text{cur}}^t - \alpha * F_z^t \end{aligned} \tag{4.1}$$

and α is a constant coefficient which controls the proportion of the adjusted distance caused by the force. With a minus sign in front of F_i^t , the forces are transformed into the robot coordinates. When an external force is detected, the gripper positions are adjusted towards the force direction in 3 dimensions in the robot coordinates. In the experiments, τ_f is set to 5N, the moving window size is 15 time steps and α is set to 0.01.

We evaluate this preliminary approach by enabling the Baxter robot to assist a human user to wear a sleeveless jacket. The robot uses two grippers to grasp the shoulder parts of the jacket and the grippers' positions are also the jacket shoulder positions. The robot assists the user to dress one arm first and then the other arm. In the experiments, we let the robot assist the user to wear the right part of the jacket first.

For the experimenter, we first collected the visual data of the upper-body motions and modelled the movement space of the hands. Then we chose a comfort starting position for the right hand. With an OpenNI skeleton tracker, we use a front-view depth sensor to recognise the upper-body pose by detecting the elbow and shoulder positions, which are sent to the robot to plan its dressing motions. As soon as there is external force resistance detected, the robot stops its current execution and locally adjusts its motions according to the detected force. After the local motion adjustment, the robot keeps moving towards the next goal position until the dressing is finished.

The robot's right gripper path is shown in Figure 4.3. We show the legend in Figure 4.3(a), where the blue circle represents a waypoint in the planned

4.2 ROBOT MOTION ADJUSTMENT BASED ON FORCE INFORMATION

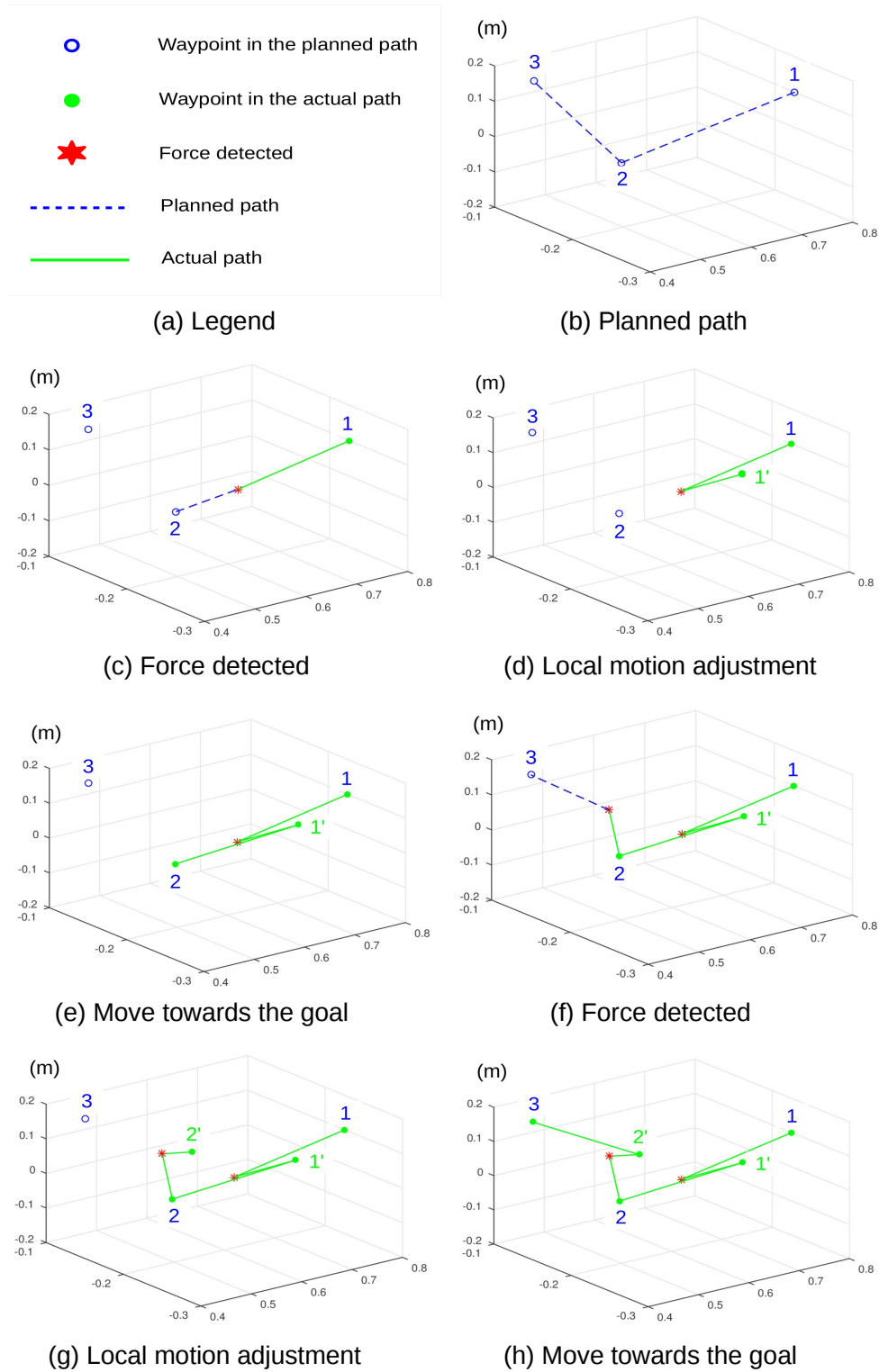


Figure 4.3: This figure shows an example of the robot's right gripper path during assistive dressing. 1, 2 and 3 in blue represent the initial positions of human right hand, elbow and shoulder. The robot assists a user to wear the right part of the jacket following the order of 1, 2 and 3. During assistive dressing, the robot keeps detecting external force resistance and responds to it through locally adjusting its motions. 1' and 2' in green are the adjusted positions. Best viewed in colour.

path, the filled green circle represents a waypoint in the actual path, the red star means that there is external force resistance detected, the blue dashed line represents a planned path, and the green line represents an actual path. Figure 4.3(b) shows the initially planned path with 3 waypoints where 1, 2, and 3 in blue are the positions of the human right hand, elbow, and shoulder. The robot's right gripper first moves to position 1 and then the waypoint position 2 is sent to the robot to reach. While the robot's gripper moves from 1 to 2, external force resistance is detected. The position when detecting force is shown in the red star in Figure 4.3(c). We use the green line to connect position 1 and the red star to show that this is the actual path that the robot gripper has passed. The blue dashed line from the red star to position 2 means that this is the planned path for the robot's gripper to execute if no force is detected. Since external force is detected, the robot stops current execution and locally adjusts its gripper position to 1' in green, which is shown in Figure 4.3(d). The new position 1' is calculated according to the equation (4.1). In Figure 4.3(e), the robot keeps moving towards the goal position 2 from the current new position 1'. In the new planned path, there is no external force disturbance detected. Then the robot's gripper moves towards the goal position 3, where force is detected again in the middle, which is shown in Figure 4.3(f). In Figure 4.3(g), the robot stops executing and adjusts its gripper to position 2'. Finally, the robot moves to the last goal position 3 from the adjusted new position 2'.

In this section, we presented a preliminary study on using force information to enable the robot to locally adjust its motion. We plan to enable the robot to memorise the adjusted path so that the same force resistance could be avoided in the next interaction. In the following sections in this chapter, we further investigate the real-time update of robot dressing path for users during interactions.

4.3 PROPOSED METHOD

In this section, we propose an iterative path optimisation method which can search for the optimal path starting with an initial path based on Adam (Kingma and Ba, 2015). We briefly introduce Adam in section 4.3.1 and then describe the proposed online iterative path optimisation method in section 4.3.2.

4.3.1 *Adaptive Moment Estimation*

The Adam method proposed in (Kingma and Ba, 2015) is for optimisation of stochastic objective functions, where only first-order gradients are required. Adam combines the advantages of two recent stochastic optimisation methods AdaGrad and RMSProp, where AdaGrad performs well with sparse gradients and RMSProp works well in non-stationary and online settings. Adam is described in Algorithm 1 in (Kingma and Ba, 2015).

$f(\theta)$ is a stochastic objective function which is differentiable w.r.t. parameters θ , and g_t represents the gradient evaluated at time step t . The algorithm updates the biased first moment estimate m_t and the biased second raw moment estimate v_t . Since m_0 and v_0 are initialised with 0, the moment estimates m_t and v_t could be biased towards 0 at the beginning of the iteration. The initialisation bias is then counteracted by calculating $\hat{m}_t$ and $\hat{v}_t$. More details about initialisation bias correction and convergence analysis of Adam can be found in (Kingma and Ba, 2015).

4.3.2 *Online Iterative Path Optimisation*

The Adam method is designed to find the global optimum for a stochastic objective function. In this section, we propose an online iterative path op-

Algorithm 1: Adam (Kingma and Ba, 2015)

Require: α : Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0 : Initial parameter vector
 $m_0 \leftarrow 0$ (Initialize 1st moment vector)
 $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
 $t \leftarrow 0$ (Initialize timestep)
while θ_t not converged **do**
 $t \leftarrow t + 1$
 $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
 $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
 $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (g_t)^2$ (Update biased second raw moment estimate)
 $\hat{m}_t \leftarrow m_t / (1 - (\beta_1)^t)$ (Compute bias-corrected first moment estimate)
 $\hat{v}_t \leftarrow v_t / (1 - (\beta_2)^t)$ (Compute bias-corrected second raw moment estimate)
 $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)
return θ_t (Resulting parameters)

timisation method based on Adam which can search for the optimal path starting with an initial path. For assistive dressing with a humanoid robot, the starting and ending positions of the optimised path are set from the vision sensor. We use detected force information to guide the search process. Specifically, when the current path is far away from the optimal path, external force resistance will be detected by the robot. The robot should use the force information to locally adjust its motion and iteratively find the optimal dressing path. How to detect external force resistance during human-robot interactions will be described in detail in section 4.4.2.

The proposed method is described in Algorithm 2. With an initial path, we keep updating until it converges to the optimal path. We define a path after the t^{th} iteration as $\mathcal{W}^t = \{P_1, \dots, P_{i-1}, P_i, P_{i+1}, \dots, P_N\}$, where $i \in \{1, \dots, N\}$, $P_i = \{x_i, y_i, z_i\}$. We represent the initial path as $\mathcal{W}^0$ and the final optimised

Algorithm 2: Online iterative path optimisation

Input : initial path $\mathcal{W}^0$
Output: optimised path $\tilde{\mathcal{W}}$
 Initialisation $m^t, v^t, t \leftarrow 0$
while $t < t_{\max}$ *or* $\mathcal{E}_{\text{energy}} > \tau_{\text{energy}}$ **do**
 $t \leftarrow t + 1$
 $\mathcal{E}_{\text{energy}} \leftarrow 0$
 for all P_i *in* $\mathcal{W}^{t-1}$ **do**
 $\text{UpdatePath}(P_i, P_{i+1}, m^{t-1}, v^{t-1}, t, \mathcal{W}^t, \mathcal{E}_{\text{energy}})$
 $m^t \leftarrow$ get average of all $m^t(n)$
 $v^t \leftarrow$ get average of all $v^t(n)$
Function $\text{UpdatePath}(P_{\text{start}}, P_{\text{end}}, m^{t-1}, v^{t-1}, t, \mathcal{W}^t, \mathcal{E}_{\text{energy}})$ **is**
 Generate path p from P_{start} to P_{end} using motion planning library
 (Chitta et al., 2012)
 for each n^{th} path point $p(n)$ **do**
 Detect $g(n)$
 if $g(n) > \tau_g$ **then**
 $m^t(n) \leftarrow \beta_1 \cdot m^{t-1} + (1 - \beta_1) \cdot g(n)$
 $v^t(n) \leftarrow \beta_2 \cdot v^{t-1} + (1 - \beta_2) \cdot (g(n))^2$
 $\hat{m}^t(n) \leftarrow m^t(n) / (1 - (\beta_1)^t)$
 $\hat{v}^t(n) \leftarrow v^t(n) / (1 - (\beta_2)^t)$
 $p(n) \leftarrow p(n) - \alpha \cdot \hat{m}^t(n) / (\sqrt{\hat{v}^t(n)} + \epsilon)$
 Add updated $p(n)$ to $\mathcal{W}^t$
 $\mathcal{E}_{\text{energy}} \leftarrow \mathcal{E}_{\text{energy}} + g(n)$
 $P_{\text{end}} \leftarrow P_{\text{end}} - \alpha \cdot \hat{m}^t(n) / (\sqrt{\hat{v}^t(n)} + \epsilon)$
 $\text{UpdatePath}(p(n), P_{\text{end}}, m^{t-1}, v^{t-1}, t, \mathcal{W}^t, \mathcal{E}_{\text{energy}})$
 end
 end
 Add P_{end} to $\mathcal{W}^t$
end

path as $\tilde{\mathcal{W}} = \{P_1, \dots, P_N\}$. In assistive dressing, P_i of $\mathcal{W}^t$ is one of the goal positions for the robot's gripper.

When we start the current iteration, we first update the counter t , thus $\mathcal{W}^{t-1}$ represents the current path after the last iteration. We use $\mathcal{E}_{\text{energy}}$ to represent the energy which is the total amount of detected external force resistance in assistive dressing. For all the path waypoints P_i in $\mathcal{W}^{t-1}$, function $\text{UpdatePath}(P_i, P_{i+1}, m^{t-1}, v^{t-1}, t, \mathcal{W}^t, \mathcal{E}_{\text{energy}})$ is called to generate $\mathcal{W}^t$. Inside function UpdatePath , P_i is passed to P_{start} and P_{i+1} to P_{end} . P_{start} is the current starting position for the robot's gripper and P_{end} is the initial

goal position. The gripper path p is planned from P_{start} to P_{end} using the motion planning library (Chitta et al., 2012).

For each n^{th} path waypoint $p(n)$, we check $g(n)$ which is the detected external force. If $g(n)$ is larger than the threshold τ_g , it means that external resistance is detected. In Adam, $g(n)$ denotes the gradients with respect to the stochastic objective at the current time step. In our algorithm, $g(n)$ is the force information where its directions and values guide the current path towards the optimal path. If $g(n) > \tau_g$, we calculate $m^t(n)$, $v^t(n)$, $\hat{m}^t(n)$, $\hat{v}^t(n)$ and update the current path waypoint $p(n)$ following the Adam method. $m^t(n)$ and $v^t(n)$ are the biased first and second moment estimates of $g(n)$. $\hat{m}^t(n)$ and $\hat{v}^t(n)$ are the bias-corrected first and second moment estimates. β_1 and β_2 are the exponential decay rates for the moment estimates. α represents the learning rate and ϵ is the smoothing term. The updated $p(n)$ represents how the gripper locally adjusts its position based on the force information. The robot stops the current execution and moves the gripper to the updated position $p(n)$. This $p(n)$ is added to $\mathcal{W}^t$ and taken as the new starting position for the gripper. The reason to add $p(n)$ to $\mathcal{W}^t$ is because we expect that the same force resistance could be avoided in the next iteration by letting the gripper move towards the updated $p(n)$ directly instead of following the previous path.

Since external resistance is detected, $\mathcal{E}_{\text{energy}}$ is updated with $g(n)$ and the initial goal position P_{end} is updated following the same update rule as $p(n)$. The reason to update P_{end} is because the goal position has changed after external resistance is detected. Then the function calls itself again with the new starting position $p(n)$ and goal position P_{end} . The final updated P_{end} is added to $\mathcal{W}^t$. In another condition, if the $g(n)$ of each $p(n)$ in the planned path is smaller than τ_g , then the original P_{end} is directly added to $\mathcal{W}^t$. For the Adam in (Kingma and Ba, 2015), $m^t(n)$, $v^t(n)$, $\hat{m}^t(n)$, and $\hat{v}^t(n)$ are calculated only once within each iteration to update parameters. However,

when our goal is to search for an optimal path, $m^t(n)$, $v^t(n)$, $\hat{m}^t(n)$, and $\hat{v}^t(n)$ are calculated multiple times for different waypoints. In the proposed method, after we finish checking all the P_i in $\mathcal{W}^{t-1}$, we update m^t and v^t by calculating the mean value of all the $m^t(n)$ and $v^t(n)$ within this iteration and the updated m^t and v^t will be used as m^{t-1} and v^{t-1} in the next iteration.

There are two terminating conditions for the whole iteration process. The first condition is when the total number of iterations exceed the maximum iterations $t_{\max}$. The second condition is when the energy $\mathcal{E}_{\text{energy}}$ is smaller than the energy threshold τ_{energy} . According to (Kingma and Ba, 2015), good default settings for the Adam parameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.

4.4 PERSONALISED ASSISTIVE DRESSING

We apply the proposed method in solving real-world applications for home-environment assistive robots. Our goal is for the robot to iteratively find the optimal personalised dressing path for a person using vision and force sensor information. We use a Baxter humanoid robot and a sleeveless jacket for the dressing assistance.

4.4.1 *Movement Space Modelling of Human Hands*

For dressing assistance, it is significant for assistive robots to know the reachable area of the human hands. For instance, it would be ineffective if an assistive robot selects a starting dressing position which cannot be reached by the user. Thus the movement space of the human hands should be studied before assistive dressing.

In Chapter 3, we proposed modelling the movement space of human upper-body joints using GMMs and we enable the robot to plan dressing motion using both the GMMs of the upper-body joints and real-time upper-body pose estimation. In this chapter, we mainly model the movement space of the human hands and we use the means of the GMMs of the hands as different candidate positions for assistive robots to choose starting dressing positions.

4.4.2 Detecting External Force Resistance

During assistive dressing, due to the movement of human arms, some external force resistance could occur and it should be detected by the robot to adjust its dressing motion. The Baxter robot is equipped with force sensor at the endpoint of each limb. The force coordinates of the robot's grippers are with respect to the endpoints of robot limbs. Whatever the orientations of robot limb endpoints are, the spatial relationships of frame axes between the robot coordinates and endpoint force coordinates can always be acquired using the standard techniques of ROS coordinate transformation. For the dressing task, we let the Baxter robot use two grippers to grasp the shoulder parts of a sleeveless jacket and we fix the orientations of robot's grippers during the whole dressing process. In our experimental set-up, the spatial relationships of frame axes between the robot coordinates and endpoint force coordinates are $(x, y, z)_{\text{gripperForce}} = -(x, y, z)_{\text{robotCoordinate}}$, which is shown in Figure 4.4. Translations between the two coordinates are not considered, because we only concern the force directions in the robot coordinates.

The force value read from each force sensor is the current force which is applied to the robot's limb endpoint in each force axis. To detect external force resistance, we use a fixed-size moving window to calculate force difference at each time step. We represent the combined force of a robot limb

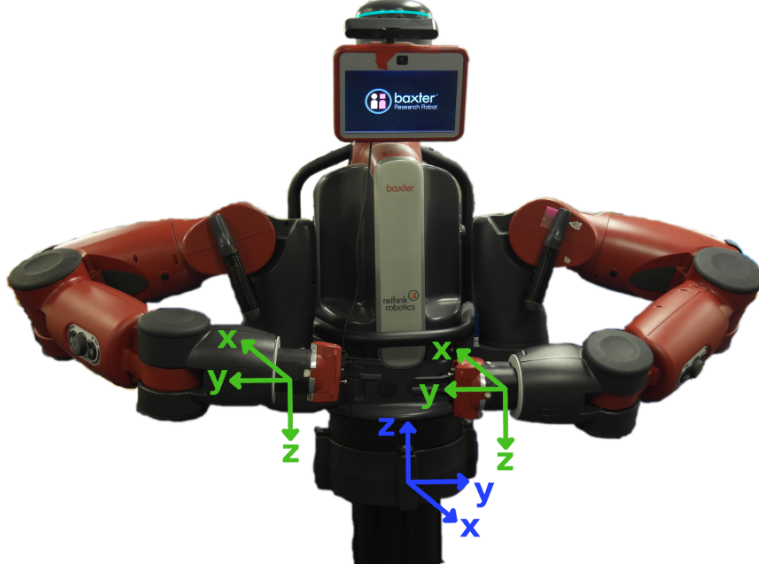


Figure 4.4: The force coordinates of the robot's grippers are shown in green and the Baxter robot coordinate frames are shown in blue. The force coordinates of the robot's grippers are with respect to the endpoints of the robot limbs.

endpoint at time step t as $f^t = \sqrt{(f_x^t)^2 + (f_y^t)^2 + (f_z^t)^2}$, where f_x^t , f_y^t and f_z^t can be directly read. At time step $t + 1$, the combined force difference is $\Delta f^{t+1} = f^{t+1} - f^t$ and the force difference in each axis is $\Delta f_x^{t+1} = f_x^{t+1} - f_x^t$, $\Delta f_y^{t+1} = f_y^{t+1} - f_y^t$, $\Delta f_z^{t+1} = f_z^{t+1} - f_z^t$. We use N to represent the time steps for the moving window. Within the moving window, the sum of combined force difference is $F = \sum_{t=t_i}^{t_i+N} \Delta f^t$ and the sum of force difference in each axis is $F_x = \sum_{t=t_i}^{t_i+N} \Delta f_x^t$, $F_y = \sum_{t=t_i}^{t_i+N} \Delta f_y^t$, $F_z = \sum_{t=t_i}^{t_i+N} \Delta f_z^t$. During robot execution of an action, without external disturbance Δf^{t+1} remains a small positive or negative value at each time step, therefore $|F|$ should be within a force range. If there is external force resistance, Δf^{t+1} will keep being positive or negative within a short time period, thus there will be a quick increase in $|F|$. If $|F| > \tau_f$, where τ_f is defined as a force threshold, it means that external force disturbance is detected. In real assistive dressing experiments, this τ_f is set to 5N and N is set to 15. Forces of the robot's gripper are read at 100Hz.

4.4.3 *Learning Optimal Personalised Dressing Path*

For different people, their arms may behave differently while putting on the clothes. However, a user usually tends to follow certain behaviour pattern for the daily activities. Our target is to let the assistive robot learn the optimal personalised dressing path for a user which consists of a sequence of endpoint positions for the robot's grippers.

With a sleeveless jacket, the robot can choose to wear first either the left part or the right part of the jacket for a human and then wear the opposite part. In this work, we let the robot assist a human to wear the right part of the jacket and then the left part. As the robot uses two grippers to hold the shoulder parts of the jacket, the robot's gripper positions also represent the jacket shoulder positions. The robot first chooses the starting dressing position for a user according to the GMMs of the user's right hand (section 4.4.1), then the user moves the right hand to this starting dressing position. The robot's starting dressing position also becomes the starting position of the human right hand P_{hand} . Starting positions for the human's right elbow P_{elbow} and shoulder $P_{shoulder}$ can then be detected by the depth camera. The initial dressing path for the robot right gripper is defined as $\mathcal{W}^0 = \{P_{hand}, P_{elbow}, P_{shoulder}\}$. Our goal is for the robot to iteratively update the dressing path by detecting external force resistance (section 4.4.2) and locally adjusting its motion until finding the optimised dressing path $\mathcal{W}^* = \{P_{hand}, P_1^*, \dots, P_{N^*}^*, P_{shoulder}\}$. For the optimal path $\mathcal{W}^*$, P_{hand} and $P_{shoulder}$ remain the same as in the initial path $\mathcal{W}^0$. This is because P_{hand} is chosen according to the movement space model of the human hand. For $P_{shoulder}$, although the human arm can move during dressing, the position of the human shoulder is not affected obviously. Therefore, only positions of the middle waypoints need to be updated during assistive dressing. This update process works similarly for the robot left gripper.

In section 4.3.2, we check $g(n)$ for each path waypoint. In assistive dressing, we let $g(n) = F$ where F represents the combined force difference, which is illustrated in section 4.4.2. For the update of path waypoint, the adjustment of position is calculated in x , y , and z axis using F_x , F_y , and F_z respectively. We let $\tau_g = \tau_f$ to represent the force threshold. τ_{energy} is set to 0, which means that when no external force resistance is detected after the current path update, the robot thinks that the user feels comfort with this path and will stop the iterations. The final updated path is then taken as the optimal personalised dressing path. Different with an optimisation process in simulation which can run a large number of times, we set the maximum iterations t_{max} to 8 since the robot is expected to find the optimal path for a person quickly.

4.5 EXPERIMENTS

We evaluated the proposed method on both synthetic dataset and real-world assistive dressing data. With the synthetic dataset, we compared the proposed method with other optimisation methods using vanilla SGD update, momentum update, Adagrad, and RMSProp.

4.5.1 *Synthetic Dataset*

We first evaluated the proposed method with 2D synthetic data by randomly generating 100 pairs of ground-truth optimal paths and initial paths. We cannot run the assistive dressing experiments in simulation since real-time force interaction data is required. In real assistive dressing, the detected force information for the current path waypoint is related to the distance from this waypoint to the optimal path. Thus we can use the distance to simulate the force information. With synthetic data, we first find the closest point on the

optimal path to the current path point and let $g(n)$ represent the Euclidean distance. Then τ_g is used to represent the distance threshold, which is set to 0.02. With synthetic data, the path between two waypoints is planned using linear regression, where the step length is set to 0.05. The energy threshold τ_{energy} is set to 0.05 and the maximum iterations t_{max} is set to 40.

Figure 4.5 shows 8 examples of iteration process with the proposed method. In each example, the magenta line represents the initial path and the red line represents the ground-truth optimal path. The blue lines represent the paths after each iteration and the black dots are the path points on the final optimised path. It can be seen that whether an initial path is close or far away from the optimal path, the final optimised path is very close to the optimal path, where the path points of the final optimised path locate almost exactly on the optimal path.

We compared the proposed method with methods using vanilla SGD update (Rumelhart, Hinton and Williams, 1986), momentum update (Rumelhart, Hinton and Williams, 1986), Adagrad (Duchi et al., 2011), and RMS-Prop (Tieleman and Hinton, 2012) by running 100 experiments with each method. The learning rate α in each method is set to 0.1. The momentum hyperparameter of momentum update is set to its typical value 0.9, and the decay rate hyperparameter of RMSProp is set to its typical value 0.99. We ran the experiments in Matlab without parallel processing. All computation was conducted on a standard desktop computer with quad-core Intel i7 processor.

In each experiment, we calculate the error ε between the final updated path $\mathcal{W}_{\text{final}} = \{[x_1, y_1], \dots, [x_i, y_i], \dots, [x_m, y_m]\}$ and its corresponding ground-truth optimal path. We use d_i to represent the distance from $[x_i, y_i]$ to the optimal path and the error is defined as $\varepsilon = \sum_{i=1}^m d_i$. For each experiment, we record the iteration number and computation time. For each method, we calculate the mean and the standard deviation of the error, iteration number,

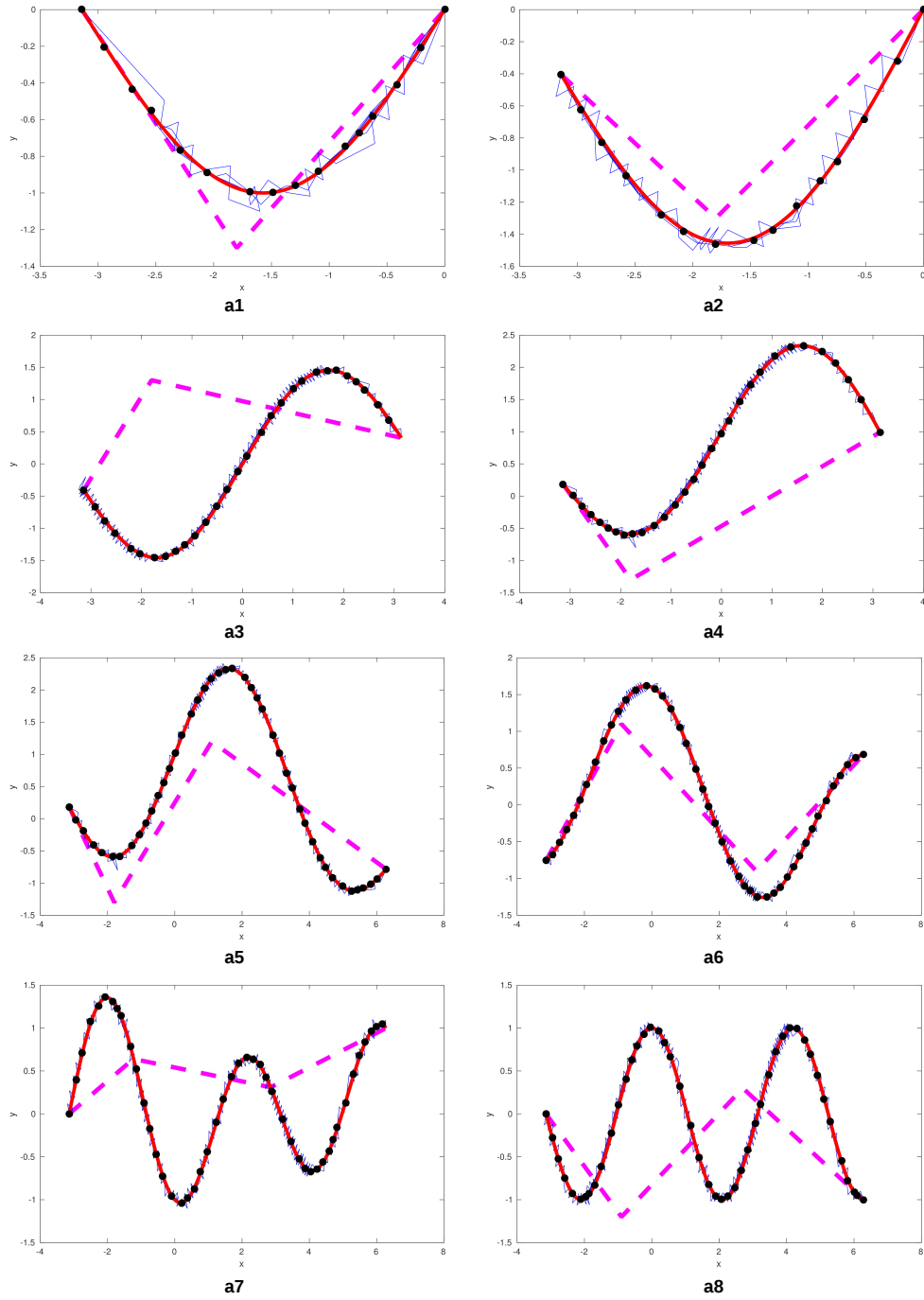


Figure 4.5: This figure shows 8 examples of iteration process with synthetic dataset. In each example, the initial path is in the magenta line and the ground-truth optimal path is in the red line. The updated paths after each iteration are in blue lines and the waypoints of the final optimised path are indicated with black dots. It can be seen that whether an initial path is close or far away from the optimal path, the final optimised path is very close to the optimal path, where the path points of the final optimised path locate almost exactly on the optimal path.

Table 4.1: Evaluation of synthetic dataset with the proposed iterative path optimisation method in Chapter 4. We randomly generated 100 pairs of ground-truth optimal paths and initial paths. We compared the proposed method in this chapter with methods using vanilla SGD update, momentum update, Adagrad, and RMSProp on the synthetic dataset. The error is the sum of the distance from each point in a final updated path to its corresponding optimal path. Iteration number and computation time represent the total number of iterations and the time spent in each experiment. We ran the experiments in Matlab without parallel processing. All computation was conducted on a standard desktop computer with quad-core Intel i7 processor. We calculate the mean and the standard deviation of the error, iteration number, and computation time. It can be seen that the proposed method achieves the smallest average error, iteration number, and computation time comparing with the other four methods. Besides, the proposed method also achieves the smallest standard deviations for the error, iteration number, and computation time. Experimental results with synthetic dataset show that the proposed method can iteratively update a path online and converge within a smaller number of iterations.

Method	Error	Iteration number	Computation time (s)
SGD (Rumelhart, Hinton and Williams, 1986)	12.23 (± 4.79)	35.51 (± 3.01)	1143.58 (± 781.94)
Momentum (Rumelhart, Hinton and Williams, 1986)	7.23 (± 3.49)	30.89 (± 5.65)	1118.02 (± 724.63)
Adagrad (Duchi et al., 2011)	4.99 (± 2.34)	17.17 (± 4.54)	609.83 (± 402.64)
RMSProp (Tieleman and Hinton, 2012)	3.27 (± 1.28)	13.42 (± 3.25)	554.62 (± 297.35)
Proposed	2.08 (± 0.93)	7.99 (± 2.18)	377.90 (± 253.71)

and computation time among 100 experiments. The experiment results are shown in Table 4.1. It can be seen that the proposed method achieves the smallest average error, iteration number, and computation time comparing with the other four methods. Besides, the proposed method also achieves the smallest standard deviations for the error, iteration number, and computation time. Experimental results with synthetic data show that the proposed method can iteratively update a path online and converge within a smaller number of iterations.

As we mentioned before, the learning rate α in each method was set to 0.1. With vanilla SGD update, the learning rate controlled the proportion to the detected resistance when updating the current position and this learning rate was always fixed. This situation looked very similar to the preliminary study in section 4.2 where the coefficient which controlled the proportion to the detected resistance was fixed. In stochastic optimisation, vanilla SGD update may not converge or converge slowly when the learning rate is small enough. Thus with the synthetic dataset, vanilla SGD update performed the worst. In stochastic optimisation, momentum update helps accelerate SGD in any direction that has consistent gradient and dampens oscillations. Thus the convergence rate with the synthetic dataset was faster than the vanilla SGD update. In stochastic optimisation, Adagrad, RMSProp, and Adam all belong to adaptive learning rate methods. Adagrad performs well with sparse gradients by performing larger updates for infrequent parameters and smaller updates for frequent parameters. RMSProp adjusts the Adagrad by using a moving average of squared gradients in order to reduce the monotonically decreasing learning rate of Adagrad. The Adam update not only stores an exponentially decaying average of past squared gradients, but also keeps an exponentially decaying average of past gradients. Thus with the synthetic dataset, the Adagrad outperformed the momentum update, the RMSProp

outperformed the Adagrad, and the proposed method based on Adam outperformed the RMSProp update.

4.5.2 *Real-world Personalised Assistive Dressing*

We evaluated the proposed method by enabling the Baxter robot to find the optimal personalised dressing paths for human users. Twelve healthy participants (seven female) ages 23-32 (mean: 27.2, std: 2.78) participated in the experiments.

We ran five experiments for each participant. We let the robot assist each user to wear the right part of the jacket first followed by the left part. Different with the assistive dressing in Chapter 3 where a user started with an initial upper-body pose, a user started with the arms by his/her side in the experiments in this chapter. The initial dressing position for the user's hand was chosen among the estimates of the GMMs of the hand. After the robot moved the jacket to the chosen starting position, positions of the user's elbow and shoulder were recognised with a front-view depth sensor using the OpenNI skeleton tracker. These positions were used as the initial dressing path for the user's arm. The detected shoulder position was used as the same ending position of the path after each iteration. In each experiment, the robot kept updating the dressing path for a user by detecting external force resistance and adjusting path points with the proposed online iterative path optimisation method.

The method was implemented in Python and all computation was conducted on a standard desktop computer with quad-core Intel i7 processor. For each user, we record the total iteration number, execution time, and energy in each experiment. The energy indicates the total amount of detected external force resistance, which is described by $\mathcal{E}_{\text{energy}}$ in section 4.3.2. For experiments with synthetic dataset, the error can be calculated between the

final updated path and the ground-truth path. However, the ground-truth dressing path of a user is not known before in real-world assistive dressing applications. Thus, we show the results of the energy instead of the error.

Experimental results are shown in Table 4.2. For all the participants, the robot found the optimal dressing paths within a maximum of 5 iterations. Apart from the 3rd, 7th, 8th, and 12th users, the robot averagely spent around 1 minute to finish the path update in one experiment. For the 3rd, 7th, 8th, and 12th users, the robot averagely spent around 2-3 minutes in each experiment and the average detected energy was about 2-3 times than the others. Because the energy is the total amount of detected external force resistance in each experiment, a higher value of energy means that there is more external force resistance detected. The more external force resistance was detected, the more time the robot spent to locally adjust its motion. Thus the average execution time for the 3rd, 7th, 8th, and 12th users was larger than that of the other users.

We show the changes in energy and execution time against iterations, averaged across all experiments and participants in Figure 4.6 and 4.7. In Figure 4.6, we show the energy in each iteration, which is the total amount of detected external force resistance. It can be seen that when the iteration number increases, the median of energy decreases, as well as the 3rd and 4th quartile of the energy. There are 4 outliers in the 1st iteration. This is because, for some users in their 1st experiments, it took them longer time to be familiar with the robot's motion and much more force resistance was generated during their interactions with the robot. Such situations usually became better in the 2nd experiments for the users. In the 3rd iteration, although the median looks similar to the median of the 2nd iteration, both the 3rd and 4th quartile of the energy decrease. The medians in the 4th and the 5th iteration are the same, which are 0N. This is because in some experiments, the energy in the 4th iteration was 0N and the whole iteration terminated. For

Table 4.2: Evaluation of real assistive dressing data with the user modelling method in Chapter 4. We apply the proposed method to enable the Baxter robot to search for the optimal personalised dressing paths for 12 human users. We ran 5 experiments for each participant. Iteration number and execution time represent the total number of iterations and the time spent in each experiment. Energy is the total amount of detected external force resistance in each experiment. Since the ground-truth dressing path of a user is not known before, we show the results of the energy instead of the error. The method was implemented in Python and all computation was conducted on a standard desktop computer with quad-core Intel i7 processor. We calculate the mean and the standard deviation of the iteration number, execution time, and energy. For all the participants, the robot found the optimal dressing paths within a maximum of 5 iterations. Apart from the 3rd, 7th, 8th, and 12th users, the robot averagely spent around 1 minute to finish the path update in one experiment. For the 3rd, 7th, 8th, and 12th users, the robot averagely spent around 2-3 minutes in each experiment and the average detected energy was about 2-3 times than the others. A higher value of energy means that there is more external force resistance detected. The more external force resistance was detected, the more time the robot spent to locally adjust its motion. Thus the average execution time for the 3rd, 7th, 8th, and 12th users was larger than that of the other users.

User	Iteration number	Execution time (s)	Energy (N)
No.1	2.6 (± 1.34)	57.32 (± 50.69)	25.15 (± 32.80)
No.2	2.4 (± 0.89)	57.79 (± 24.09)	25.94 (± 11.06)
No.3	2.4 (± 0.55)	143.04 (± 44.12)	56.24 (± 43.36)
No.4	2.4 (± 0.55)	60.31 (± 19.40)	20.79 (± 8.93)
No.5	2.4 (± 0.89)	47.78 (± 23.90)	18.43 (± 15.60)
No.6	2.6 (± 0.55)	48.35 (± 18.52)	21.31 (± 12.45)
No.7	3.2 (± 0.45)	172.64 (± 22.35)	65.37 (± 19.82)
No.8	3.6 (± 0.55)	190.74 (± 41.86)	73.25 (± 39.83)
No.9	2.6 (± 0.55)	55.87 (± 21.37)	24.46 (± 17.54)
No.10	3.0 (± 0.71)	61.93 (± 26.25)	27.66 (± 23.41)
No.11	2.2 (± 0.45)	46.97 (± 14.77)	19.56 (± 6.82)
No.12	3.8 (± 0.45)	195.26 (± 19.85)	70.43 (± 13.28)

the experiments which did not terminate after the 4th iteration, the energy became 0N in the 5th iteration. In Figure 4.7, we show the execution time in each iteration. The medians of the execution time in each iteration are similar. From the 1st iteration to the 4th iteration, the 3rd and 4th quartile of execution time decrease. This is because in Figure 4.6, the median, as well as the 3rd and 4th quartile of energy decrease from the 1st iteration to the 4th iteration. When the energy decreases, it means that less amount of external force resistance is detected, thus the robot spends less time to locally adjust its motion and the total execution time becomes less. In Figure 4.7, there is an outlier in the 2nd iteration. This is because although most users spent longer time in the 1st iteration and less time in the 2nd iteration, there can be the case that a user still interacted a lot with the robot due to his/her personal preference in the 2nd iteration. For the outlier which is below the median in the 4th iteration, it can be explained that there was no external force resistance detected in that experiment and the robot finished the execution quickly. Another reason can be that since we used position control of MoveIt! motion planning library to plan the robot's motion, the speed of the robot's motion was not directly controlled by us. Thus, sometimes when the robot's motion was smooth, the robot had some acceleration in its motion. For the other 3 outliers which are above the median in the 4th iteration, it can be explained that the corresponding experiments did not terminate in the 4th iteration, which meant that there was external force resistance detected and the robot spent some time to locally adjust its motion.

We use Figure 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, and 4.15 to show a complete process of path iteration with real-world assistive dressing data. The data is from the robot's right gripper path when dressing one of the participants. Figure 4.8 shows the legend used in Figure 4.9, 4.10, 4.11, 4.12, 4.13, and 4.14. Figure 4.9 and 4.10 show the 1st iteration. Figure 4.11 and 4.12

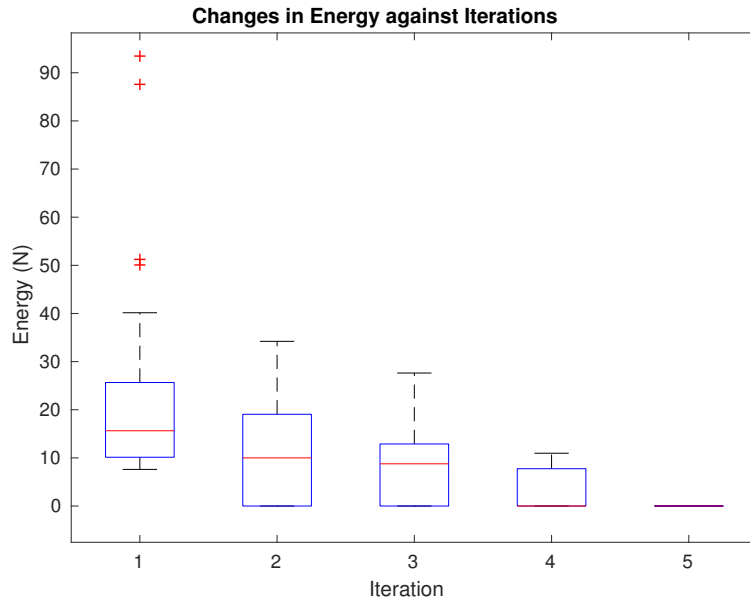


Figure 4.6: Changes in energy against iterations, averaged across all experiments and participants. It can be seen that when the iteration number increases, the median of energy decreases, as well as the 3rd and 4th quartile of the energy.

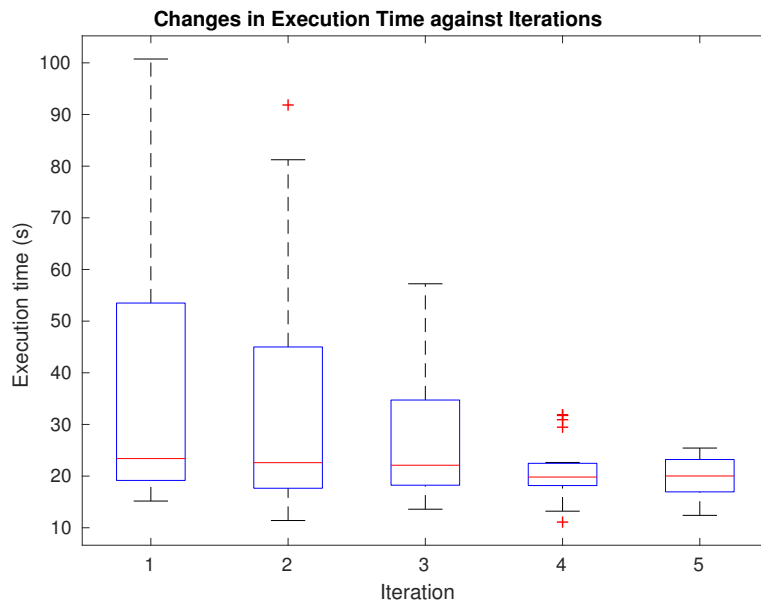


Figure 4.7: Changes in execution time against iterations, averaged across all experiments and participants. The medians of the execution time in each iteration are similar. From the 1st iteration to the 4th iteration, the 3rd and 4th quartile of execution time decreases.

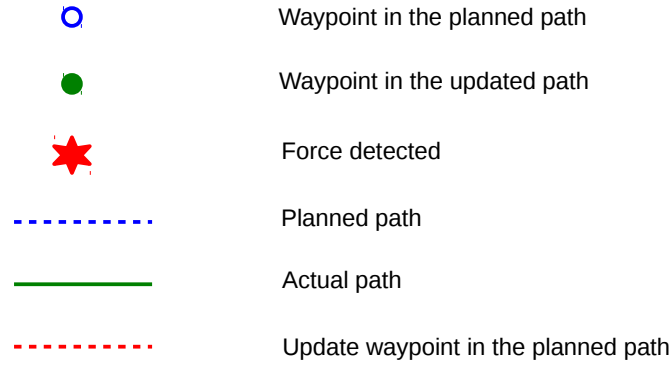


Figure 4.8: The legend used in Figure 4.9, 4.10, 4.11, 4.12, 4.13, and 4.14. An empty blue circle represents a waypoint in the planned path. A filled green circle represents a waypoint in the updated path. A red star represents the position where external force is detected. The blue dashed line is the planned path which is supposed to be executed by the robot's gripper. The green line is the actual path which is actually executed by the robot's gripper. The red dashed line represents how a waypoint in the planned path is updated due to the external resistance.

show the 2nd iteration. Figure 4.13 and 4.14 show the 3rd iteration, which is also the last one. Figure 4.15 shows how the energy changes in each iteration.

In Figure 4.8, we use an empty blue circle to represent a waypoint in the planned path while we use a filled green circle to represent a waypoint in the updated path. We use a red star to represent the position where an external force is detected. A planned path for the robot's gripper is in a blue dashed line. An actual path where the robot's gripper actually follows is in a green line. A red dashed line represents how a waypoint in the planned path is updated due to the external resistance.

Figure 4.9 and 4.10 show the 1st iteration of robot's right gripper path for dressing. In Figure 4.9(a1), the waypoints in the initial dressing path are indicated with 1, 2, and 3 in blue, which are the positions of the user's right hand, elbow, and shoulder respectively. This initial path means that the robot's gripper is supposed to move from 1 to 2, and then from 2 to 3. In Figure 4.9(a2), there is external resistance detected while the robot's gripper moves from 1 to 2. The position when detecting external resistance is indicated with a red star. The path from 1 in blue to the red star is in the

green line, which means that this path is actually executed by the robot's gripper. As we mentioned in section 4.3.2, the starting and ending positions of the optimised path are set from the vision sensor, and they are the same as the starting and ending positions in the initial path. Thus 1 in blue also becomes the starting position in the updated path, which is represented as 1 in green. The path from the red star to 2 in blue is in the blue dashed line, which means that it is supposed to be the planned path for the robot's gripper if no external resistance is detected. Because this path is not really executed by the robot, it is in the blue dashed line instead of the green line. In Figure 4.9(a3), the robot's gripper locally adjusts its position to 2 in green following the proposed method after external resistance is detected, and 2 in green is added to the updated path. The original goal position 2 in blue is then updated to 2' in blue following the same update rule. We use the red dashed line to connect 2 in blue to 2' in blue to represent the waypoint update in the planned path. In Figure 4.9(a4), the robot's gripper moves from the current new position 2 in green towards the updated goal position 2' in blue, while external resistance is detected again. In Figure 4.10(a5), the robot's gripper locally adjusts its position to 3 in green following the proposed method after external resistance is detected, and 3 in green is added to the updated path; the original goal position 2' in blue is then updated to 2'' in blue following the same update rule. In Figure 4.10(a6), the robot's gripper moves from the current new position 3 in green towards the updated goal position 2'' in blue. Since no external resistance is detected, the updated goal position 2'' in blue is directly added to the updated path, and we use 4 in green to replace 2'' in blue. In Figure 4.10(a7), the robot's gripper moves from the current position 4 in green towards the ending position 3 in blue. Since no external resistance is detected, the initial goal position 3 in blue is directly added to the updated path, and we use 5 in green to replace 3 in blue. Position 1, 2, 3, 4, and 5 in green become the new waypoints

in the updated path, which are shown in Figure 4.10(a8). The energy in this iteration is 22.21N, which is the total amount of detected external force resistance.

Figure 4.11 and 4.12 show the 2nd iteration of robot's right gripper path for dressing. Figure 4.11(b1) shows the initial path in this iteration, which is the same as the updated path after the 1st iteration. In Figure 4.11(b2), the robot's gripper moves from the starting position 1 in blue to the first goal position 2 in blue. Since no external resistance is detected, position 1 and 2 in blue are directly added to the updated path and are then represented as 1 and 2 in green. In Figure 4.11(b3), the robot's gripper moves from the current position 2 in green to the goal position 3 in blue. Since no external resistance is detected, position 3 in blue is directly added to the updated path, which is then represented as 3 in green. In Figure 4.11(b4), the robot's gripper moves from the current position 3 in green to the goal position 4 in blue. Since no external resistance is detected, position 4 in blue is directly added to the updated path, which is then represented as 4 in green. In Figure 4.12(b5), the robot's gripper moves from the current position 4 in green towards the ending position 5 in blue, while external resistance is detected. In Figure 4.12(b6), the robot's gripper locally adjusts its position to 5 in green following the proposed method after external resistance is detected, and 5 in green is added to the updated path. Since the original goal position 5 in blue is the ending position, it is still taken as the next goal position without update. In Figure 4.12(b7), the robot's gripper moves from the current new position 5 in green to the goal position 5 in blue. Since no external resistance is detected, position 5 in blue is directly added to the updated path, which is then represented as 6 in green. Position 1, 2, 3, 4, 5, and 6 in green become the new waypoints in the updated path, which are shown in Figure 4.12(b8). The energy in this iteration is 8.54N, which is the total amount of detected external force resistance.

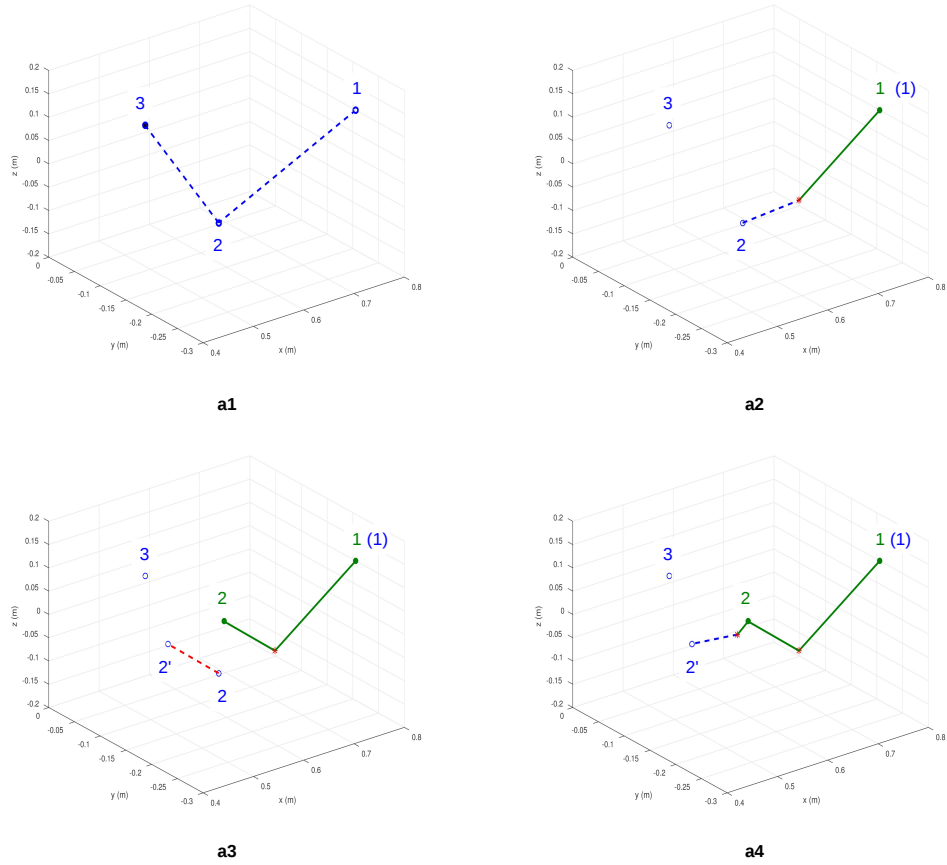


Figure 4.9: 1st iteration: part A. (a1) 1, 2, and 3 in blue indicate the waypoints in the initial path, which are the positions of the user's right hand, elbow and shoulder respectively. (a2) The robot's gripper moves from 1 in blue towards 2 in blue while external resistance is detected. The starting position 1 in blue is also in the updated path, which is represented as 1 in green. The position when detecting external resistance is indicated with the red star. (a3) The robot's gripper locally adjusts its position to 2 in green following the proposed method, and position 2 in green is added to the updated path. The initial goal position 2 in blue is updated to 2' in blue following the same update rule, and we use the red dashed line to connect 2 in blue to 2' in blue to represent the waypoint update in the planned path. (a4) The robot's gripper moves from the current new position 2 in green towards the updated goal position 2' in blue, while external resistance is detected again.

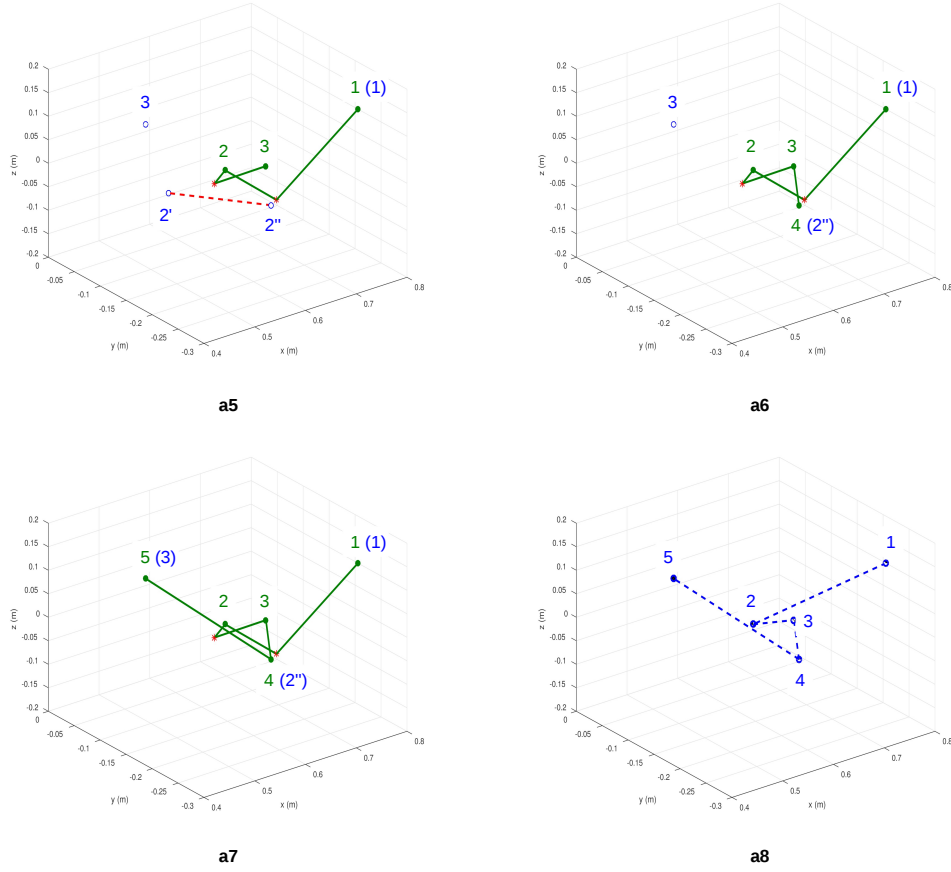


Figure 4.10: 1st iteration: part B. (a5) The robot's gripper locally adjusts its position to 3 in green following the proposed method, and position 3 in green is added to the updated path. The initial goal position 2' in blue is updated to 2'' in blue following the same update rule, and we use the red dashed line to connect 2' in blue to 2'' in blue to represent the waypoint update in the planned path. (a6) The robot's gripper moves from the current new position 3 in green towards the updated goal position 2'' in blue. Since no external resistance is detected, the updated goal position 2'' in blue is directly added to the updated path, and we use 4 in green to replace 2'' in blue. (a7) The robot's gripper moves from the current position 4 in green towards the ending position 3 in blue. Since no external resistance is detected, the initial goal position 3 in blue is directly added to the updated path, and we use 5 in green to replace 3 in blue. (a8) Position 1, 2, 3, 4, and 5 in blue are the new waypoints in the updated path.

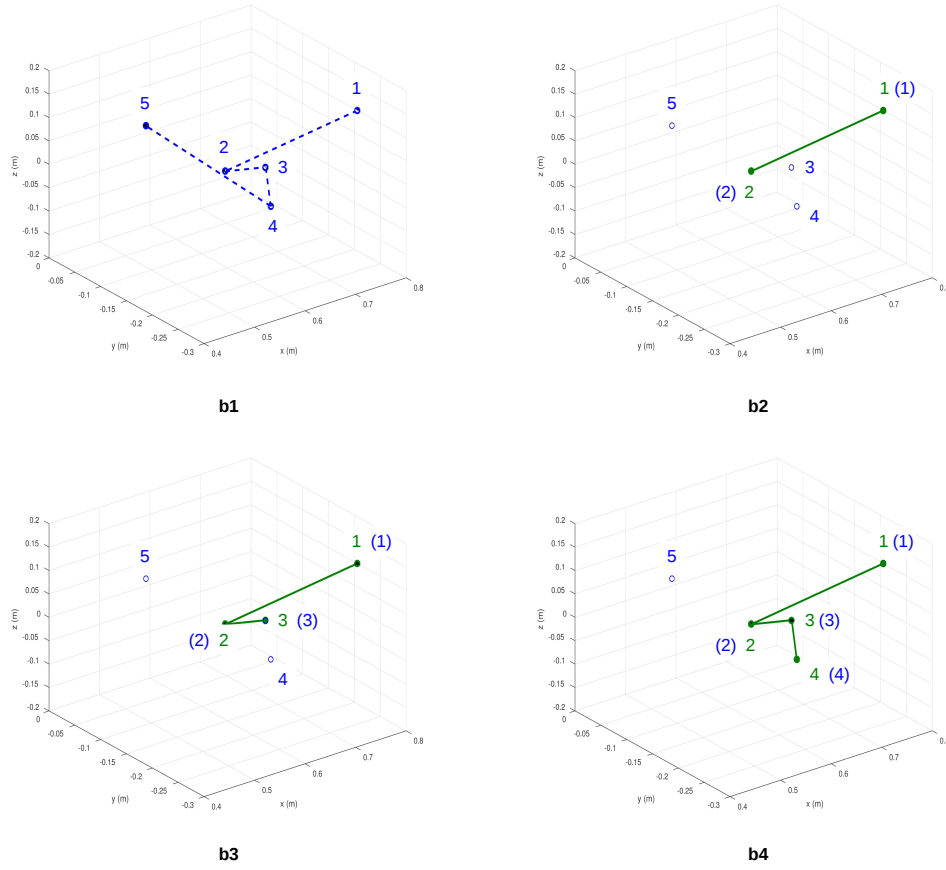


Figure 4.11: 2nd iteration: part A. (b1) shows the initial path in this iteration, which is the same as the updated path after the 1st iteration. (b2) The robot's gripper moves from the starting position 1 in blue to the first goal position 2 in blue. Since no external resistance is detected, position 1 and 2 in blue are directly added to the updated path, which are then represented as 1 and 2 in green. (b3) The robot's gripper moves from the current position 2 in green to the goal position 3 in blue. Since no external resistance is detected, position 3 in blue is directly added to the updated path, which is then represented as 3 in green. (b4) The robot's gripper moves from the current position 3 in green to the goal position 4 in blue. Since no external resistance is detected, position 4 in blue is directly added to the updated path, which is then represented as 4 in green.

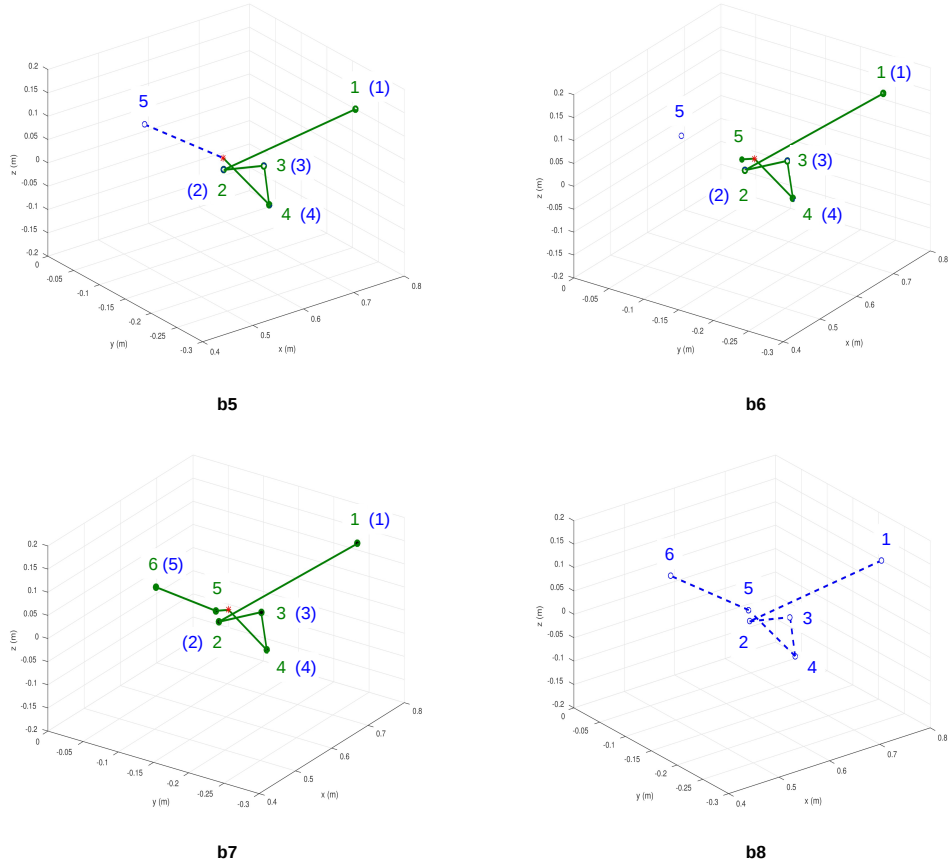


Figure 4.12: 2nd iteration: part B. (b5) The robot's gripper moves from the current position 4 in green towards the ending position 5 in blue, while external resistance is detected. (b6) The robot's gripper locally adjusts its position to 5 in green following the proposed method after external resistance is detected, and 5 in green is added to the updated path. Since the original goal position 5 in blue is the ending position, it is still taken as the next goal position without updating. (b7) The robot's gripper moves from the current new position 5 in green to the goal position 5 in blue. Since no external resistance is detected, position 5 in blue is directly added to the updated path, which is then represented as 6 in green. (b8) Position 1, 2, 3, 4, 5, and 6 in green are the new waypoints in the updated path

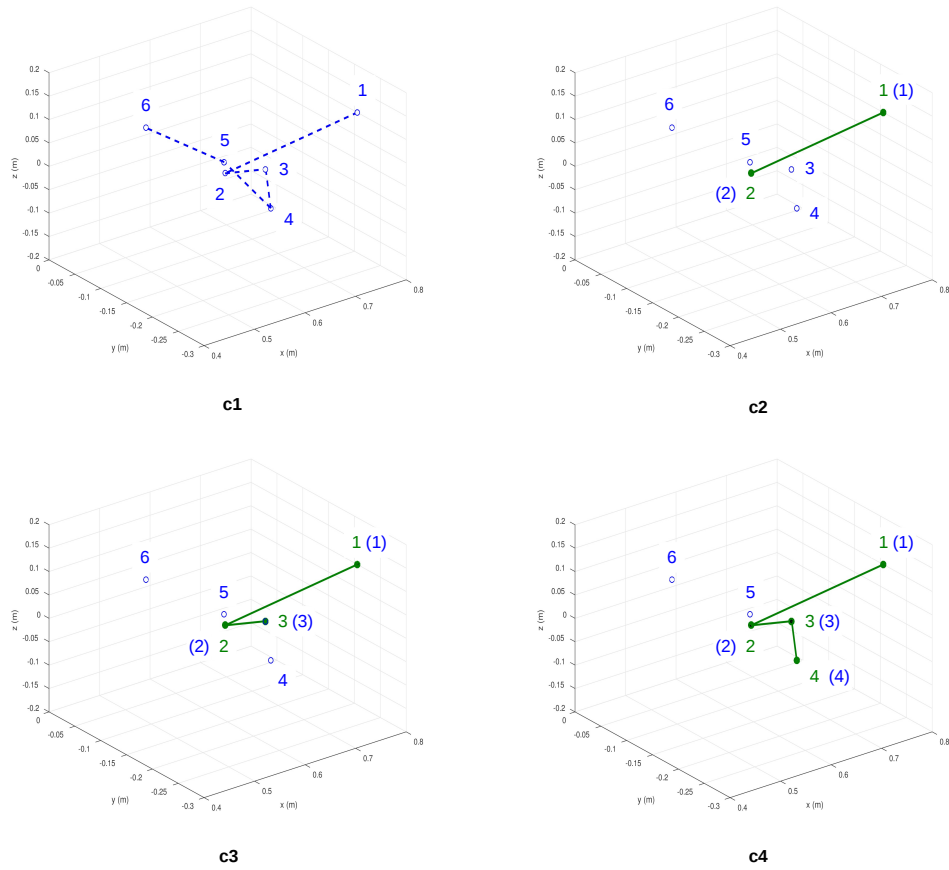


Figure 4.13: 3rd iteration: part A. (c1) shows the initial path in this iteration, which is the same as the updated path after the 2nd iteration. (c2) The robot's gripper moves from the starting position 1 in blue to the first goal position 2 in blue. Since no external resistance is detected, position 1 and 2 in blue are directly added to the updated path and are then represented as 1 and 2 in green. (c3) The robot's gripper moves from the current position 2 in green to the goal position 3 in blue. Since no external resistance is detected, position 3 in blue is directly added to the updated path, which is then represented as 3 in green. (c4) The robot's gripper moves from the current position 3 in green to the goal position 4 in blue. Since no external resistance is detected, position 4 in blue is directly added to the updated path, which is then represented as 4 in green.

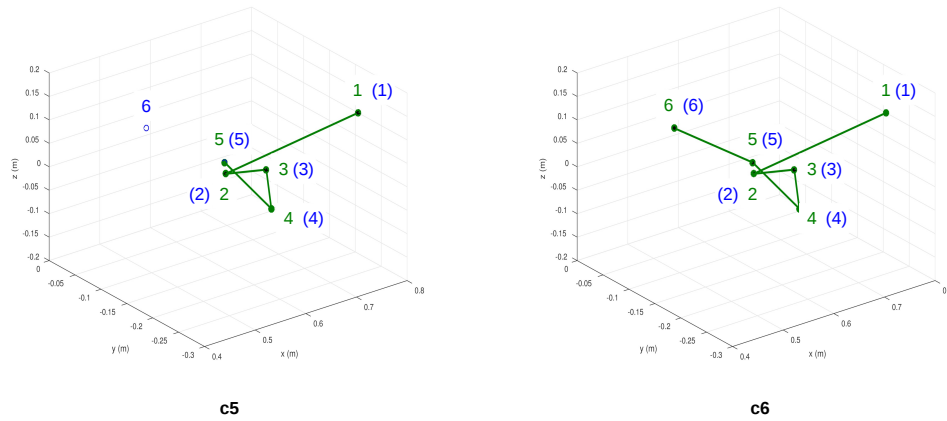


Figure 4.14: 3rd iteration: part B. (c5) The robot's gripper moves from the current position 4 in green to the goal position 5 in blue. Since no external resistance is detected, position 5 in blue is directly added to the updated path, which is then represented as 5 in green. (c6) The robot's gripper moves from the current position 5 in green to the goal position 6 in blue. Since no external resistance is detected, position 6 in blue is directly added to the updated path, which is then represented as 6 in green.

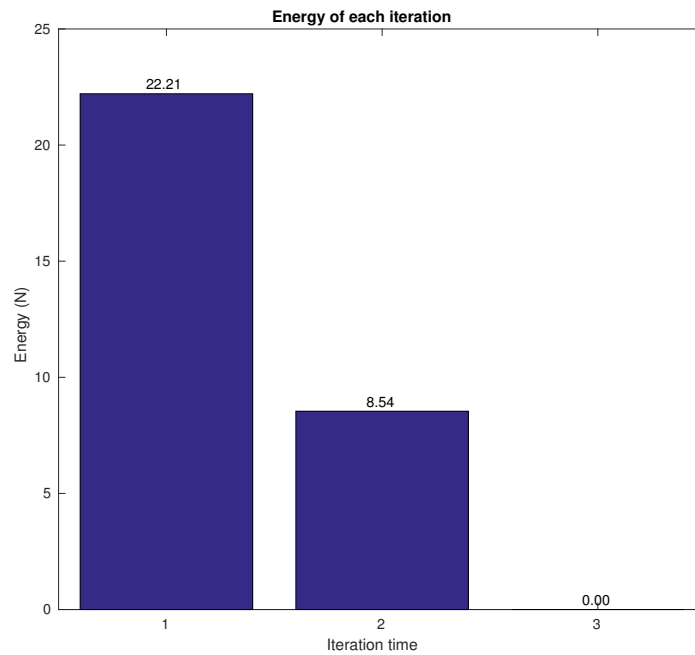
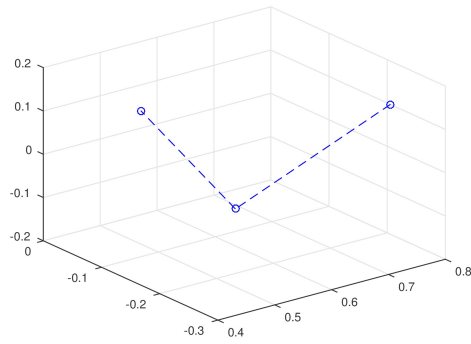
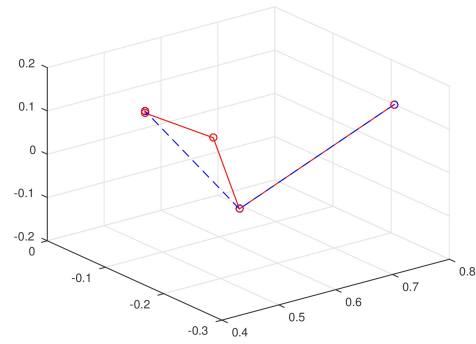


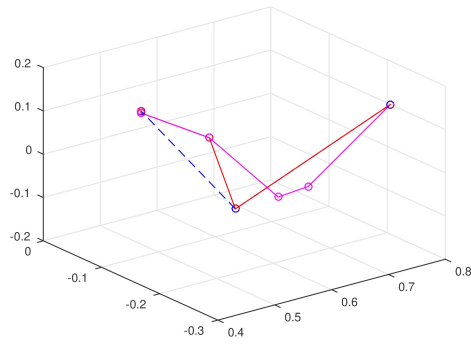
Figure 4.15: The energy of each iteration. This figure shows the energy which is the detected external force resistance in each iteration. The energy in the 1st iteration is 22.21N, which corresponds to Figure 4.9 and 4.10. The energy in the 2nd iteration is 8.54N, which corresponds to Figure 4.11 and 4.12. The energy in the 3rd iteration is 0N, which corresponds to Figure 4.13 and 4.14. After the 3rd iteration, the robot stops iterating and the final updated path becomes the optimal path.



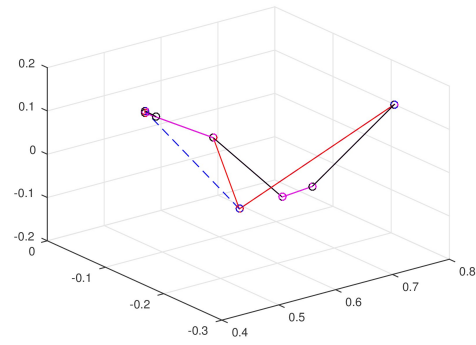
(a) Initial path



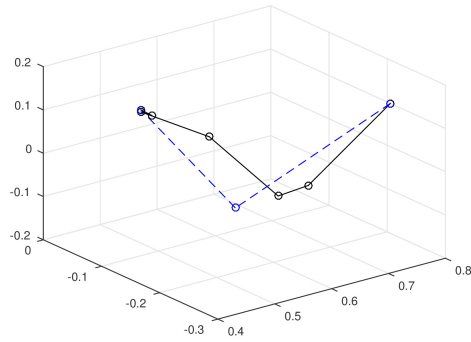
(b) Iter.#1: energy 20.2N



(c) Iter.#2: energy 11.6N



(d) Iter.#3: energy 5.7N



(e) Iter.#4: energy 0N

Figure 4.16: This figure shows how the path is updated from one of the assistive dressing experiment results of one participant. (a) shows the initial path in a blue dotted line. (b) shows the updated path in a red line. (c) shows the updated path in a magenta line. (d) shows the updated path in a black line. (e) shows the final optimised path in a black line comparing with the initial path.

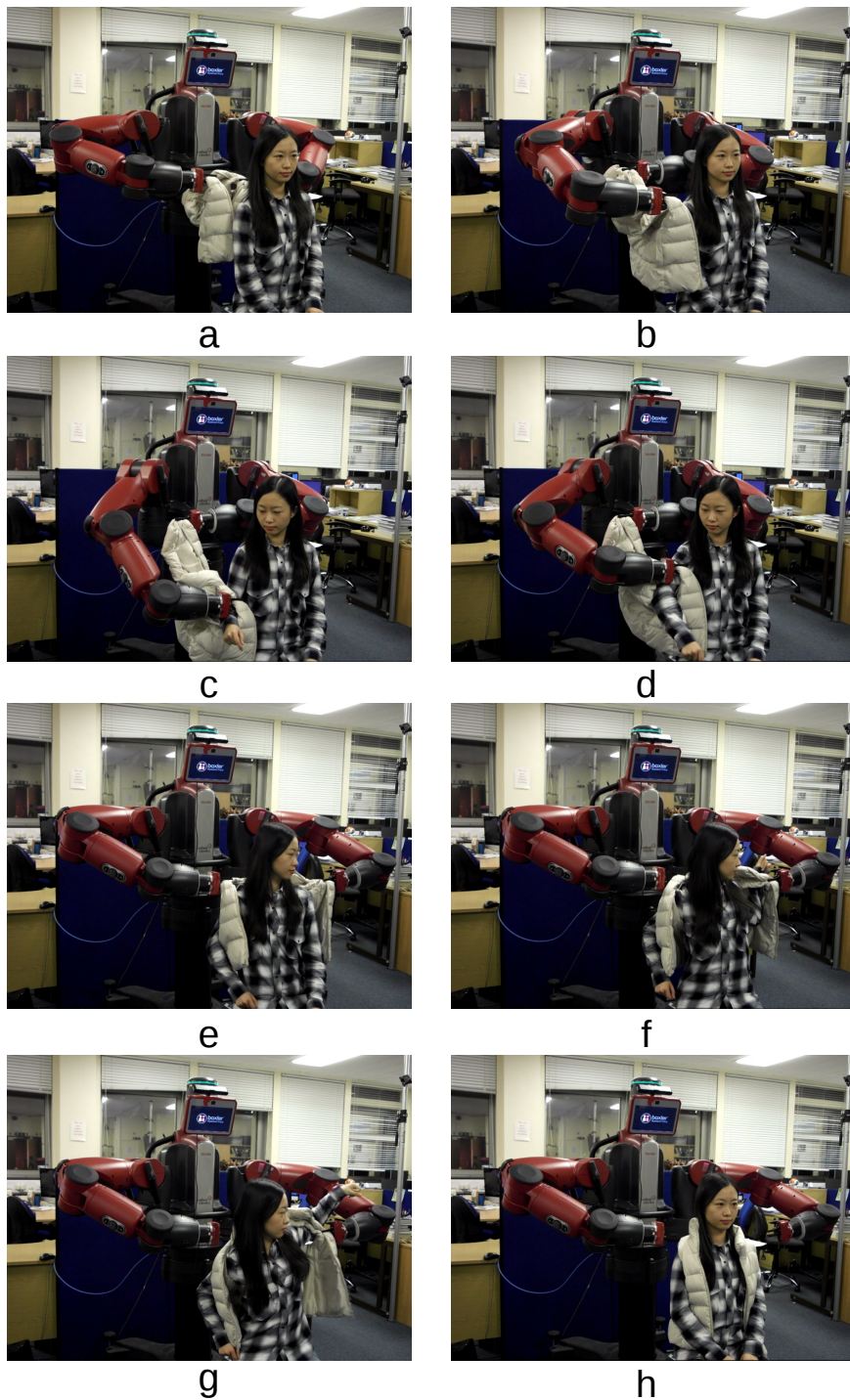


Figure 4.17: Sequential shots of personalised assistive dressing with the user modelling method in Chapter 4. (a) The robot used two grippers to grasp the shoulder parts of a sleeveless jacket. (b) The robot placed the jacket to the starting dressing position for the user's right hand. (c) The robot assisted the user to dress the right hand. (d) The robot assisted the user to dress the right elbow. (e) The robot assisted the user to dress the right shoulder while the robot's left gripper moved towards the user's left shoulder. (f) The robot assisted the user to dress the left hand. (g) The robot assisted the user to dress the left arm. (h) The robot finished assistive dressing.

Figure 4.13 and 4.14 show the 3rd iteration of robot's right gripper path for dressing. Figure 4.13(c1) shows the initial path in this iteration, which is the same as the updated path after the 2nd iteration. The robot's gripper moves from 1 to 2, from 2 to 3, from 3 to 4, from 4 to 5, and finally from 5 to 6. There is no resistance detected during this iteration and the final updated path is the same as the initial path. Since the energy in this iteration is 0N, the robot stops iterating and the final updated path becomes the optimal personalised dressing path for the user's right arm.

Without showing external resistance detection and the local adjustment of robot's gripper motion, we show one path iteration process for wearing the right arm of another participant in Figure 4.16. Figure 4.16(a) shows the initial path in a blue dotted line where the initial path waypoints are indicated by blue circles. Figure 4.16(b) shows the updated path in a red line with new path waypoints indicated by red circles after 1st iteration. It can be seen that one new path waypoint is added in the updated path. The energy of this iteration is 20.2N. Figure 4.16(c) shows the updated path in a magenta line with new path waypoints indicated by magenta circles after 2nd iteration. Comparing with the path after 1st iteration, one old path waypoint is updated and a new path point is added. The energy of this iteration is 11.6N. Figure 4.16(d) shows the updated path in a black line with new path waypoints indicated by black circles. Comparing with the path after 2nd iteration, one new path waypoint is added. Since there is only a small change in the updated path, the energy is 5.7N in this iteration which is smaller than the energy in the last iteration. Figure 4.16(e) shows the final updated path in a black line comparing with the initial path. In the 4th iteration, the robot assists the user to wear the right arm following the updated path after the 3rd iteration. Since there is no external force resistance detected, the final energy is 0N and the robot thinks that the user feels comfort with the current path. Thus the updated path after the 3rd iteration becomes the final updated path

after the 4th iteration. Some screenshots of assistive dressing are shown in Figure 4.17¹.

In the experiment, the robot chose an initial dressing position for the user's hand in the 1st iteration while the user started with the arms by his/her side. This initial position was chosen among the means of the GMMs of the hand and was fixed in the following iterations. It means that the robot always moved the clothes to the same initial position for the user's hand in each iteration. We mentioned and explained this both in Algorithm 2 and in section 4.4.3. For the user's hand, the means of the GMMs represent the positions with a higher probability to be reached. Based on this initial dressing position, we enabled the robot to search for the optimal personalised dressing path. Under different initial dressing positions, the preferred dressing path for the user's arm may not be the same. If we let the user choose the initial dressing position in each iteration, it was hard for the user to make sure that he/she started with the same initial pose each time. Besides, if the user chose the initial dressing position by himself/herself in each iteration, it was very likely that the iteration would never end. Thus we enabled the robot to choose the initial dressing position in each iteration. No matter how the initial dressing position was chosen, as soon as the initial dressing position was the same in each iteration and the user's arm movement followed a certain pattern, the proposed method can iteratively search for the preferred dressing path.

As we have mentioned both at the beginning of Chapter 3 and 4, the definition of the comfort criterion in this thesis is that there is no external force resistance detected during assistive dressing. The definition of the comfort criterion was inspired by the preliminary study of force sensor information in section 4.2. When external force resistance was detected, we enabled the robot to locally adjust its motion according to the directions and values of the

¹ The video results can be found at <http://www3.imperial.ac.uk/personalrobotics/videos>

force information. Due to the vision occlusions, the robot cannot always adjust its motion by observing the human's upper-body motion. With a sleeveless jacket, when external resistance is detected, it actually means that the human's arm moves and the previous dressing path cannot satisfy the user. If the robot finds the movement pattern of the user's arm and performs the dressing assistance following this movement pattern/path, external force resistance should be avoided. Based on the above analysis, we define the comfort criterion in this thesis. Currently, we use only the vision and force information. In the future work, more sensor information can be added to the dressing system, such as sensors that can detect the user's heart rates or temperatures, recognise the user's mood. With more sensors, the definition of the comfort criterion should be modified. For instance, if the user's heart rates or temperatures are below certain thresholds, and the user's mood is classified as happy or neutral, we can believe that the user feels comfortable at the moment. Besides, different users' opinions on the definition of the comfort criterion can be considered in the future in order to give a complete definition. The definition of the comfort criterion can also be influenced by the types of the clothes. If the robot assists human users to dress in a jacket with sleeves, much more force resistance will be detected during assistive dressing. Thus the definition of the comfort criterion when dressing with different types of clothes can be different.

We enable the robot to iteratively search for the optimal personalised dressing path using force sensor information. In our experiment, the ground-truth optimal dressing path for a user is not known before. Since people tend to follow certain movement patterns for their dressing behaviours, if we can learn the movement dynamics of the human upper body, then the ground-truth optimal dressing path can be inferred. However, in this thesis, we did not learn the human dynamics due to the following reasons. First, changes in the human upper-body poses during the whole assistive dressing should

be clear in order to learn the human dynamics. However, severe vision occlusions can occur, which cause failures in the human upper-body pose estimation. Thus we are unable to record the complete series of human upper-body poses throughout the dressing assistance. Second, we may record the user's arm motion without clothes by letting the user perform his/her preferred dressing motion. However, we are then not sure whether the recorded human motion can represent the ground-truth personalised dressing path when interacting with clothes. But if we try to record the human dynamics during assistive dressing, we will meet the occlusion problems discussed before. Based on the above discussion, we choose not to learn the human dynamics, but use force sensor information to infer the personalised dressing path.

4.6 CONCLUSION

In this chapter, we have presented a user modelling method using force information. We applied the proposed method to enable the Baxter robot to search for the optimal personalised dressing path for a human user. We first presented a preliminary study of using force sensor information to enable the robot to detect external resistance and locally adjust its motion during assistive dressing. To enable the robot to memorise the updated path and avoid the same resistance in the next round human-robot interaction, we further proposed an online iterative path optimisation method based on adaptive moment estimation to search for an optimal path in the space.

We first evaluated the proposed method on the synthetic dataset. We randomly generated 100 pairs of ground-truth optimal paths and initial paths, and compared the proposed method with methods using vanilla SGD update, momentum update, Adagrad, and RMSProp. Experimental results showed that the proposed method achieved the smallest average error, itera-

tion number, and computation time comparing with the other four methods. Besides, the proposed method also achieved the smallest standard deviations for the error, iteration number, and computation time. Experimental results with the synthetic dataset showed that the proposed method can iteratively update a path online and converge within a smaller number of iterations.

We also evaluated the proposed user modelling method with twelve healthy participants. For all the participants, the robot found the optimal dressing paths within a maximum of 5 iterations. For 8 out of the 12 participants, the robot averagely spent around 1 minute to finish the path update in one experiment. For 4 out of the 12 participants, the robot averagely spent around 2-3 minutes in each experiment and the average detected energy was about 2-3 times than the others. A higher value of energy means that there is more external force resistance detected. The more external force resistance was detected, the more time the robot spent to locally adjust its motion.

USER MODELLING USING VISION AND FORCE INFORMATION

The main motivation of this thesis is to build user models using multi-modal information and apply the user modelling method to a real assistive robot application, which is personalised assistive dressing by humanoid robots in home environments. As vision is one of the most commonly used sensor information in human-robot interactions, we first study how to use vision information to recognise human upper-body pose and model the movement space of the upper-body joints in Chapter 3. In assistive dressing, the robot makes use of the GMMs of the human upper-body joints and real-time upper-body pose estimation to plan the dressing motion. Then the robot assists the user to dress following the planned path, where feedback from the user is not taken into consideration. However, vision occlusions could cause pose recognition failures during assistive dressing. In Chapter 4, we introduce force sensor information for the robot to detect external force resistance in real time during assistive dressing. As soon as a disturbance is detected, the robot locally adjusts its motion and iteratively searches for the optimal personalised dressing path. Vision information is only used at the beginning of dressing, which is to determine the starting dressing position for the human hand and the initial dressing path. However, vision information is not used in the iterative update process of searching for the optimal dressing path.

With the GMMs of the human upper-body joints, we can learn the reachable area of each joint in the space. With the force sensor information, the robot can detect if the user feels comfortable in real time and adjust its

motion. However, environment noise can also interfere with force sensor information. We hypothesise that by combining the GMMs of the human upper-body joints and the online iterative path optimisation process, the optimal personalised dressing path can fulfil both the reachability and comfort criteria for human users.

5.1 PROPOSED METHOD

In this section, we propose an online iterative path optimisation method using multi-modal information, by combining the GMMs of the human upper-body joints with the online iterative path optimisation method in Chapter 4. We introduce a stick model to model the body part which connects two upper-body joints of the same human arm, where the human upper-body joints are modelled with GMMs. Due to vision occlusions, the robot cannot always recognise the human upper-body pose and know the spatial relationship between the clothes and the human body. Thus the stick model is meant to deal with vision occlusions for the robot to infer the next goal position during assistive dressing. The main difference between the proposed method in this chapter and Chapter 4 is that we update the next goal position of the path by maximising the joint Gaussian probability of the two endpoints of a stick model instead of following the same update rule for the current position.

5.1.1 *Online Iterative Path Optimisation using Multi-modal Information*

We have modelled the movement space of each human upper-body joint independently using GMMs in Chapter 3. However, each upper-body joint movement is not completely independent. For instance, the movement of the human right hand can affect the movement of the right elbow and

shoulder. We make use of a stick model to describe the movement relationships between different upper-body joints on the same human arm. A stick model means that we view the body part which connects two upper-body joints of the same human arm as a straight stick. In computer vision, some of the model-based human body tracking methods are based on a human stick model (Bonnechere et al., 2014; Lai et al., 2012; Moeslund and Granum, 2001). How to make use of a stick model with the proposed method in this chapter will be described in detail in section 5.1.2.

The proposed method is described in Algorithm 3. We define a path after the t^{th} iteration as $\mathcal{W}^t = \{\mathcal{W}_k^t\}_{k=1}^K$ which consists of K sub-paths $\mathcal{W}_k^t$, where $\mathcal{W}_k^t = \{P_{\text{start}}, \dots, P_i, \dots, P_{\text{end}}\}$, $P_i = \{x_i, y_i, z_i\}$. For instance, the path $\mathcal{W}^0$ to dress the user's right arm consists of the sub-path $\mathcal{W}_1^0$ (from the right hand to the right elbow) and the sub-path $\mathcal{W}_2^0$ (from the right elbow to the right shoulder). Before the iteration starts, $\mathcal{W}_k^0 = \{P_{\text{start}}, P_{\text{end}}\}$ where P_{start} and P_{end} are the two endpoints of a stick model. The endpoint of a stick model is a human upper-body joint, where its movement space can be modelled using GMMs.

We use t_{max} to represent the maximum number of iterations. When we start the current iteration, we first update the counter t , thus $\mathcal{W}_k^{t-1}$ represents the current sub-path after the last iteration. We use $\mathcal{E}_{\text{energy}}$ to represent the energy which is the total amount of detected external force resistance in assistive dressing. For all the path waypoints P_i in every sub-path $\mathcal{W}_k^{t-1}$, function *UpdatePath* is called to generate $\mathcal{W}_k^t$. When function *UpdatePath* is called, P_i is passed to P_{cur} and P_{i+1} to P_{next} , and P_{end} is passed to a parameter with the same name P_{end} . Inside function *UpdatePath*, P_{end} is the last waypoint in $\mathcal{W}_k^{t-1}$, P_{cur} represents the current starting position for the robot's gripper, and P_{next} represents the next goal position. The robot's gripper's path p is planned from P_{cur} to P_{next} using the motion planning library (Chitta et al., 2012).

For each n^{th} path waypoint $p(n)$, we check $g(n)$ which is the detected external force. This process is the same as in Chapter 4. If $g(n)$ is larger than the threshold τ_g , it means that external resistance is detected. In Adam, $g(n)$ denotes the gradients with respect to the stochastic objective at the current time step. In our algorithm, $g(n)$ is the force information where its directions and values guide the current path towards the optimal path. If $g(n) > \tau_g$, we calculate $m^t(n)$, $v^t(n)$, $\hat{m}^t(n)$, $\hat{v}^t(n)$ and update the current path waypoint $p(n)$ following the Adam method, which is the same as in Chapter 4. $m^t(n)$ and $v^t(n)$ are the biased first and second moment estimates of $g(n)$. $\hat{m}^t(n)$ and $\hat{v}^t(n)$ are the bias-corrected first and second moment estimates. β_1 and β_2 are the exponential decay rates for the moment estimates. α represents the learning rate and ϵ is the smoothing term. The updated $p(n)$ represents how the gripper locally adjusts its position based on the force information. The robot stops the current execution and moves the gripper to the updated position $p(n)$. This $p(n)$ is added to $\mathcal{W}_k^t$ and taken as the new starting position for the gripper. The reason to add $p(n)$ to $\mathcal{W}_k^t$ is because we expect that the same force resistance could be avoided in the next iteration by letting the gripper move towards the updated $p(n)$ directly instead of following the previous path. Since external resistance is detected, $\mathcal{E}_{\text{energy}}$ is updated with $g(n)$, which is the same as in Chapter 4. To decide the next goal position for the gripper, function *ChooseNextGoal* is called.

Inside function *ChooseNextGoal*, $p(n)$ is passed to P_{mid} and P_{end} is the current last waypoint in $\mathcal{W}_k^{t-1}$. P_{mid} locates somewhere between the endpoints of the stick, but its exact position is not known. Based on the position of P_{mid} , our goal is to update the position of P_{end} , where P_{mid} and the updated P_{end} should be on the same stick model. We propose to update the position of P_{end} by maximising the joint Gaussian probability of the stick endpoints within a random search. We use j_{max} to present the maximum iteration number within function *ChooseNextGoal*. The proposed

Algorithm 3: Online Iterative path optimisation using multi-modal information

Input : initial path $\mathcal{W}^0$
Output: optimised path $\tilde{\mathcal{W}}$
 Initialisation $m^t, v^t, t \leftarrow 0$
while $t < t_{\max}$ *or* $\mathcal{E}_{\text{energy}} > \tau_{\text{energy}}$ **do**
 $t \leftarrow t + 1$
 $\mathcal{E}_{\text{energy}} \leftarrow 0$
 for all P_i *in every* $\mathcal{W}_k^{t-1}$ **do**
 $\text{UpdatePath}(P_i, P_{i+1}, P_{\text{end}}, m^{t-1}, v^{t-1}, t, \mathcal{W}_k^t, \mathcal{E}_{\text{energy}})$
 $m^t \leftarrow$ get average of all $m^t(n)$
 $v^t \leftarrow$ get average of all $v^t(n)$
Function $\text{UpdatePath}(P_{\text{cur}}, P_{\text{next}}, P_{\text{end}}, m^{t-1}, v^{t-1}, t, \mathcal{W}_k^t, \mathcal{E}_{\text{energy}})$ **is**
 Generate path p from P_{cur} to P_{next}
 for each n^{th} path point $p(n)$ **do**
 Detect $g(n)$
 if $g(n) > \tau_g$ **then**
 $m^t(n) \leftarrow \beta_1 \cdot m^{t-1} + (1 - \beta_1) \cdot g(n)$
 $v^t(n) \leftarrow \beta_2 \cdot v^{t-1} + (1 - \beta_2) \cdot (g(n))^2$
 $\hat{m}^t(n) \leftarrow m^t(n) / (1 - (\beta_1)^t)$
 $\hat{v}^t(n) \leftarrow v^t(n) / (1 - (\beta_2)^t)$
 $p(n) \leftarrow p(n) - \alpha \cdot \hat{m}^t(n) / (\sqrt{\hat{v}^t(n)} + \epsilon)$
 Add updated $p(n)$ to $\mathcal{W}_k^t$
 $\mathcal{E}_{\text{energy}} \leftarrow \mathcal{E}_{\text{energy}} + g(n)$
 $\text{ChooseNextGoal}(p(n), P_{\text{end}})$
 $P_{\text{next}} \leftarrow P_{\text{end}}$
 $\text{UpdatePath}(p(n), P_{\text{next}}, P_{\text{end}}, m^{t-1}, v^{t-1}, t, \mathcal{W}_k^t, \mathcal{E}_{\text{energy}})$
 end
 end
 Add P_{next} to $\mathcal{W}_k^t$
end
Function $\text{ChooseNextGoal}(P_{\text{mid}}, P_{\text{end}})$ **is**
 $j \leftarrow 0, p_{\text{jointMax}} \leftarrow 0$
 while $j < j_{\max}$ **do**
 Randomly generate P_{endTmp}^j within a search range
 Relax P_{mid} within a small search range, get P_{midTmp}^j
 Use $P_{\text{endTmp}}^j, P_{\text{midTmp}}^j$ to calculate P_{startTmp}^j
 Calculate probability $p(P_{\text{endTmp}}^j)$ and $p(P_{\text{startTmp}}^j)$
 $p_{\text{joint}}^j = p(P_{\text{endTmp}}^j) \cdot p(P_{\text{startTmp}}^j)$
 if $p_{\text{joint}}^j > p_{\text{jointMax}}$ **then**
 $p_{\text{jointMax}} \leftarrow p_{\text{joint}}^j$
 $P_{\text{end}} \leftarrow P_{\text{endTmp}}^j$
 end
 $j \leftarrow j + 1$
 end
end

method is as following. First, we randomly generate a candidate new position P_{endTmp}^j for P_{end} within a search range. Then, we also relax the position of P_{mid} within a smaller search range to get P_{midTmp}^j . Because P_{endTmp}^j and P_{midTmp}^j are supposed to be on the same stick model and the length of the stick model is known before, we can calculate the candidate position P_{startTmp}^j for the other endpoint of the stick model. How to calculate P_{startTmp}^j will be explained in detail in section 5.1.2. P_{startTmp}^j and P_{endTmp}^j are the two endpoints of a stick model, where they also represent the positions of two upper-body joints on the same human arm. The movement space of each human upper-body joint is modelled with GMMs, so we can calculate the probability $p(P_{\text{endTmp}}^j)$ and $p(P_{\text{startTmp}}^j)$ for the two candidate endpoints given their GMMs. Because we model the movement space of each human upper-body joint independently, we then calculate the joint probability p_{joint}^j for the two endpoints. We use p_{jointMax} to represent the current maximum joint probability. If p_{joint}^j is larger than p_{jointMax} , then we update p_{jointMax} with p_{joint}^j and P_{end} with P_{endTmp}^j . When the iteration time reaches j_{max} , the final P_{end} will be taken as the updated endpoint position for the stick model.

After calling function *ChooseNextGoal*, P_{end} is updated. Back to function *UpdatePath*, we also update P_{next} with P_{end} , where P_{next} is the next new goal position. Then function *UpdatePath* is called again with the updated parameters. The final updated P_{next} is added to $\mathcal{W}_k^t$. In another condition inside function *UpdatePath*, if $g(n)$ of each $p(n)$ in the planned path is smaller than τ_g , then the original P_{next} is directly added to $\mathcal{W}_k^t$.

In the proposed method, after we finish checking all the P_i in every $\mathcal{W}_k^{t-1}$, we update m^t and v^t by calculating the mean value of all the $m^t(n)$ and $v^t(n)$ within this iteration and the updated m^t and v^t will be used as m^{t-1} and v^{t-1} in the next iteration, which is the same as in Chapter 4.

There are two terminating conditions for the whole iteration process. The first condition is when the total number of iterations exceed the maximum iterations $t_{\max}$. The second condition is when the energy $\mathcal{E}_{\text{energy}}$ is smaller than the energy threshold τ_{energy} . According to (Kingma and Ba, 2015), good default settings for the Adam parameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.

5.1.2 Stick Model for Upper-body Joints

We use a stick to model the body part connecting two upper-body joints of the same human arm. For instance, the forearm which connects the hand and the elbow, or the upper arm which connects the elbow and the shoulder is viewed as a stick.

In this section, we present how to use P_{endTmp}^j and P_{midTmp}^j to calculate P_{startTmp}^j in function *ChooseNextGoal*($P_{\text{mid}}, P_{\text{end}}$) in Algorithm 3. Given a stick model, one endpoint position P_{endTmp}^j , and a middle point P_{midTmp}^j which locates between the two endpoints, we calculate the position for the other endpoint of the stick P_{startTmp}^j .

Figure 5.1 shows an example of a stick model in the 3D space. $\mathbf{P}_1$ and $\mathbf{P}_3$ are the two endpoints of the stick model, and $\mathbf{P}_2$ locates between the two endpoints. The length of the stick model L is known before. In assistive dressing, L is the distance between the two joints of the human arm, which can be measured using the vision information. The positions of $\mathbf{P}_1$ and $\mathbf{P}_2$ are known, and we calculate the position of $\mathbf{P}_3$ in the following way:

$$\begin{aligned} \mathbf{v} &= \frac{\mathbf{P}_2 - \mathbf{P}_1}{\|\mathbf{P}_2 - \mathbf{P}_1\|} \\ \mathbf{V} &= L \cdot \mathbf{v} \\ \mathbf{P}_3 &= \mathbf{P}_1 + \mathbf{V} \end{aligned} \tag{5.1}$$

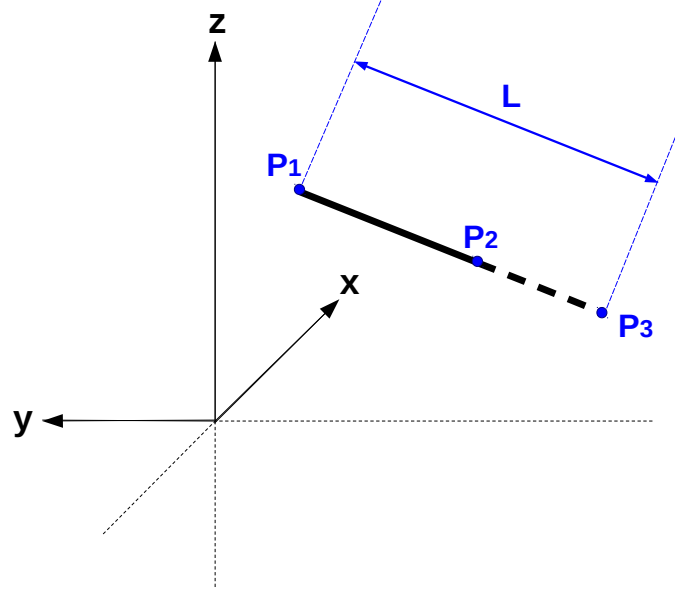


Figure 5.1: 3D illustration of a stick model. Given the length of the stick model L , positions of one endpoint P_1 , and a middle point P_2 , we calculate the position of the other endpoint P_3 . This calculation corresponds to use P_{endTmp}^j and P_{midTmp}^j to calculate $P_{startTmp}^j$ in function *ChooseNextGoal*(P_{mid}, P_{end}) in Algorithm 3.

where $\mathbf{v}$ is the unit vector between P_1 and P_2 , and $\mathbf{V}$ is the vector pointing from P_1 to P_3 . We calculate the position of P_3 by adding the vector $\mathbf{V}$ to the position of P_1 .

5.2 EXPERIMENTS

We evaluate the proposed method on both synthetic dataset and real-world assistive dressing data by comparing with the proposed methods in Chapter 3 and 4. In Chapter 3, we recognise human upper-body pose in real time with a top-view depth sensor and we define 8 upper-body joints, which are L/R hand, forearm, upper arm, and shoulder. In this chapter and Chapter 4, we recognise human upper-body pose in real time with a front-view depth sensor using the OpenNI skeleton tracker. The skeleton tracker provides positions of 6 upper-body joints, which are L/R hand, elbow, and shoulder.

There are some small differences when deciding the initial dressing path between Chapter 3 and 4. In Chapter 3, the robot dresses one of the user's arms following the order of hand, forearm, upper arm, and shoulder. The exact values for the goal positions in each step are determined based on the GMMs of the joints and real-time upper-body pose estimation. In Chapter 4, the initial dressing position for the user's hand is chosen among the estimates of the GMMs of the hand, then the robot moves its gripper to the human elbow and shoulder, where the positions of these human joints are recognised with the front-view depth sensor. In the assistive dressing in Chapter 3, a user started with an initial pose and the robot planned the dressing motion according to both the real-time upper-body pose estimation and the GMMs of the upper-body joints. In the assistive dressing in Chapter 4, a user started with the arms by his/her side and the robot moved the clothes to the chosen initial position for the user's hand according to the GMMs of the hand.

The main contribution in Chapter 3 is the movement space modelling of the human upper-body joints using GMMs and the main contribution in Chapter 4 is the online iterative path optimisation method using force information. To make a comparison among the user modelling methods in Chapter 3, 4, and 5, we use the front-view depth sensor to recognise human upper-body pose and uniformly decide the initial dressing path following the method in Chapter 4, where a user started with the arms by the side. All computation was conducted on a standard desktop computer with quad-core Intel i7 processor.

5.2.1 *Synthetic Dataset*

We first evaluate the proposed method in this chapter with 2D synthetic data. In assistive dressing, the initial dressing path for one arm of the user

contains 3 waypoints which are the positions of the human hand, elbow, and shoulder. We believe that the human joint position which maximises the probability given by the GMMs is the position which can be most frequently reached by this joint. Thus we randomly generate 3 GMMs to decide the 3 waypoints in an initial path. The starting waypoint and the ending waypoint are chosen by maximising the probability given by the corresponding GMMs using a nonlinear programming solver, and the middle waypoint of the initial path is randomly chosen. The main reason to randomly choose the middle waypoint is because we use the starting waypoint, middle waypoint, and ending waypoint to simulate the human joint positions of hand, elbow, and shoulder. The line connecting the starting waypoint and the middle waypoint is a stick model, where the length of the stick model is the Euclidean distance between the starting waypoint and the middle waypoint. The line connecting the middle waypoint and the ending waypoint is another stick model, where its length is the Euclidean distance between the two endpoints. However, the GMMs of each human joint are independent. After we choose the position which maximises the probability given by the 1st GMMs as the starting waypoint, the middle waypoint which locates on the other side of the stick model may not be the position which maximises the probability given by the 2nd GMMs. Thus we allow some flexibility and randomness for choosing the middle waypoint of the initial path with synthetic data.

After determining an initial path, we randomly generate a reference force path for this initial path, where the starting and ending waypoints of the reference force path are the same as the ones of the initial path. This reference force path is similar to the optimal path we generate with synthetic data in section 4.5.1. However, the reference force path here is not the optimal path. It is used for simulating the detected force information. With synthetic data, we first find the closest point on the reference force path to the current path point and let $g(n)$ represent the Euclidean distance in Algorithm 3. τ_g is

used to represent the distance threshold, which was set to 0.1 in the experiment. The path between two waypoints is planned using linear regression, where the step length was set to 0.3. The maximum iteration number $t_{\max}$ for the entire loop was set to 10 and the energy threshold τ_{energy} was set to 0.01. The radius of the search range to generate P_{endTmp}^j was set to 0.5 while the radius of the search range to relax P_{midTmp}^j was set to 0.2. Inside Function *ChooseNextGoal*, the maximum iteration time $j_{\max}$ was set to 20.

Figure 5.2 and 5.3 show the qualitative results with 4 sets of synthetic data. Figures on the left columns show the reference force paths and the initial paths with the GMMs. Figures on the right columns show the final updated paths together with the reference force paths and the initial paths. We use a red line to represent a reference force path, and we use a black line to represent an initial path where waypoints of the initial path are indicated by black circles. The GMMs are shown in green, where the transparency of each Gaussian model depends on its mixture probability. The larger the mixture probability is, the darker the colour of the Gaussian model is. The centre of each Gaussian model is marked with a green cross. The waypoints of the final updated path are indicated with blue circles.

For the 1st dataset, the first, second, and third joints are modelled with 3 GMMs, 2 GMMs, and 1 GMM separately in Figure 5.2(a1). Comparing with the reference force path, the final updated path in Figure 5.2(a2) has more waypoints which are located inside the GMMs area of the second joint. It can be seen that the left part of the final updated path is getting closer to the reference force path while trying to stay inside the GMMs area. Since the right part of the reference force path mostly stays in the GMMs area, the right part of the final updated path is close to the reference force path. Being able to get closer to the reference force path is because of the detected force information which is simulated by the Euclidean distance. Being able to remain in the GMMs area is because the GMMs of the joints is taken

into consideration when updating waypoints positions. It is noted that the GMMs indicates the reachable area while the force information indicates the human comfort.

For the 2nd dataset, the first, second, and third joints are modelled with 3 GMMs, 3 GMMs, and 3 GMMs separately in Figure 5.2(b1). The initial path mostly remains in the GMMs area while the middle part of the reference force path rarely stays in the GMMs area. It can be seen from Figure 5.2(b2) that the starting and ending part of the final updated path is close to the reference force path. The middle part of the final updated path lies in the middle area between the initial path and the reference force path, which is a compromise between the reachability and the comfort criteria.

For the 3rd dataset, the first, second, and third joints are modelled with 2 GMMs, 4 GMMs, and 2 GMMs separately in Figure 5.3(c1). Since both the initial path and the reference force path mostly remain in the GMMs area, the final updated path in Figure 5.3(c2) is close to the reference force path only with a little compromise between the reachability and the comfort criteria. It shows that as long as the reference force path mostly remains in the GMMs area, the detected force information which is simulated by the Euclidean distance becomes the main factor in the iterative process.

For the 4th dataset, the first, second, and third joints are modelled with 2 GMMs, 3 GMMs, and 2 GMMs separately in Figure 5.3(d1). The left part of the reference force path mostly remains in the GMMs area while the right part rarely remains. Thus in Figure 5.3(d2), the left part of the final updated path is close to the reference force path, where the detected force information is the main factor which influences the path update. The right part of the final updated path is getting closer to the reference force path while trying to stay inside the GMMs area, which is a compromise between the reachability and the comfort criteria.

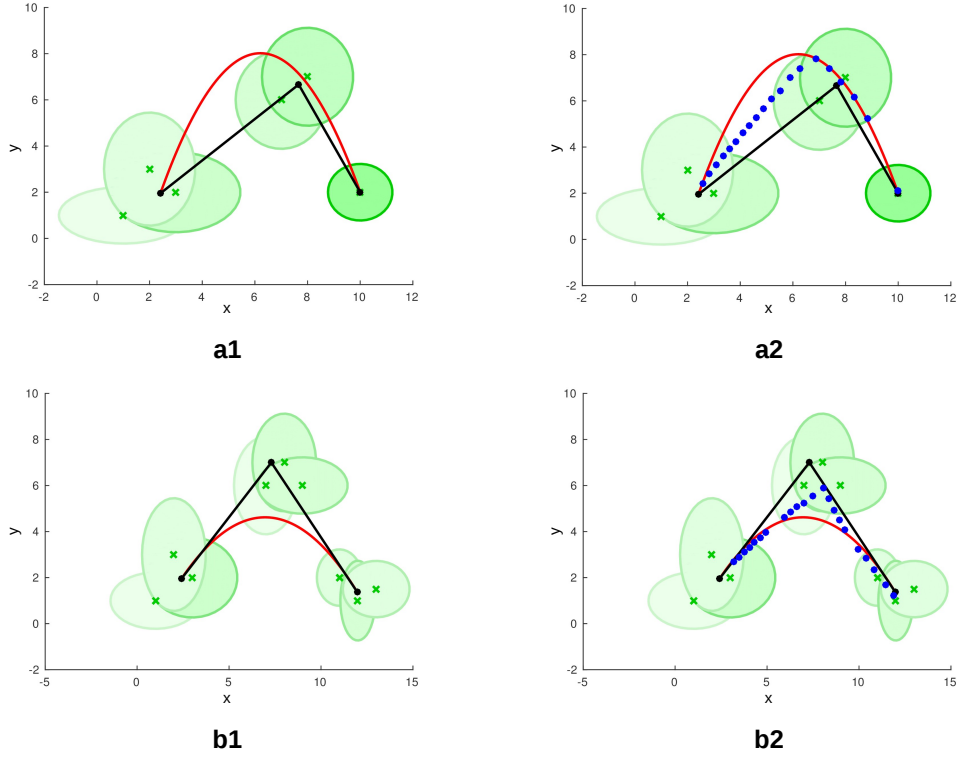


Figure 5.2: Experimental result 1 with synthetic dataset using the proposed method in Chapter 5. We use a red line to represent a reference force path, and we use a black line to represent an initial path where waypoints of the initial path are indicated by black circles. The GMMs are shown in green, where the transparency of each Gaussian model depends on its mixture probability. The larger the mixture probability is, the darker the colour of the Gaussian model is. The centre of each Gaussian model is marked with a green cross. The waypoints of the final updated path are indicated with blue circles. (a1) For the 1st dataset, the first, second, and third joints are modelled with 3 GMMs, 2 GMMs, and 1 GMM separately. (a2) Comparing with the reference force path, the final updated path has more waypoints which are located inside the GMMs area of the second joint. It can be seen that the left part of the final updated path is getting closer to the reference force path while trying to stay inside the GMMs area. Since the right part of the reference force path mostly stays in the GMMs area, the right part of the final updated path is close to the reference force path. (b1) For the 2nd dataset, the first, second, and third joints are modelled with 3 GMMs, 3 GMMs, and 3 GMMs separately. The initial path mostly remains in the GMMs area while the middle part of the reference force path rarely stays in the GMMs area. (b2) The starting and ending part of the final updated path is close to the reference force path. The middle part of the final updated path lies in the middle area between the initial path and the reference force path, which is a compromise between the reachability and the comfort criteria.

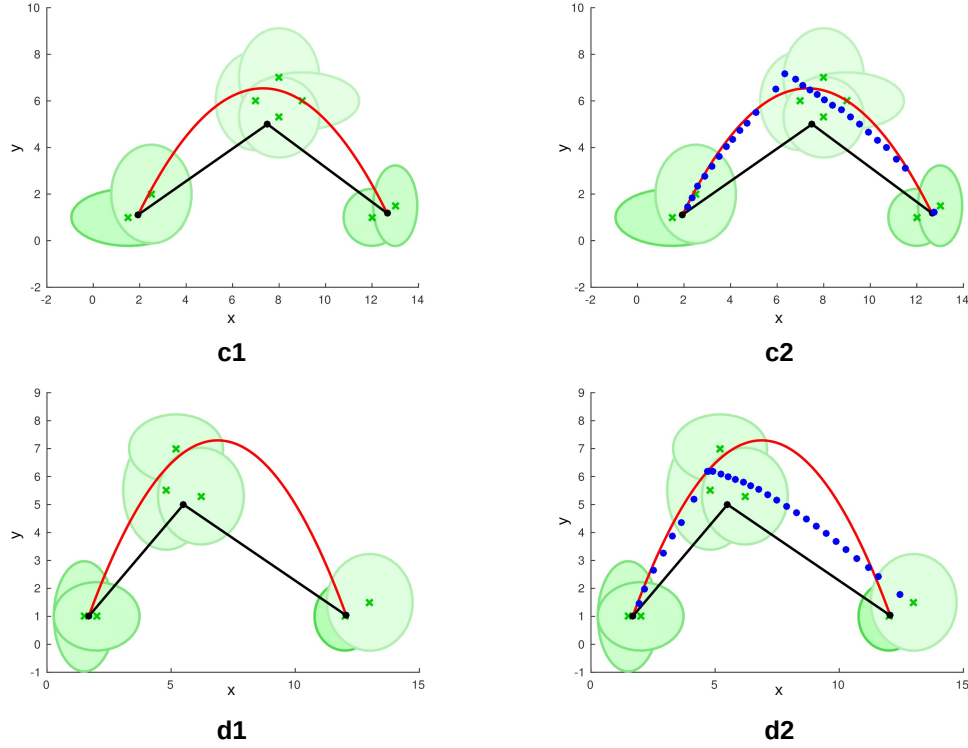


Figure 5.3: Experimental result 2 with synthetic dataset using the proposed method in Chapter 5. We use a red line to represent a reference force path, and we use a black line to represent an initial path where waypoints of the initial path are indicated by black circles. The GMMs are shown in green, where the transparency of each Gaussian model depends on its mixture probability. The larger the mixture probability is, the darker the colour of the Gaussian model is. The centre of each Gaussian model is marked with a green cross. The waypoints of the final updated path are indicated with blue circles. (c1) For the 3rd dataset, the first, second, and third joints are modelled with 2 GMMs, 4 GMMs, and 2 GMMs separately. Since both the initial path and the reference force path mostly remain in the GMMs area, the final updated path in (c2) is close to the reference force path only with a little compromise between the reachability and the comfort criteria. It shows that as long as the reference force path mostly remains in the GMMs area, the detected force information which is simulated by the Euclidean distance becomes the main factor in the iterative process. (d1) For the 4th dataset, the first, second, and third joints are modelled with 2 GMMs, 3 GMMs, and 2 GMMs separately. The left part of the reference force path mostly remains in the GMMs area while the right part rarely remains. Thus in (d2), the left part of the final updated path is close to the reference force path, where the detected force information is the main factor which influences the path update. The right part of the final updated path is getting closer to the reference force path while trying to stay inside the GMMs area, which is a compromise between the reachability and the comfort criteria.

Experimental results with the synthetic dataset in Figure 5.2 and 5.3 show that the final updated path can achieve a balance between the reachability and the comfort criteria when they are contradicted. When the reachability criterion is satisfied, the comfort criterion becomes the main criterion which influences the path update. When the reachability criterion is not satisfied, the proposed method can find a compromise between the two criteria.

Figure 5.4, 5.5, 5.6, and 5.7 show the comparison results with 4 sets of synthetic data among the proposed methods in Chapter 3, 4, and 5. Figure 5.4(a1), 5.5(b1), 5.6(c1), and 5.7(d1) show the reference force paths and the initial paths with the GMMs. We use a red line to represent a reference force path, and we use a black line to represent an initial path where waypoints of the initial path are indicated by black circles. The GMMs are shown in green where the transparency of each Gaussian model depends on its mixture probability. The larger the mixture probability is, the darker the colour of the Gaussian model is. The centre of each Gaussian model is marked with a green cross.

Figure 5.4(a2), 5.5(b2), 5.6(c2), and 5.7(d2) show the final paths in blue with the proposed method in Chapter 3. The paths are in blue dashed lines where blue circles indicate the waypoints of the paths. It can be seen that the final paths are the same with the initial paths. This is because no real-time feedback information is taken into consideration during path planning. After an initial path is chosen, there is no other factor which can change or update the path.

Figure 5.4(a3), 5.5(b3), 5.6(c3), and 5.7(d3) show the final paths with the proposed method in Chapter 4. The waypoints of the final paths are indicated with blue circles. It can be seen that the final paths are close to the reference force paths. This is because the detected force information which is simulated by the Euclidean distance is the main and only factor which affects the whole iterative path update process.

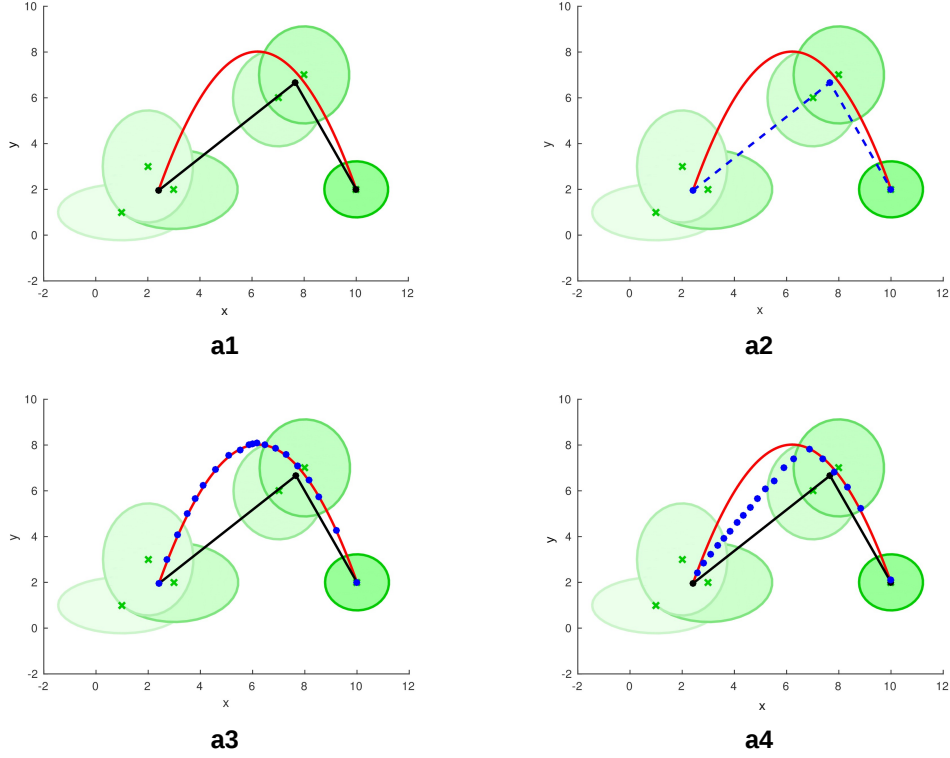


Figure 5.4: Comparison result 1 with synthetic dataset among the proposed methods in Chapter 3, 4, and 5. (a1) shows the reference force path and the initial path with the GMMs. We use a red line to represent the reference force path, and we use a black line to represent the initial path where waypoints of the initial path are indicated by the black circles. The GMMs are shown in green, where the transparency of each Gaussian model depends on its mixture probability. The larger the mixture probability is, the darker the colour of the Gaussian model is. The centre of each Gaussian model is marked with a green cross. (a2) shows the final path in blue with the proposed method in Chapter 3. The path is in the blue dashed line, where blue circles indicate the waypoints of the path. It can be seen that the final path is the same with the initial path. This is because no real-time feedback information is taken into consideration during path planning. After an initial path is chosen, there is no other factor which can change or update the path. (a3) shows the final path with the proposed method in Chapter 4. The waypoints of the final path are indicated by the blue circles. It can be seen that the final path is close to the reference force path. This is because the detected force information which is simulated by the Euclidean distance is the main and only factor which affects the whole iterative path update process. (a4) shows the final path with the proposed method in Chapter 5. The waypoints of the final path are indicated by the blue circles. The left part of the final path gets closer to the reference force path while trying to remain in the GMMs area. The right part of the final path is close to the reference force path. It can be seen that the final path achieves a balance between the reachability and the comfort criteria.

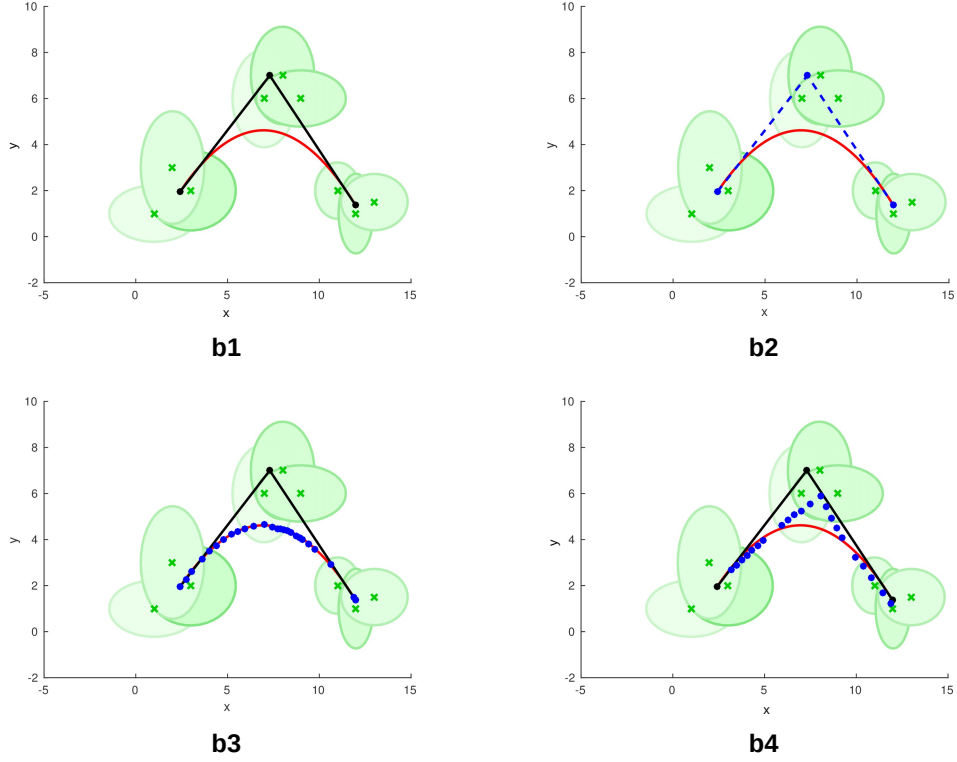


Figure 5.5: Comparison result 2 with synthetic dataset among the proposed methods in Chapter 3, 4, and 5. (b1) shows the reference force path and the initial path with the GMMs. We use a red line to represent the reference force path, and we use a black line to represent the initial path where waypoints of the initial path are indicated by the black circles. The GMMs are shown in green, where the transparency of each Gaussian model depends on its mixture probability. The larger the mixture probability is, the darker the colour of the Gaussian model is. The centre of each Gaussian model is marked with a green cross. (b2) shows the final path in blue with the proposed method in Chapter 3. The path is in the blue dashed line, where blue circles indicate the waypoints of the path. It can be seen that the final path is the same with the initial path. This is because no real-time feedback information is taken into consideration during path planning. After an initial path is chosen, there is no other factor which can change or update the path. (b3) shows the final path with the proposed method in Chapter 4. The waypoints of the final path are indicated by the blue circles. It can be seen that the final path is close to the reference force path. This is because the detected force information which is simulated by the Euclidean distance is the main and only factor which affects the whole iterative path update process. (b4) shows the final path with the proposed method in Chapter 5. The waypoints of the final path are indicated by the blue circles. The starting and ending part of the final path is close the reference force path while the middle part makes a compromise between the initial path and the reference force path. It can be seen that the final path achieves a balance between the reachability and the comfort criteria.

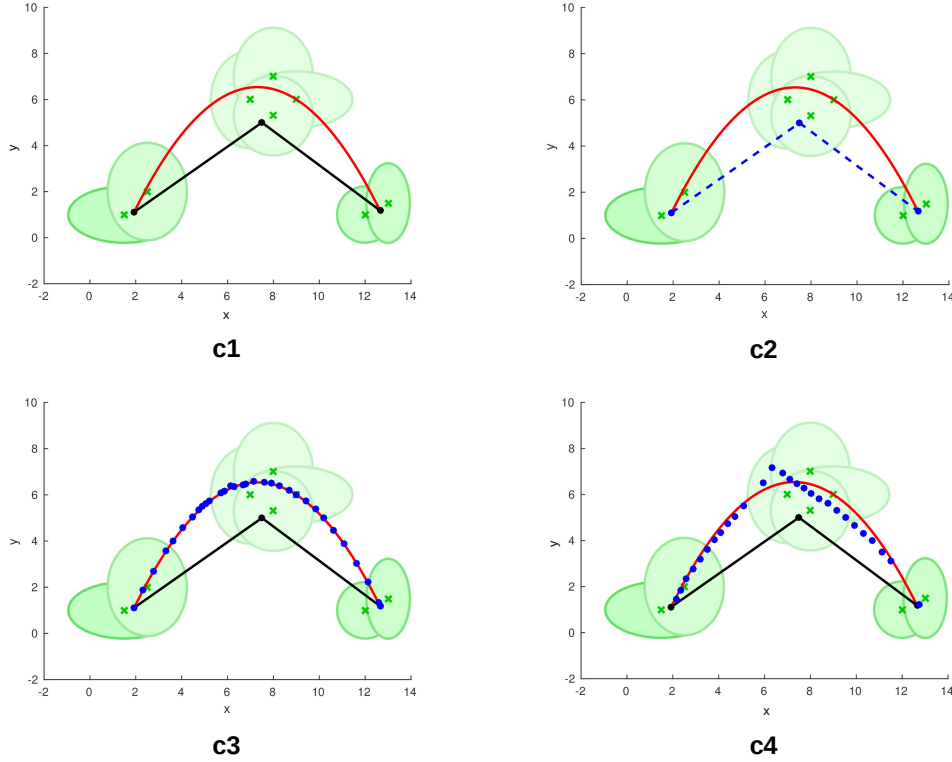


Figure 5.6: Comparison result 3 with synthetic dataset among the proposed methods in Chapter 3, 4, and 5. (c1) shows the reference force path and the initial path with the GMMs. We use a red line to represent the reference force path, and we use a black line to represent the initial path where waypoints of the initial path are indicated by the black circles. The GMMs are shown in green, where the transparency of each Gaussian model depends on its mixture probability. The larger the mixture probability is, the darker the colour of the Gaussian model is. The centre of each Gaussian model is marked with a green cross. (c2) shows the final path in blue with the proposed method in Chapter 3. The path is in the blue dashed line, where blue circles indicate the waypoints of the path. It can be seen that the final path is the same with the initial path. This is because no real-time feedback information is taken into consideration during path planning. After an initial path is chosen, there is no other factor which can change or update the path. (c3) shows the final path with the proposed method in Chapter 4. The waypoints of the final path are indicated by the blue circles. It can be seen that the final path is close to the reference force path. This is because the detected force information which is simulated by the Euclidean distance is the main and only factor which affects the whole iterative path update process. (c4) shows the final path with the proposed method in Chapter 5. The waypoints of the final path are indicated by the blue circles. The final path is quite close to the reference force path. It can be seen that when the reachability criterion is satisfied, the comfort criterion becomes the main criterion which influences the path update.

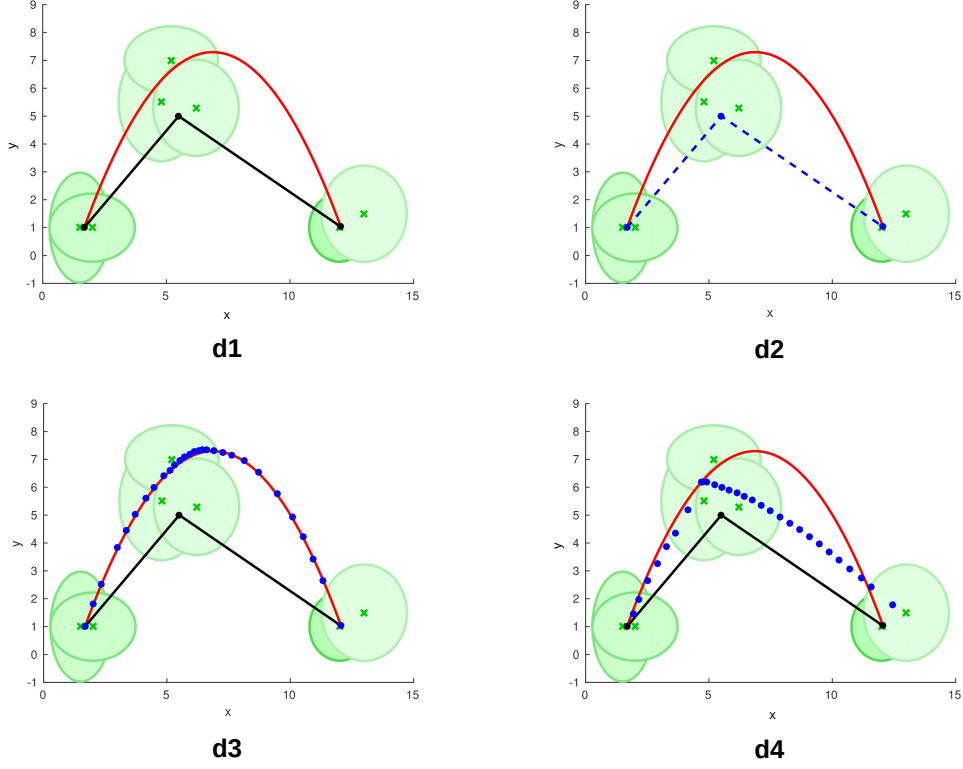


Figure 5.7: Comparison result 4 with synthetic dataset among the proposed methods in Chapter 3, 4, and 5. (d1) shows the reference force path and the initial path with the GMMs. We use a red line to represent the reference force path, and we use a black line to represent the initial path where waypoints of the initial path are indicated by the black circles. The GMMs are shown in green, where the transparency of each Gaussian model depends on its mixture probability. The larger the mixture probability is, the darker the colour of the Gaussian model is. The centre of each Gaussian model is marked with a green cross. (d2) shows the final path in blue with the proposed method in Chapter 3. The path is in the blue dashed line, where blue circles indicate the waypoints of the path. It can be seen that the final path is the same with the initial path. This is because no real-time feedback information is taken into consideration during path planning. After an initial path is chosen, there is no other factor which can change or update the path. (d3) shows the final path with the proposed method in Chapter 4. The waypoints of the final path are indicated by the blue circles. It can be seen that the final path is close to the reference force path. This is because the detected force information which is simulated by the Euclidean distance is the main and only factor which affects the whole iterative path update process. (d4) shows the final path with the proposed method in Chapter 5. The waypoints of the final path are indicated by the blue circles. The left part of the final path is close to the reference force path. The right part of the final path gets closer to the reference force path while trying to remain in the GMMs area. It can be seen that the final path achieves a balance between the reachability and the comfort criteria.

Figure 5.4(a4), 5.5(b4), 5.6(c4), and 5.7(d4) show the final paths with the proposed method in Chapter 5. The waypoints of the final paths are indicated with blue circles. It can be seen that the final paths achieve a balance between the reachability and the comfort criteria. More reasoning has been presented in Figure 5.2 and 5.3.

5.2.2 Real-world Personalised Assistive Dressing

We evaluated the proposed method with 8 healthy participants (five female) ages 24-30 (mean:26.63, std:1.85) by running comparison experiments. For each participant, we ran 3 experiments to enable the Baxter robot to provide personalised dressing assistance following the user modelling methods in Chapter 3, 4, and 5. The maximum iteration number $t_{\max}$ was set to 8 and the energy threshold τ_{energy} was set to 0N.

Each user was told to pretend with one of the following arm movement limitations: (1) pretend to have more movement flexibility in the left and right directions for both arms (2 users), (2) pretend to have more movement flexibility in the forward and backwards directions for both arms (3 users), (3) pretend to have more movement flexibility in the right arm and less movement flexibility in the left arm (3 users). The motion data of each user's arms was first recorded and modelled with GMMs, then the robot's dressing motion for each user was planned separately using the user modelling methods in Chapter 3, 4, and 5. During assistive dressing, each user still tried to control the arm motion following the given movement limitations.

For each user, the initial dressing path was determined according to the GMMs of the human joints, where the starting position of each joint was the one which maximised its probability given by the corresponding GMMs. As we mentioned before, the joint's position which maximises its Gaussian mixture probability can be viewed as the most frequent visiting position for this

joint with high comfort. According to the recorded motion data, the length of the forearm and upper arm can be calculated for each user. For each arm of a user, the initial positions of the hand, elbow, and shoulder should follow the kinematic structure of the human arm. However, since the GMMs of the joints on the same arm were modelled independently, the initial positions chosen for the joints may sometimes not fulfil this requirement. In the experiments with the synthetic dataset, we introduced some randomness when choosing the initial position for the middle point. Thus if it was found that for the same human arm, the Euclidean distances between the initial positions of the hand and elbow, between the initial positions of the elbow and shoulder had more than 1.5cm difference with the length of the forearm and upper arm, the initial position for the human elbow was then locally adjusted by choosing the closest position with a relatively larger probability given the GMMs. The maximum difference we found in the experiments is 2cm.

Three representatives of the experimental results are shown in Figure 5.8, 5.9, and 5.10. All the data is in the robot's coordinates frame. For each user's left arm, the GMMs of hand, elbow, and shoulder are drawn in black, green, and cyan. For each user's right arm, the GMMs of hand, elbow, and shoulder are drawn in red, blue, and yellow. An initial dressing path is represented by a black dashed line connecting the black dots. A final updated path is represented by a blue line connecting the blue dots.

Figure 5.8 shows comparison result 1 with real assistive dressing data, where the user pretended to have more movement flexibility in the left and right directions for both arms. It can be seen from Figure 5.8(a1) that the user's left hand, elbow, and shoulder are modelled with 3 GMMs, 2 GMMs, and 1 GMM while the right hand, elbow, and shoulder are also modelled with 3 GMMs, 2 GMMs, and 1 GMM. With the user modelling method in Chapter 3, the initial dressing path also becomes the final dressing path. This is because feedback information which can affect the dressing path

is not taken into consideration. Figure 5.8(a2) and 5.8(a3) show the final updated paths in blue with the user modelling methods in Chapter 4 and 5 respectively. Although the paths in Figure 5.8(a2) and 5.8(a3) look roughly similar, the final path in Figure 5.8(a3) is closer to the initial path with more path waypoints inside the GMMs area of the upper-body joints comparing with the final path in Figure 5.8(a2). This is because if we try to fit a stick model to each path waypoint in Figure 5.8(a3), the starting and ending points of the stick should be within the GMMs area of the upper-body joints with a higher joint Gaussian probability of the two endpoints.

Figure 5.9 shows comparison result 2 with real assistive dressing data, where the user pretended to have more movement flexibility in the forward and backwards directions for both arms. It can be seen from Figure 5.9(b1) that the user's left hand, elbow, and shoulder are modelled with 3 GMMs, 2 GMMs, and 1 GMM while the right hand, elbow, and shoulder are also modelled with 3 GMMs, 2 GMMs, and 1 GMM. Figure 5.9(b2) and 5.9(b3) show the final updated paths in blue with the user modelling methods in Chapter 4 and 5 respectively. For both arms, the final paths from hands to elbows almost look the same in Figure 5.9(b2) and 5.9(b3). This is because the user's arms have more movement flexibility in forward and backwards directions. If the final updated path using force information mainly locates within the GMMs area of the upper-body joints, the final updated path using both vision and force information would look similar. As we discussed before with the synthetic dataset, when there is no contradiction between the reachability and the comfort criteria, the comfort criterion becomes the main criterion which influences the path update. For both arms, the final paths from elbows to shoulders in Figure 5.9(b3) are closer to the initial paths comparing with the paths in Figure 5.9(b2).

Figure 5.10 shows comparison result 3 with real assistive dressing data, where the user pretended to have more movement flexibility in the right

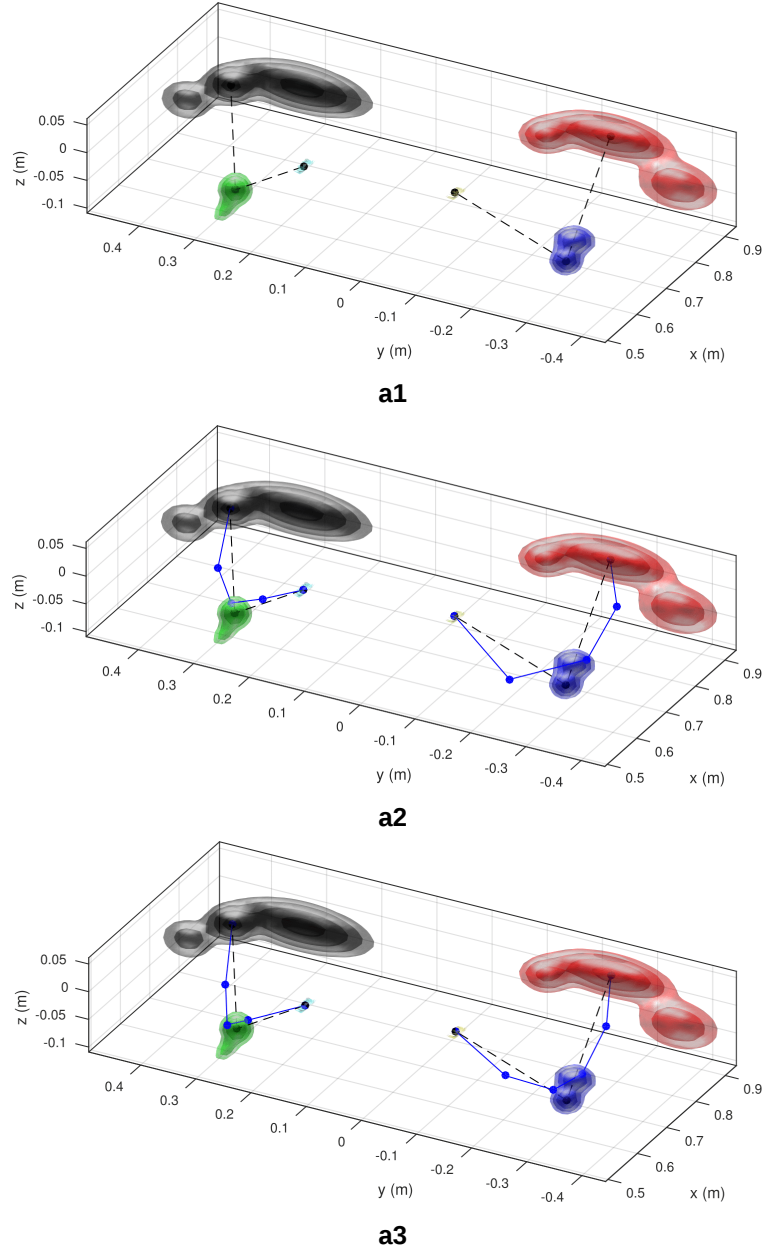


Figure 5.8: Comparison result 1 with real assistive dressing data among the proposed user modelling methods in Chapter 3 (a1), 4 (a2), and 5 (a3). The GMMs of the user's left hand, elbow, and shoulder are drawn in black, green, and cyan. The GMMs of the user's right hand, elbow, and shoulder are drawn in red, blue, and yellow. The dressing path in (a1) is the initial dressing path in (a2) and (a3), which is shown by the black dashed line where path waypoints are indicated with the black circles. The waypoints of the final updated path in (a2) and (a3) are indicated with the blue circles connected by the blue lines. (a1) The initial dressing path also becomes the final dressing path. This is because feedback information which can affect the dressing path is not taken into consideration. The final path in (a3) is closer to the initial path with more path waypoints inside the GMMs area of the upper-body joints comparing with the final path in (a2). This is because if we try to fit a stick model to each path waypoint in (a3), the starting and ending points of the stick should be within the GMMs area of the upper-body joints with a higher joint Gaussian probability of the two endpoints.

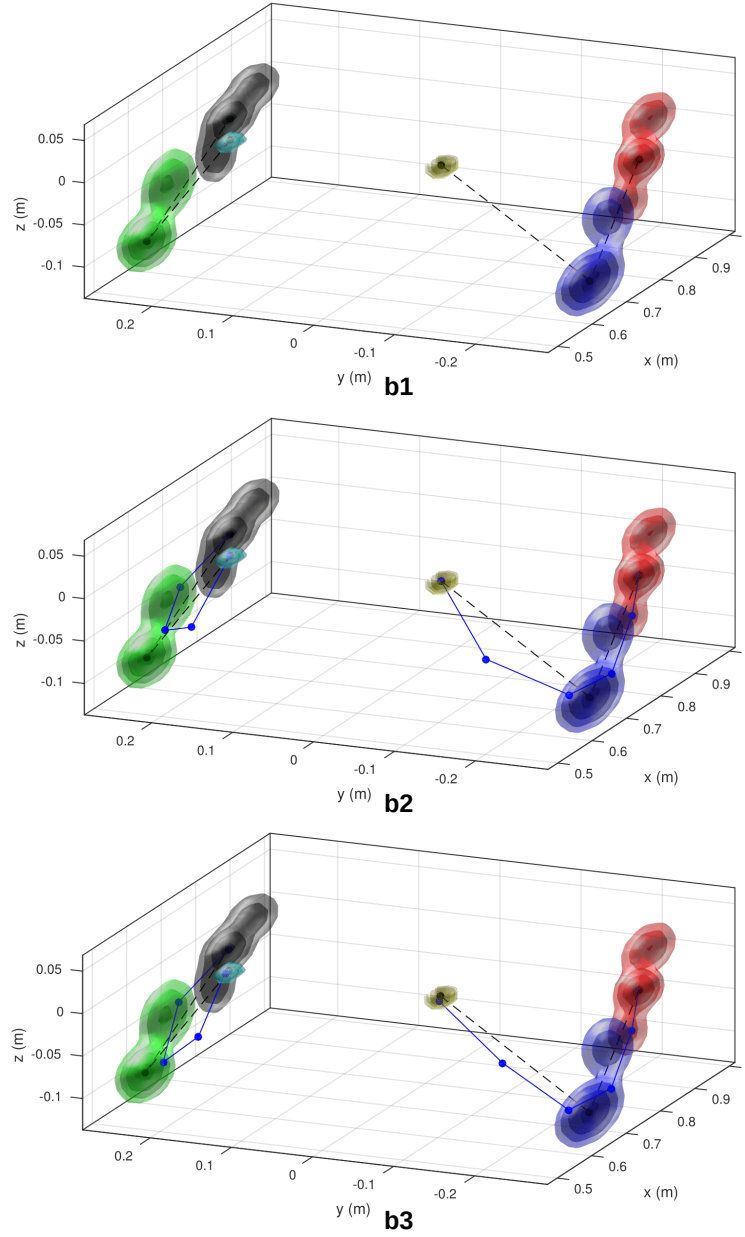


Figure 5.9: Comparison result 2 with real assistive dressing data among the proposed user modelling methods in Chapter 3 (b1), 4 (b2), and 5 (b3). The GMMs of the user's left hand, elbow, and shoulder are drawn in black, green, and cyan. The GMMs of the user's right hand, elbow, and shoulder are drawn in red, blue, and yellow. The dressing path in (b1) is the initial dressing path in (b2) and (b3), which is shown by the black dashed line where path waypoints are indicated with the black circles. The waypoints of the final updated path in (b2) and (b3) are indicated with the blue circles connected by the blue lines. (b1) The initial dressing path also becomes the final dressing path. This is because feedback information which can affect the dressing path is not taken into consideration. For both arms, the final paths from hands to elbows almost look the same in (b2) and (b3). This is because the user's arms have more movement flexibility in forward and backward directions. If the final updated path using force information mainly locates within the GMMs area of the upper-body joints, the final updated path using both vision and force information would look similar.

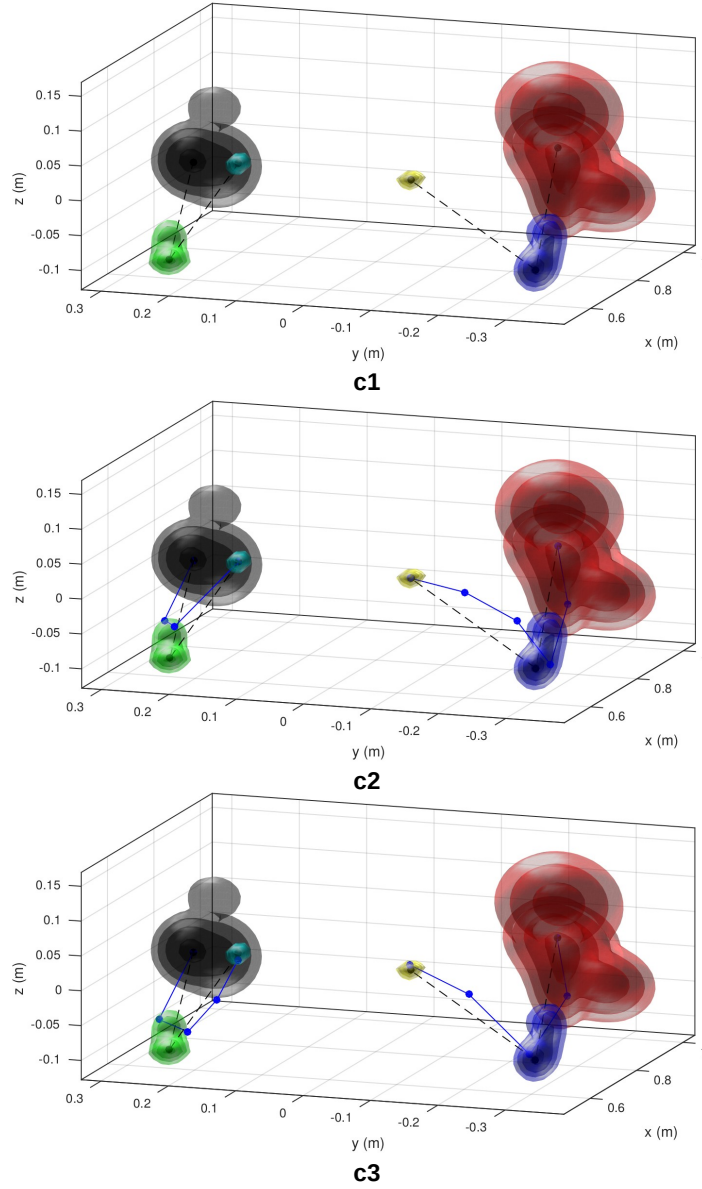


Figure 5.10: Comparison result 3 with real assistive dressing data among the proposed user modelling methods in Chapter 3 (c1), 4 (c2), and 5 (c3). The GMMs of the user's left hand, elbow, and shoulder are drawn in black, green, and cyan. The GMMs of the user's right hand, elbow, and shoulder are drawn in red, blue, and yellow. The dressing path in (c1) is the initial dressing path in (c2) and (c3), which is shown by the black dashed line where path waypoints are indicated with the black circles. The waypoints of the final updated path in (c2) and (c3) are indicated with the blue circles connected by the blue lines. (c1) The initial dressing path also becomes the final dressing path. This is because feedback information which can affect the dressing path is not taken into consideration. For the left arm, the final path in (c3) has more path waypoints inside the upper-body joints GMMs area comparing with the final path in (c2). For the right arm, the final path from hand to elbow in (c3) looks similar to the final path in (c2). This is because the comfort criterion mainly determines the path update when both the reachability and comfort criteria can be fulfilled. For the right arm, the final path from elbow to hand in (c3) is closer to the initial path comparing with the path in (c2).

arm and less movement flexibility in the left arm. It can be seen from Figure 5.10(c1) that the user's left hand, elbow, and shoulder are modelled with 3 GMMs, 2 GMMs, and 1 GMM while the right hand, elbow, and shoulder are also modelled with 4 GMMs, 4 GMMs, and 1 GMM. Figure 5.10(c2) and 5.10(c3) show the final updated paths in blue with the user modelling methods in Chapter 4 and 5 respectively. For the left arm, the final path in Figure 5.10(c3) has more path waypoints inside the upper-body joints' GMMs area compared with the final path in Figure 5.10(c2). For the right arm, the final path from hand to elbow in Figure 5.10(c3) looks similar to the final path in Figure 5.10(c2). This is because the comfort criterion mainly determines the path update when both the reachability and comfort criteria can be fulfilled. For the right arm, the final path from elbow to hand in Figure 5.10(c3) is closer to the initial path comparing with the path in Figure 5.10(c2).

Experiments with real-world assistive dressing data showed that the proposed user modelling method using vision and force information in this chapter can enable the robot to search for an optimal personalised dressing path for a user by achieving a balance between the reachability and the comfort criteria.

5.3 CONCLUSION

In this chapter, we have presented a user modelling method using vision and force information. We applied the proposed method to enable the Baxter robot to search for the optimal personalised dressing path for a human user while trying to fulfil both the reachability and the comfort criteria. We proposed an online iterative path optimisation method using multi-modal information, by combining the GMMs of the human upper-body joints with the online iterative path optimisation in Chapter 4. We introduced a stick

model to model the body part which connected two upper-body joints of the same human arm, where the human upper-body joints were modelled with GMMs. The main difference between the proposed method in this chapter and Chapter 4 is that we update the next goal position of the path by maximising the joint Gaussian probability of the two endpoints of a stick model instead of following the same update rule for the current position.

We ran experiments on both the synthetic dataset and the real-world assistive dressing data by comparing the proposed method with the other two user modelling methods in Chapter 3 and 4. Experimental results with both synthetic dataset and real-world data have shown that the final updated path can achieve a balance between the reachability and the comfort criteria when they are contradicted. When the reachability criterion is satisfied, the comfort criterion becomes the main criterion which influences the path update. When the reachability criterion is not satisfied, the proposed method can find a compromise between the two criteria.

CONCLUSIONS AND FUTURE WORK

In this chapter, we first summarise the main contributions of this thesis, followed by discussing the limitations in our work. Finally, we discuss about the possible future work as an extension to this thesis.

6.1 SUMMARY OF CONTRIBUTIONS

Everyone in the world is different from each other, from personalities, psychological and physiological characteristics to behaviours and skill sets. For assistive robots to better assist human users, we believe that personalisation plays a significant role, where assistive robots should be able to learn the human preferences and utilise this knowledge in human-robot interactions. The main contribution of this thesis is a user modelling method using multi-modal information. We apply the proposed method in solving a challenging real-world application *personalised assistive dressing by humanoid robots*.

In Chapter 3, we first proposed a method to model the movement space of the human upper-body joints using GMMs. A top-view depth sensor was mounted on top of the robot's face screen and we recognised the human upper-body pose from a single depth image in real-time using randomised decision forests. Through collecting the human motion information, the movement space of each human upper-body joint was modelled with GMMs so that we can learn the reachable area of each upper-body joint. We enabled the Baxter humanoid robot to provide personalised dressing assistance according to the GMMs of the human upper-body joints and real-time

upper-body pose estimation. We realised personalisation in assistive dressing mainly based on the reachability criterion.

Since vision occlusions could occur and no real-time feedback information was taken into consideration, we proposed to use real-time force sensor information for the robot to locally adjust its motion in Chapter 4. Detected force information can reflect if a human user feels comfortable with the robot's motion. Apart from this, we believe that a human user tends to follow certain behaviour patterns in daily activities. Thus we proposed an online iterative path optimisation method based on adaptive moment estimation and enabled the robot to search for the optimal personalised dressing path for a human user. We realised personalisation in assistive dressing mainly based on the comfort criterion.

To enable personalisation in assistive dressing based on both the reachability and the comfort criteria, in Chapter 5 we proposed an online iterative path optimisation method using multi-modal information by combining the GMMs of the human upper-body joints with the online iterative path optimisation process. Experimental results on both synthetic dataset and real-world assistive dressing data show that when there is no contradiction between the two criteria, the comfort criterion becomes the main criterion which affects the path optimisation, otherwise the proposed method can achieve a balance between the two criteria.

6.2 LIMITATIONS

Currently, we model the movement space of the human upper-body joints using GMMs, where each upper-body joint is modelled independently. However, the movement of each human upper-body joint is closely related to the movement of other joints on the same human arm. For instance, the movement of a human hand affects the movement of the elbow and shoulder on

the same arm, and the movement of a human elbow also has an effect on the placement of the hand and shoulder on the same arm. We try to compensate for this disadvantage by introducing a stick model in Chapter 5, where a stick model is used to model the body part connected by two arm joints. With the stick model, we can infer the positions of the two endpoints and then calculate the joint Gaussian probability of the two endpoints. Within a searching range, we select the endpoint position which maximises the joint probability. However, a stick model can only model the movement relationships of two arm joints. A human arm, which consists of three arm joints, moves as a whole. Thus a stick model is still not enough to describe the movement relationships among the joints of the hand, elbow, and shoulder on the same arm.

To recognise the human upper-body pose with a top-view depth sensor which can be attached and integrated with the Baxter robot, we use randomised decision forests to classify every pixel to a body part from a single depth image in Chapter 3. We define 4 body joints for each arm, which are hand, forearm, upper arm, and shoulder. We extract the 3D positions of each arm joints by calculating the means of each defined upper-body joint area. It can be seen that the human forearm locates between the hand and elbow, and the human upper arm locates between the elbow and shoulder. In Chapter 3, the dressing order for a human arm is hand, forearm, upper arm, and shoulder. However, the elbow position which cannot be directly inferred from the positions of the forearm and upper arm is actually ignored. We can see that if a human arm stays still, the best dressing order should be hand, elbow, and shoulder. In Chapter 4, we use a front-view depth sensor instead to recognise human upper-body pose with the OpenNI skeleton tracker, which can directly provide the positions of hands, elbows, and shoulders. The advantage of utilising a top-view depth sensor is that the camera can be mounted on top of the robot's face screen. If the robot can move during assistive dress-

ing, the camera can directly move along with the robot. However, the Baxter robot we currently have in the lab has no moving platform. Besides, using a top-view depth sensor requires collecting a large amount of image training data for each user, which is time costly. Thus we make use of a front-view depth sensor to recognise human upper-body pose and transform the human upper-body pose from the camera coordinates to the robot coordinates.

However, no matter with the randomised decision forests for a top-view camera or the OpenNI skeleton tracker for a front-view camera, severe occlusions could occur during assistive dressing, especially when the robot's arms, the human body, and clothes are in close contact. There have been computer vision methods on human pose recognition that can deal with self-occlusions. For instance, Girshick *et al.* proposed to use Hough forests to recognise human pose from a single depth image (Girshick *et al.*, 2011). In the training process, training data must include a large amount of self-occluded images with ground-truth body joint labels. In the testing process, each pixel in the depth image would vote for a 3D position for each body joint and the final absolute 3D body joint positions are extracted using mean shift. Thus even if some body parts are occluded by other body parts, the occluded body joints positions can still be inferred. However, this method cannot solve the occlusion problem in our assistive dressing. First, not only self-occlusions but also occlusions from the robot's arms and clothes make the situations more complicated. Second, ground-truth body joints are hard to generate for training images with various occlusions. There is no doubt that being able to keep recognising human upper-body pose during assistive dressing is beneficial for the robot to adjust its motion. Besides, the robot would also benefit from knowing the spatial relationships between human body and the clothes so that the robot can be clear with the progress of dressing.

To detect external force resistance during assistive dressing, we use a fixed-size moving window to calculate the sum of force difference. We make use of the detected force information from the endpoints of robot's limbs. However, there are no force sensors attached on the robot's grippers, where the robot uses grippers to grasp the shoulder parts of a sleeveless jacket. Although we try to minimise the detected force noise by calculating the sum of force difference within a sliding window, some environment noise still cannot be avoided. If the robot's grippers are attached with force or tactile sensors, detected external force resistance would have less noise and the robot can react more sensitively to the disturbance. Besides, we use position control to control the robot's arm motion, where we send goal positions command to the robot and the robot's arms motion is planned using the motion planning library. However, force control or torque control would be a more natural way to control robot motion.

In Chapter 4, we proposed an online iterative path optimisation method to enable the robot to search for the optimal personalised dressing path. In the experiment, the initial dressing position for a user's hand was decided and chosen by the robot according to the GMMs of the user's hand. At the beginning of each iteration, the robot placed the clothes to this initial position. Thus the final optimised dressing path is with respect to a fixed initial position of the user. As we have discussed before, if we let the user choose the initial dressing position in each iteration, then it might be difficult for the user to place the hand in the same position every time. If the user started with different initial hand's positions each time, then the iteration may never end without setting a maximum iteration number. With fixed initial dressing position, our method can discover the preferred movement pattern/path of the user's arm. Given different initial starting positions, our method can still work, as long as the initial starting positions in each iteration are the same. However, it can be seen that experiments need to be re-run for dif-

ferent initial starting positions. Although the proposed method finds the optimal personalised dressing path with respect to the current fixed initial dressing position, it cannot directly generalise to situations with a different fixed initial dressing position.

We define 2 main criteria to evaluate the performance of the proposed methods in the thesis, which are the reachability criterion and the comfort criterion. For the definition of the comfort criterion, we currently focus more on the engineering perspective. Thus it is defined as there is no external force resistance detected during assistive dressing. However, from a broader view, the comfort criterion should also mean the user feels comfortable and satisfied. In order to evaluate this, sensors which can detect the user's heart rates, mood, or facial expression should be introduced. Questionnaires should be filled by users in order to collect their subjective opinions on the assistive dressing system. All the discussed information above or even more should be taken into consideration when deciding a complete definition of the comfort criterion.

Although the ultimate goal of assistive dressing is to assist real disabled people with their dressing, we need to solve the engineering problems first by building a stable and robust assistive dressing system, which itself is full of challenges. That's why all the experiments in this thesis were conducted with healthy participants in order to evaluate the technical part of the dressing system. In the experiments in Chapter 3 and 5, we instructed the participants to pretend with different kinds of movement limitations during both data collection for the upper-body movement space modelling and assistive dressing by humanoid robots. Although we instructed the participants to keep consistency in their behaviours, we had no measure of how compliant they actually were. Besides, since there are still some engineering problems to be solved in the future, we did not design and conduct user studies to receive their subjective feedback on the dressing system.

As we mentioned earlier in the thesis, there are 3 main components in assistive dressing, which are human users, assistive robots and clothes. However, there has been no prior research which took all the 3 components into consideration. Some prior research studied robots manipulating with various kinds of clothes by folding or unfolding them. Some research work studied how to enable robots to learn the motor skills of dressing and experiments in these work were usually conducted with human mannequins instead of real human users. Although there has been research work on assistive dressing from different perspectives, there is much less prior research which enabled robots to dress real human users. Human mannequins can stay still during experiments. With this assumption, some research work studied the dynamics of the clothes during assistive dressing. However, real human users are much more complicated and the dynamics of the clothes can be affected not only by the behaviours/poses of human users but also the type of the clothes human users wear. To enable assistive robots to dress real human users, sensor information about users' states should be collected. That's why we use both vision and force sensor information to build user models to enable robots to provide dressing assistance to real human users. However, we used a sleeveless jacket in the dressing experiments and the clothes state was not considered in this work.

6.3 FUTURE WORK

To improve the movement space modelling of the human upper-body joints, future work could study the kinematics of the human arms and model the movement relationships among the human upper-body joints using machine learning algorithms, such as Gaussian Process.

Since vision information provides a significant clue during human-robot interactions, it would be effective to continuously check the spatial relation-

ships among the robot arms, the human body, and the clothes. Future work could study a robust vision algorithm to recognise human upper-body pose under severe occlusions caused by the robot's arms and clothes.

To control the dressing motion of the assistive robot, future work could study force control or torque control to make the robot motion more natural. If we can directly control the applied force on the endpoints of robot limbs, detecting external resistance by reading the force value of robot's grippers would be more accurate and smooth.

Robots manipulating with soft materials is another popular research topic. In this thesis, we fix the orientations of robot's grippers during assistive dressing and each of the robot's arms is responsible for dressing one of the user's arms. However, in order to enable assistive robots to assist real human users with more complicated clothes, it brings more challenges in the collaboration of the robot's arms. For instance, if the robot assists a user to dress with a jacket with sleeves and the user's arm gets stuck in one of the sleeves, then the robot might need to use both arms to manipulate with one sleeve of the jacket. Thus future work could study how to enable the robot to manipulate with the clothes to provide more flexible dressing motion.

The factor of time is not considered in assistive dressing, including the dynamics of human motion and clothes. Future work could incorporate time into assistive dressing and enable the robot to assist human users to dress with more complicated clothes, such as a jacket with sleeves. Future work could also study how to enable the robot to undress a human user.

The proposed iterative path optimisation using multi-modal information can be extended with other machine learning algorithms, such as reinforcement learning and deep learning. Future work could also include new applications with different robotic platforms, such as iCub and NAO.

During assistive dressing, different goal positions are sent to the robot and the robot plans its motion using the motion planning library. Future work

could study how to teach the robot to assist humans with their dressing. A human teacher can show some demonstrations on how to dress different users and the robot can learn the dressing motion by modelling the human teacher's behaviours.

If the assistive robot has a moving platform, the robot will have more flexibility when dressing the user. For instance, even for a very tall user with longer arms, the robot will still be able to dress the user by moving to the front left side or front right side. Then it will be another challenging problem on how to control the movement of the robot platform.

To enable assistive robots to dress real human users in a jacket with sleeves, human dynamics can be learned to guide the robot to plan its dressing motion in order to deal with vision occlusions. However, learning human dynamics itself faces challenges in vision. Future work can use the position, velocity, or torque information of the robot's end effector or all the robot's arms' joints to infer the human pose at each time step. Besides, a heat sensor can be introduced to distinguish the human body from the robot and the clothes in order to estimate the human motion. The inferred human upper-body pose at each time step during assistive dressing can then be used to learn the motion dynamics of the human upper body. The state of the clothes is another significant factor that needs to be considered in order to enable assistive robots to assist users to wear different kinds of clothes. For instance, if the user's arm gets stuck in one of the sleeves of a jacket, then the robot should detect the current shape/configuration of the clothes and decide how to adjust the configuration of the clothes to continue the dressing process. It is challenging because the robot should not only detect which part of the clothes causes the problem but also infer the configuration of the user's arm which might be partly or completely hidden by the clothes.

Future work can conduct some user studies. Before designing the assistive dressing system, researchers can use questionnaires to collect the following

information from disabled people or people with mobility limitations, such as what they think is the difficult part during dressing, what they expect most from assistive dressing robots, what their attitudes towards such dressing robots. Based on the survey, attentions can be paid to solve the above concerns from human users when designing the assistive dressing system. When evaluating the system, questionnaires can also be designed based on the previous survey to check whether the proposed system provides dressing assistance in the way that users prefer. The current assistive dressing system will benefit from the feedback from real users in order to improve its performance.

Apart from assistive dressing, the idea of using vision and force sensor information for personalisation in this thesis can be applied to other research fields, such as rehabilitation robotics and medical robotics. For people who suffer from stroke and require rehabilitation exercise, a probabilistic representation of their upper-body movement space can be learned first with the GMMs method so that experts can design the personalised rehabilitation tasks for different users based on their reachable areas. Rehabilitation tasks usually require users to move with a rehabilitation robotic arm following certain paths. According to the force sensor feedback from the robotic arm, the robot can either change the compliance of its own motion or guide the user to move towards another area. When considering both vision and force information together for such rehabilitation robotic arms, it can provide a better understanding of the movement area of the human arms. For instance, a position which is frequently visited by a user's hand may have a high probability given the GMMs. However, if it is found that the force sensor feedback for this position is also high, it actually means that this position might be originally difficult for the user to reach and the user tries to practice more to reach this position.

APPENDIX

A.1 ROBOT PLATFORM

The experiments in this thesis are conducted using a Baxter research robot developed under rethink robotics. Baxter is a humanoid robot, which is approximately 1.8 meters in height. It has two 7 degree-of-freedom arms and various sensors including force, position, and torque sensing for each joint. There are two RGB cameras at the endpoints of two limbs. The robot arms are also equipped with infrared range sensors and accelerometers. There is another RGB camera on the robot's head which is a 1024×600 SVGA LCD screen.

The robot arms are actuated with Series Elastic Actuators (SEAs) where a spring is inserted between the motor and the load, which results in natural compliance and inherent safety for the robot. Apart from this, the robot has an emergency stop button which can stop the robot's motion during any execution.

The robot supports the following control modes, which are joint position, joint velocity, and joint torque control modes. The robot safety system also has collision avoidance to prevent self-collision and to detect collision with the environment. Humans can enable the Zero-G mode by grasping the cuff over the robot arm's groove and then move the end effector of the robot arm to the desired position.



Figure A.1: The Baxter humanoid robot we use for the assistive dressing experiments in this thesis.

The Baxter SDK provides interfaces for researchers to control the robot hardware layer via ROS. Any programming language that supports ROS can be used to control and program the robot directly.

A.2 MOVEIT!

In this thesis, the robot motion is planned using MoveIt! the standard ROS motion planning framework. MoveIt! has been widely used as the open-source software for mobile manipulation with more than 65 robots. It incorporates recent advances in kinematics, control, motion planning, 3D perception, navigation and manipulation. With MoveIt!, robot's arms motion can be planned together or separately. Either goal pose or goal position can be sent to the robot to execute. For each arm, the planned motion is a series of joint angle positions at each time step. MoveIt! is supported in Baxter through software configurations. Depth sensors can also be integrated with Baxter and MoveIt! for the robot to view the world in 3D.

BIBLIOGRAPHY

- Aggarwal, J. K. and Ryoo, M. S. (2011), 'Human activity analysis: A review', *ACM Computing Surveys (CSUR)* **43**(3), 16.
- Akata, Z., Perronnin, F., Harchaoui, Z. and Schmid, C. (2014), 'Good practice in large-scale learning for image classification', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(3), 507–520.
- Amirabdollahian, F., op den Akker, R., Bedaf, S., Bormann, R., Draper, H., Evers, V., Gelderblom, G. J., Ruiz, C. G., Hewson, D., Hu, N. et al. (2013), Accompany: Acceptable robotics companions for ageing years—multidimensional aspects of human-system interactions, in 'IEEE 6th International Conference on Human System Interactions (HSI)', pp. 570–577.
- Arel, I., Rose, D. C. and Karnowski, T. P. (2010), 'Deep machine learning—a new frontier in artificial intelligence research [research frontier]', *IEEE Computational Intelligence Magazine* **5**(4), 13–18.
- Argall, B. D., Chernova, S., Veloso, M. and Browning, B. (2009), 'A survey of robot learning from demonstration', *Robotics and autonomous systems* **57**(5), 469–483.
- Banala, S. K., Agrawal, S. K. and Scholz, J. P. (2007), Active leg exoskeleton (alex) for gait rehabilitation of motor-impaired patients, in 'IEEE 10th International Conference on Rehabilitation Robotics', pp. 401–407.
- Banala, S. K., Kim, S. H., Agrawal, S. K. and Scholz, J. P. (2009), 'Robot assisted gait training with active leg exoskeleton (alex)', *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **17**(1), 2–8.

BIBLIOGRAPHY

- Barea, R., Boquete, L., Mazo, M. and López, E. (2002), 'System for assisted mobility using eye movements based on electrooculography', *IEEE transactions on neural systems and rehabilitation engineering* **10**(4), 209–218.
- Bemelmans, R., Gelderblom, G. J., Jonker, P. and De Witte, L. (2012), 'Socially assistive robots in elderly care: A systematic review into effects and effectiveness', *Journal of the American Medical Directors Association* **13**(2), 114–120.
- Bengio, Y. (2009), 'Learning deep architectures for ai', *Foundations and trends® in Machine Learning* **2**(1), 1–127.
- Bengio, Y., Courville, A. and Vincent, P. (2013), 'Representation learning: A review and new perspectives', *IEEE transactions on pattern analysis and machine intelligence* **35**(8), 1798–1828.
- Bengio, Y., Goodfellow, I. J. and Courville, A. (2015), 'Deep learning', *An MIT Press book in preparation. Draft chapters available at <http://www.iro.umontreal.ca/bengioy/dlbook>*.
- Benyon, D. and Murray, D. (1993), 'Applying user modeling to human-computer interaction design', *Artificial Intelligence Review* **7**(3-4), 199–225.
- Berkovsky, S., Freyne, J. and Smith, G. (2012), Personalized network updates: increasing social interactions and contributions in social networks, in 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 1–13.
- Bersch, C., Pitzer, B. and Kammel, S. (2011), Bimanual robotic cloth manipulation for laundry folding, in 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 1413–1419.
- Billard, A., Calinon, S., Dillmann, R. and Schaal, S. (2008), Robot programming by demonstration, in 'Springer handbook of robotics', Springer, pp. 1371–1394.

- Bishop, C. M. (2006), 'Pattern recognition', *Machine Learning* **128**.
- Bohnert, F., Zukerman, I. and Laures, J. (2012), Geckommender: Personalised theme and tour recommendations for museums, in 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 26–37.
- Bonnechere, B., Jansen, B., Salvia, P., Bouzahouene, H., Omelina, L., Moiseev, F., Sholukha, V., Cornelis, J., Rooze, M. and Jan, S. V. S. (2014), 'Validity and reliability of the kinect within functional assessment activities: comparison with standard stereophotogrammetry', *Gait & posture* **39**(1), 593–598.
- Broadbent, E., Stafford, R. and MacDonald, B. (2009), 'Acceptance of health-care robots for the older population: review and future directions', *International Journal of Social Robotics* **1**(4), 319–330.
- Broekens, J., Heerink, M. and Rosendal, H. (2009), 'Assistive social robots in elderly care: a review', *Gerontechnology* **8**(2), 94–103.
- Buys, K., Cagniard, C., Baksheev, A., De Laet, T., De Schutter, J. and Pantofaru, C. (2014), 'An adaptable system for rgb-d based human body detection and pose estimation', *Journal of visual communication and image representation* **25**(1), 39–52.
- Cakmak, M., Chao, C. and Thomaz, A. L. (2010), 'Designing interactions for robot active learners', *IEEE Transactions on Autonomous Mental Development* **2**(2), 108–118.
- Cakmak, M., Srinivasa, S. S., Lee, M. K., Forlizzi, J. and Kiesler, S. (2011), Human preferences for robot-human hand-over configurations, in 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 1986–1993.

BIBLIOGRAPHY

- Cakmak, M. and Thomaz, A. L. (2012), Designing robot learners that ask good questions, *in* 'Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction', pp. 17–24.
- Calinon, S., Bruno, D. and Caldwell, D. G. (2014), A task-parameterized probabilistic model with minimal intervention control, *in* 'IEEE International Conference on Robotics and Automation', pp. 3339–3344.
- Cantrell, R., Schermerhorn, P. and Scheutz, M. (2011), Learning actions from human-robot dialogues, *in* 'IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)', pp. 125–130.
- Carignan, C. and Liszka, M. (2005), Design of an arm exoskeleton with scapula motion for shoulder rehabilitation, *in* 'IEEE 12th International Conference on Advanced Robotics', pp. 524–531.
- Carlson, T. and Demiris, Y. (2008), Human-wheelchair collaboration through prediction of intention and adaptive assistance, *in* 'IEEE International Conference on Robotics and Automation', pp. 3926–3931.
- Carlson, T. and Demiris, Y. (2012), 'Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **42**(3), 876–888.
- Carlson, T. and Millan, J. d. R. (2013), 'Brain-controlled wheelchairs: a robotic architecture', *IEEE Robotics and Automation Magazine* **20**(EPFL-ARTICLE-181698), 65–73.
- Cavallaro, E. E., Rosen, J., Perry, J. C. and Burns, S. (2006), 'Real-time myoprocessors for a neural controlled powered exoskeleton arm', *IEEE Transactions on Biomedical Engineering* **53**(11), 2387–2396.
- Cavallaro, E., Rosen, J., Perry, J. C., Burns, S. and Hannaford, B. (2005), Hill-based model as a myoprocessor for a neural controlled powered exoskel-

- eton arm-parameters optimization, *in* 'IEEE international Conference on Robotics and Automation', pp. 4514–4519.
- Cena, F., Likavec, S. and Osborne, F. (2012), Property-based interest propagation in ontology-based user model, *in* 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 38–50.
- Chan, W., Jaitly, N., Le, Q. and Vinyals, O. (2016), Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, *in* 'IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 4960–4964.
- Cheng, B. and Titterton, D. M. (1994), 'Neural networks: A review from a statistical perspective', *Statistical science* pp. 2–30.
- Chitta, S., Sucan, I. and Cousins, S. (2012), 'Moveit!', *IEEE Robotics Automation Magazine* **19**(1), 18–19.
- Choi, B. and Jo, S. (2013), 'A low-cost eeg system-based hybrid brain-computer interface for humanoid robot navigation and recognition', *PloS one* **8**(9), e74583.
- Cinar, E. and Sahin, F. (2013), 'New classification techniques for electroencephalogram (eeg) signals and a real-time eeg control of a robot', *Neural Computing and Applications* **22**(1), 29–39.
- Ciregan, D., Meier, U. and Schmidhuber, J. (2012), Multi-column deep neural networks for image classification, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3642–3649.
- Colomé, A., Planells, A. and Torras, C. (2015), A friction-model-based framework for reinforcement learning of robotic tasks in non-rigid environments, *in* 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 5649–5654.

BIBLIOGRAPHY

- Comaniciu, D. and Meer, P. (2002), 'Mean shift: A robust approach toward feature space analysis', *IEEE Transactions on pattern analysis and machine intelligence* **24**(5), 603–619.
- Dahl, G. E., Yu, D., Deng, L. and Acero, A. (2012), 'Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition', *IEEE Transactions on Audio, Speech, and Language Processing* **20**(1), 30–42.
- Dautenhahn, K., Walters, M., Woods, S., Koay, K. L., Nehaniv, C. L., Sisbot, A., Alami, R. and Siméon, T. (2006), How may i serve you?: a robot companion approaching a seated person in a helping context, in 'Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction', ACM, pp. 172–179.
- de La Gorce, M., Fleet, D. J. and Paragios, N. (2011), 'Model-based 3d hand pose estimation from monocular video', *IEEE transactions on pattern analysis and machine intelligence* **33**(9), 1793–1805.
- Desarkar, M. S., Saxena, R. and Sarkar, S. (2012), Preference relation based matrix factorization for recommender systems, in 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 63–75.
- Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T. (2014), Decaf: A deep convolutional activation feature for generic visual recognition., in 'ICML', pp. 647–655.
- Doumanoglou, A., Kargakos, A., Kim, T.-K. and Malassiotis, S. (2014), Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning, in 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 987–993.

- Duchi, J., Hazan, E. and Singer, Y. (2011), 'Adaptive subgradient methods for online learning and stochastic optimization', *The Journal of Machine Learning Research* **12**, 2121–2159.
- Egmont-Petersen, M., de Ridder, D. and Handels, H. (2002), 'Image processing with neural networks—a review', *Pattern recognition* **35**(10), 2279–2301.
- Eriksson, J., Mataric, M. J. and Winstein, C. (2005), Hands-off assistive robotics for post-stroke arm rehabilitation, in 'IEEE International Conference on Rehabilitation Robotics (ICORR)', pp. 21–24.
- Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D. and Twombly, X. (2007), 'Vision-based hand pose estimation: A review', *Computer Vision and Image Understanding* **108**(1), 52–73.
- Ewen, R. (2014), *An introduction to theories of personality*, Psychology Press.
- Fanelli, G., Weise, T., Gall, J. and Van Gool, L. (2011), Real time head pose estimation from consumer depth cameras, in 'Joint Pattern Recognition Symposium', Springer, pp. 101–110.
- Fanello, S. R., Gori, I., Metta, G. and Odone, F. (2013), 'Keep it simple and sparse: real-time action recognition.', *Journal of Machine Learning Research* **14**(1), 2617–2640.
- Fasola, J. and Mataric, M. (2013), 'A socially assistive robot exercise coach for the elderly', *Journal of Human-Robot Interaction* **2**(2), 3–32.
- Fasola, J. and Mataric, M. J. (2012), 'Using socially assistive human–robot interaction to motivate physical exercise for older adults', *Proceedings of the IEEE* **100**(8), 2512–2526.
- Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996), 'From data mining to knowledge discovery in databases', *AI magazine* **17**(3), 37.

BIBLIOGRAPHY

- Ferrari, V., Marin-Jimenez, M. and Zisserman, A. (2008), Progressive search space reduction for human pose estimation, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1–8.
- Ferris, D. P., Sawicki, G. S. and Domingo, A. (2005), 'Powered lower limb orthoses for gait rehabilitation', *Topics in spinal cord injury rehabilitation* **11**(2), 34.
- Figueiredo, M. A. and Jain, A. K. (2002), 'Unsupervised learning of finite mixture models', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **24**(3), 381–396.
- Fischer, G. (2001), 'User modeling in human–computer interaction', *User modeling and user-adapted interaction* **11**(1-2), 65–86.
- Fischer, T. and Demiris, Y. (2016), Markerless perspective taking for humanoid robots in unconstrained environments, *in* 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 3309–3316.
- Flandorfer, P. (2012), 'Population ageing and socially assistive robots for elderly persons: the importance of sociodemographic factors for user acceptance', *International Journal of Population Research* **2012**.
- Fong, T., Nourbakhsh, I. and Dautenhahn, K. (2003), 'A survey of socially interactive robots', *Robotics and autonomous systems* **42**(3), 143–166.
- Fong, T., Thorpe, C. and Baur, C. (2003), Collaboration, dialogue, human–robot interaction, *in* 'Robotics Research', Springer, pp. 255–266.
- Galán, F., Nuttin, M., Lew, E., Ferrez, P. W., Vanacker, G., Philips, J. and Millán, J. d. R. (2008), 'A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots', *Clinical Neurophysiology* **119**(9), 2159–2169.

- Gao, Y., Chang, H. J. and Demiris, Y. (2015), User modelling for personalised dressing assistance by humanoid robots, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 1840–1845.
- Girshick, R., Shotton, J., Kohli, P., Criminisi, A. and Fitzgibbon, A. (2011), Efficient regression of general-activity human poses from depth images, *in* 'IEEE International Conference on Computer Vision', pp. 415–422.
- Gong, Y., Beck, J. E. and Ruiz, C. (2012), Modeling multiple distributions of student performances to improve predictive accuracy, *in* 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 102–113.
- Gonzalez-Jorge, H., Riveiro, B., Vazquez-Fernandez, E., Martínez-Sánchez, J. and Arias, P. (2013), 'Metrological evaluation of microsoft kinect and asus xtion sensors', *Measurement* **46**(6), 1800–1806.
- Graves, A., Mohamed, A.-r. and Hinton, G. (2013), Speech recognition with deep recurrent neural networks, *in* 'IEEE international conference on acoustics, speech and signal processing', pp. 6645–6649.
- Gravot, F., Haneda, A., Okada, K. and Inaba, M. (2006), Cooking for humanoid robot, a task that needs symbolic and geometric reasonings, *in* 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 462–467.
- Han, J., Pei, J. and Kamber, M. (2011), *Data mining: concepts and techniques*, Elsevier.
- Hand, D. J., Mannila, H. and Smyth, P. (2001), *Principles of data mining*, MIT press.
- Hannon, J., McCarthy, K., O'Mahony, M. P. and Smyth, B. (2012), A multifaceted user model for twitter, *in* 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 303–309.

BIBLIOGRAPHY

- Heerink, M., Kröse, B., Evers, V., Wielinga, B. et al. (2008), 'The influence of social presence on acceptance of a companion robot by older people', *Journal of Physical Agents* **2**(2), 33–40.
- Hegels, D., Wiederkehr, T. and Müller, H. (2015), 'Simulation based iterative post-optimization of paths of robot guided thermal spraying', *Robotics and Computer-Integrated Manufacturing* **35**, 1–15.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. et al. (2012), 'Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups', *IEEE Signal Processing Magazine* **29**(6), 82–97.
- Ho, N., Tong, K., Hu, X., Fung, K., Wei, X., Rong, W. and Susanto, E. (2011), An emg-driven exoskeleton hand robotic training device on chronic stroke subjects: task training system for stroke rehabilitation, in 'IEEE international conference on rehabilitation robotics', pp. 1–5.
- Hothi, J. and Hall, W. (1998), An evaluation of adapted hypermedia techniques using static user modelling, in 'Proceedings of the second workshop on adaptive hypertext and hypermedia', pp. 45–50.
- Hu, J., Lu, J. and Tan, Y.-P. (2014), Discriminative deep metric learning for face verification in the wild, in 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1875–1882.
- Huang, G. B., Lee, H. and Learned-Miller, E. (2012), Learning hierarchical representations for face verification with convolutional deep belief networks, in 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2518–2525.
- Hüntemann, A., Demeester, E., Vander Poorten, E., Van Brussel, H. and De Schutter, J. (2013), Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model,

- in 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 4376–4383.*
- Iqbal, J., Tsagarakis, N. G. and Caldwell, D. G. (2011), A multi-dof robotic exoskeleton interface for hand motion assistance, *in 'IEEE Annual International Conference of the IEEE Engineering in Medicine and Biology Society', pp. 1575–1578.*
- Iwamura, Y., Shiomi, M., Kanda, T., Ishiguro, H. and Hagita, N. (2011), Do elderly people prefer a conversational humanoid as a shopping assistant partner in supermarkets?, *in '6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)', pp. 449–457.*
- Jain, A., Tompson, J., LeCun, Y. and Bregler, C. (2014), Modeep: A deep learning framework using motion features for human pose estimation, *in 'Asian Conference on Computer Vision', Springer, pp. 302–315.*
- Jin Chang, H. and Demiris, Y. (2015), Unsupervised learning of complex articulated kinematic structures combining motion and skeleton information, *in 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3138–3146.*
- Jung, H., Ju, J. and Kim, J. (2014), Rigid motion segmentation using randomized voting, *in 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1210–1217.*
- Kalkusch, M., Lidy, T., Knapp, N., Reitmayr, G., Kaufmann, H. and Schmalstieg, D. (2002), Structured visual markers for indoor pathfinding, *in '1st IEEE international workshop on Augmented reality toolkit'.*
- Kanda, T., Ishiguro, H., Imai, M. and Ono, T. (2003), Body movement analysis of human-robot interaction, *in 'IJCAI', Citeseer, pp. 177–182.*
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014), Large-scale video classification with convolutional neural networks,

BIBLIOGRAPHY

- in 'IEEE conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1725–1732.*
- Kawasaki, H., Ito, S., Ishigure, Y., Nishimoto, Y., Aoki, T., Mouri, T., Sakaeda, H. and Abe, M. (2007), Development of a hand motion assist robot for rehabilitation therapy by patient self-motion control, *in 'IEEE 10th International Conference on Rehabilitation Robotics', pp. 234–240.*
- Kennedy, J. (2011), Particle swarm optimization, *in 'Encyclopedia of machine learning', Springer, pp. 760–766.*
- Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S. and Fitzgibbon, A. (2015), Learning an efficient model of hand shape variation from depth images, *in 'IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2540–2548.*
- Kidd, C. D., Taggart, W. and Turkle, S. (2006), A sociable robot to encourage social interaction among the elderly, *in 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 3972–3976.*
- King, C.-H., Chen, T. L., Jain, A. and Kemp, C. C. (2010), Towards an assistive robot that autonomously performs bed baths for patient hygiene, *in 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 319–324.*
- Kingma, D. and Ba, J. (2015), A method for stochastic optimization, *in 'International Conference on Learning Representation'.*
- Kita, Y., Kanehiro, F., Ueshiba, T. and Kita, N. (2011), Clothes handling based on recognition by strategic observation, *in 'IEEE 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)', pp. 53–58.*
- Klee, S. D., Ferreira, B. Q., Silva, R., Costeira, J. P., Melo, F. S. and Veloso, M. (2015), Personalized assistance for dressing users, *in 'International Conference on Social Robotics', Springer, pp. 359–369.*

- Kobsa, A. (2001), 'Generic user modeling systems', *User modeling and user-adapted interaction* **11**(1-2), 49–63.
- Koganti, N., Ngeo, J. G., Tomoya, T., Ikeda, K. and Shibata, T. (2015), Cloth dynamics modeling in latent spaces and its application to robotic clothing assistance, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 3464–3469.
- Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* 'Advances in neural information processing systems', pp. 1097–1105.
- Kruse, D., Radke, R. J. and Wen, J. T. (2015), Collaborative human-robot manipulation of highly deformable materials, *in* 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 3782–3787.
- Kucukyilmaz, A. and Demiris, Y. (2015), One-shot assistance estimation from expert demonstrations for a shared control wheelchair system, *in* '24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)', pp. 438–443.
- Lai, K., Konrad, J. and Ishwar, P. (2012), A gesture-driven computer interface using kinect, *in* 'IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)', pp. 185–188.
- Lang, S., Kleinhagenbrock, M., Hohenner, S., Fritsch, J., Fink, G. A. and Sagerer, G. (2003), Providing the basis for human-robot-interaction: A multi-modal attention system for a mobile robot, *in* 'Proceedings of the 5th international conference on Multimodal interfaces', ACM, pp. 28–35.
- Larizza, M., Zukerman, I., Bohnert, F., Russell, R. A., Busija, L., Albrecht, D. W. and Rees, G. (2012), Studies to determine user requirements regarding in-home monitoring systems, *in* 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 139–150.

BIBLIOGRAPHY

- Latombe, J.-C. (2012), *Robot motion planning*, Vol. 124, Springer Science & Business Media.
- LeCun, Y., Bengio, Y. and Hinton, G. (2015), 'Deep learning', *Nature* **521**(7553), 436–444.
- Lee, D., Yamazaki, T. and Helal, S. (2009), 'Robotic companions for smart space interactions', *IEEE Pervasive Computing* **8**(2), 78–84.
- Lee, J.-H. et al. (2012), Full-body imitation of human motions with kinect and heterogeneous kinematic structure of humanoid robot, in 'IEEE/SICE International Symposium on System Integration (SII)', pp. 93–98.
- Lee, K., Su, Y., Kim, T.-K. and Demiris, Y. (2013), 'A syntactic approach to robot imitation learning using probabilistic activity grammars', *Robotics and Autonomous Systems* **61**(12), 1323–1334.
- Lee, M. K., Forlizzi, J., Kiesler, S., Rybski, P., Antanitis, J. and Savetsila, S. (2012), Personalization in hri: A longitudinal field experiment, in '7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)', pp. 319–326.
- Leeb, R., Tonin, L., Rohm, M., Desideri, L., Carlson, T. and Millan, J. d. R. (2015), 'Towards independence: a bci telepresence robot for people with severe motor disabilities', *Proceedings of the IEEE* **103**(6), 969–982.
- Lemaignan, S., Warnier, M., Sisbot, E. A. and Alami, R. (2014), Human-robot interaction: Tackling the ai challenges, Technical report.
- Leyzberg, D., Spaulding, S. and Scassellati, B. (2014), Personalizing robot tutors to individuals' learning differences, in 'ACM/IEEE international conference on Human-robot interaction', pp. 423–430.
- Lo, A. C., Guarino, P. D., Richards, L. G., Haselkorn, J. K., Wittenberg, G. F., Federman, D. G., Ringer, R. J., Wagner, T. H., Krebs, H. I., Volpe, B. T. et al.

- (2010), 'Robot-assisted therapy for long-term upper-limb impairment after stroke', *New England Journal of Medicine* **362**(19), 1772–1783.
- Lowe, D. G. (1999), Object recognition from local scale-invariant features, in '7th IEEE international conference on Computer vision', Vol. 2, pp. 1150–1157.
- Lum, P. S., Burgar, C. G., Shor, P. C., Majmundar, M. and Van der Loos, M. (2002), 'Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke', *Archives of physical medicine and rehabilitation* **83**(7), 952–959.
- Luo, R. C., Chen, S. Y. and Yeh, K. C. (2014), Human body trajectory generation using point cloud data for robotics massage applications, in 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 5612–5617.
- Luo, S., Mou, W., Althoefer, K. and Liu, H. (2015), 'Novel tactile-sift descriptor for object shape recognition', *IEEE Sensors Journal* **15**(9), 5001–5009.
- Maglaveras, N., Stamkopoulos, T., Diamantaras, K., Pappas, C. and Strintzis, M. (1998), 'Ecg pattern recognition and classification using non-linear transformations and neural networks: a review', *International journal of medical informatics* **52**(1), 191–208.
- Maitin-Shepard, J., Cusumano-Towner, M., Lei, J. and Abbeel, P. (2010), Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding, in 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 2308–2315.
- Marcheschi, S., Salsedo, F., Fontana, M. and Bergamasco, M. (2011), Body extender: whole body exoskeleton for human power augmentation, in 'IEEE

BIBLIOGRAPHY

- International Conference on Robotics and Automation (ICRA)', pp. 611–616.
- Martin, C., Schaffernicht, E., Scheidig, A. and Gross, H.-M. (2006), 'Multi-modal sensor fusion using a probabilistic aggregation scheme for people detection and tracking', *Robotics and Autonomous Systems* **54**(9), 721–728.
- Masehian, E. and Sedighizadeh, D. (2007), 'Classic and heuristic approaches in robot motion planning-a chronological review', *World Academy of Science, Engineering and Technology* **23**, 101–106.
- Masiero, S., Celia, A., Rosati, G. and Armani, M. (2007), 'Robotic-assisted rehabilitation of the upper limb after acute stroke', *Archives of physical medicine and rehabilitation* **88**(2), 142–149.
- Mason, M. and Lopes, M. (2011), Robot self-initiative and personalization by learning through repeated interactions, in '6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)', pp. 433–440.
- Matarić, M. J., Eriksson, J., Feil-Seifer, D. J. and Winstein, C. J. (2007), 'Socially assistive robotics for post-stroke rehabilitation', *Journal of NeuroEngineering and Rehabilitation* **4**(1), 1.
- Matsumoto, Y., Ino, T. and Ogasawara, T. (2001), Development of intelligent wheelchair system with face and gaze based interface, in '10th IEEE International Workshop on Robot and Human Interactive Communication', pp. 262–267.
- Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (2013), *Machine learning: An artificial intelligence approach*, Springer Science & Business Media.
- Moeslund, T. B. and Granum, E. (2001), 'A survey of computer vision-based human motion capture', *Computer vision and image understanding* **81**(3), 231–268.

- Moeslund, T. B., Hilton, A. and Krüger, V. (2006), 'A survey of advances in vision-based human motion capture and analysis', *Computer vision and image understanding* **104**(2), 90–126.
- Mohareri, O., Salcudean, S. E. and Nguan, C. (2013), Asymmetric force feedback control framework for teleoperated robot-assisted surgery, in 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 5800–5806.
- Montaner, M., López, B. and De La Rosa, J. L. (2003), 'A taxonomy of recommender agents on the internet', *Artificial intelligence review* **19**(4), 285–330.
- Mukai, T., Onishi, M., Odashima, T., Hirano, S. and Zhiwei, L. (2008), 'Development of the tactile sensor system of a human-interactive robot ri-man', *IEEE Transactions on Robotics* **24**(2), 505–512.
- Murphy, K. P. (2012), *Machine learning: a probabilistic perspective*, MIT press.
- Musto, C., Narducci, F., Lops, P., Semeraro, G., De Gemmis, M., Barbieri, M., Korst, J., Pronk, V. and Clout, R. (2012), Enhanced semantic tv-show representation for personalized electronic program guides, in 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 188–199.
- Nair, V. and Hinton, G. E. (2009), 3d object recognition with deep belief nets, in 'Advances in Neural Information Processing Systems', pp. 1339–1347.
- Nakamura, Y. and Hanafusa, H. (1986), 'Inverse kinematic solutions with singularity robustness for robot manipulator control', *Journal of dynamic systems, measurement, and control* **108**(3), 163–171.
- Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H. and Ng, A. Y. (2011), Multimodal deep learning, in 'Proceedings of the 28th international conference on machine learning (ICML)', pp. 689–696.

BIBLIOGRAPHY

- Nickel, K. and Stiefelwagen, R. (2003), Pointing gesture recognition based on 3d-tracking of face, hands and head orientation, *in* '5th ACM International conference on Multimodal interfaces', pp. 140–146.
- Nikolaidis, S., Ramakrishnan, R., Gu, K. and Shah, J. (2015), Efficient model learning from joint-action demonstrations for human-robot collaborative tasks, *in* 'Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction', pp. 189–196.
- Nixon, M. (2008), *Feature extraction & image processing*, Academic Press.
- Ohmura, Y. and Kuniyoshi, Y. (2007), Humanoid robot which can lift a 30kg box by whole body contact and tactile feedback, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 1136–1141.
- Oikonomidis, I., Kyriazis, N. and Argyros, A. A. (2011), Efficient model-based 3d tracking of hand articulations using kinect., *in* 'British Machine Vision Conference (BMVC)', Vol. 1, p. 3.
- Ouyang, W., Chu, X. and Wang, X. (2014), Multi-source deep learning for human pose estimation, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2329–2336.
- Pandey, A. K. and Alami, R. (2010), Mightability maps: A perceptual level decisional framework for co-operative and competitive human-robot interaction, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 5842–5848.
- Pandey, A. K., Ali, M. and Alami, R. (2013), 'Towards a task-aware proactive sociable robot based on multi-state perspective-taking', *International Journal of Social Robotics* 5(2), 215–236.
- Paquette, L., Lebeau, J.-F. and Mayers, A. (2012), Automating the modeling of learners' erroneous behaviors in model-tracing tutors, *in* 'International

- Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 316–321.
- Perry, J. C., Rosen, J. and Burns, S. (2007), 'Upper-limb powered exoskeleton design', *IEEE/ASME transactions on mechatronics* **12**(4), 408.
- Petit, M., Fischer, T. and Demiris, Y. (2016), 'Lifelong augmentation of multimodal streaming autobiographical memories', *IEEE Transactions on Cognitive and Developmental Systems* .
- Poppe, R. (2007), 'Vision-based human motion analysis: An overview', *Computer vision and image understanding* **108**(1), 4–18.
- Poppe, R. (2010), 'A survey on vision-based human action recognition', *Image and vision computing* **28**(6), 976–990.
- Raja, P. and Pugazhenth, S. (2012), 'Optimal path planning of mobile robots: A review', *International Journal of Physical Sciences* **7**(9), 1314–1320.
- Ramisa, A., Alenya, G., Moreno-Noguer, F. and Torras, C. (2012), Using depth and appearance features for informed robot grasping of highly wrinkled clothes, in 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 1703–1708.
- Rebsamen, B., Guan, C., Zhang, H., Wang, C., Teo, C., Ang, M. H. and Burdet, E. (2010), 'A brain controlled wheelchair to navigate in familiar environments', *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **18**(6), 590–598.
- Rich, E. (1979), 'User modeling via stereotypes', *Cognitive science* **3**(4), 329–354.
- Robinson, H., MacDonald, B., Kerse, N. and Broadbent, E. (2013), 'The psychosocial effects of a companion robot: A randomized controlled trial', *Journal of the American Medical Directors Association* **14**(9), 661–667.

BIBLIOGRAPHY

- Rogez, G., Rihan, J., Ramalingam, S., Orrite, C. and Torr, P. H. (2008), Randomized trees for human pose detection, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1–8.
- Ros, R. and Demiris, Y. (2013), Creative dance: An approach for social interaction between robots and children, *in* 'International Workshop on Human Behavior Understanding', Springer, pp. 40–51.
- Rozo, L., Calinon, S., Caldwell, D., Jimenez, P. and Torras, C. (2016), 'Learning physical collaborative robot behaviors from human demonstrations', *IEEE Trans. on Robotics* .
- Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986), 'Learning internal representations by error propagation', *Parallel Distributed Processing* pp. 318–362.
- Rumelhart, D. E., Hintont, G. E. and Williams, R. J. (1986), 'Learning representations by back-propagating errors', *NATURE* **323**, 9.
- Sak, H., Senior, A., Rao, K., Irsoy, O., Graves, A., Beaufays, F. and Schalkwyk, J. (2015), Learning acoustic frame labeling for speech recognition with recurrent neural networks, *in* 'IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 4280–4284.
- Salvendy, G. (2012), *Handbook of human factors and ergonomics*, John Wiley & Sons.
- Sameni, R. and Clifford, G. D. (2010), 'A review of fetal ecg signal processing; issues and promising directions', *The open pacing, electrophysiology & therapy journal* **3**, 4.
- Sarabia, M. and Demiris, Y. (2013), A humanoid robot companion for wheelchair users, *in* 'International Conference on Social Robotics', Springer, pp. 432–441.

- Satoh, H., Kawabata, T. and Sankai, Y. (2009), Bathing care assistance with robot suit hal, *in* 'IEEE International Conference on Robotics and Biomimetics (ROBIO)', pp. 498–503.
- Schaal, S. (1999), 'Is imitation learning the route to humanoid robots?', *Trends in cognitive sciences* 3(6), 233–242.
- Schabowsky, C. N., Godfrey, S. B., Holley, R. J. and Lum, P. S. (2010), 'Development and pilot testing of hexorr: hand exoskeleton rehabilitation robot', *Journal of neuroengineering and rehabilitation* 7(1), 1.
- Schmidhuber, J. (2015), 'Deep learning in neural networks: An overview', *Neural Networks* 61, 85–117.
- Schmidts, A. M., Lee, D. and Peer, A. (2011), Imitation learning of human grasping skills from motion and force data, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 1002–1007.
- Schroeter, C., Mueller, S., Volkhardt, M., Einhorn, E., Huijnen, C., van den Heuvel, H., van Berlo, A., Bley, A. and Gross, H.-M. (2013), Realization and user evaluation of a companion robot for people with mild cognitive impairments, *in* 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 1153–1159.
- Seemann, E., Nickel, K. and Stiefelhagen, R. (2004), Head pose estimation using stereo vision for human-robot interaction, *in* '6th IEEE International Conference on Automatic Face and Gesture Recognition', pp. 626–631.
- Seltzer, M. L., Yu, D. and Wang, Y. (2013), An investigation of deep neural networks for noise robust speech recognition, *in* 'IEEE International Conference on Acoustics, Speech and Signal Processing', pp. 7398–7402.
- Shen, J., Brdiczka, O., Ducheneaut, N., Yee, N. and Begole, B. (2012), Inferring personality of online gamers by fusing multiple-view predictions, *in*

BIBLIOGRAPHY

- 'International Conference on User Modeling, Adaptation, and Personalization', Springer, pp. 261–273.
- Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M. and Moore, R. (2013), 'Real-time human pose recognition in parts from single depth images', *Communications of the ACM* **56**(1), 116–124.
- Simonyan, K. and Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* .
- Simpson, R. C. (2005), 'Smart wheelchairs: A literature review', *Journal of rehabilitation research and development* **42**(4), 423.
- Sisbot, E. A., Marin, L. F. and Alami, R. (2007), Spatial reasoning for human robot interaction, in 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 2281–2287.
- Socher, R., Huval, B., Bath, B., Manning, C. D. and Ng, A. Y. (2012), Convolutional-recursive deep learning for 3d object classification, in 'Advances in Neural Information Processing Systems', pp. 665–673.
- Soh, H. and Demiris, Y. (2013), When and how to help: An iterative probabilistic model for learning assistance by demonstration, in 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 3230–3236.
- Soh, H. and Demiris, Y. (2014), 'Incrementally learning objects by touch: On-line discriminative and generative models for tactile-based recognition', *IEEE transactions on haptics* **7**(4), 512–525.
- Stiefelhagen, R., Ekenel, H. K., Fugen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M. and Waibel, A. (2007), 'Enabling multimodal human–robot interaction for the karlsruhe humanoid robot', *IEEE Transactions on Robotics* **23**(5), 840–851.

- Stiefelhagen, R., Fugen, C., Gieselmann, R., Holzapfel, H., Nickel, K. and Waibel, A. (2004), Natural human-robot interaction using speech, head pose and gestures, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', Vol. 3, pp. 2422–2427.
- Sun, Y., Wang, X. and Tang, X. (2013), Deep convolutional network cascade for facial point detection, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3476–3483.
- Sun, Y., Wang, X. and Tang, X. (2014), Deep learning face representation from predicting 10,000 classes, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1891–1898.
- Taigman, Y., Yang, M., Ranzato, M. and Wolf, L. (2014), Deepface: Closing the gap to human-level performance in face verification, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1701–1708.
- Tamei, T., Matsubara, T., Rai, A. and Shibata, T. (2011), Reinforcement learning of clothing assistance with a dual-arm robot, *in* '11th IEEE/RAS International Conference on Humanoid Robots (Humanoids)', pp. 733–738.
- Tang, D., Jin Chang, H., Tejani, A. and Kim, T.-K. (2014), Latent regression forest: Structured estimation of 3d articulated hand posture, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 3786–3793.
- Tang, D., Taylor, J., Kohli, P., Keskin, C., Kim, T.-K. and Shotton, J. (2015), Opening the black box: Hierarchical sampling optimization for estimating human hand pose, *in* 'Proceedings of the IEEE International Conference on Computer Vision', pp. 3325–3333.
- Tang, D., Yu, T.-H. and Kim, T.-K. (2013), Real-time articulated hand pose estimation using semi-supervised transductive regression forests, *in* 'Pro-

BIBLIOGRAPHY

- ceedings of the IEEE International Conference on Computer Vision', pp. 3224–3231.
- Tapus, A., Maja, M. and Scassellatti, B. (2007), 'The grand challenges in socially assistive robotics', *IEEE Robotics and Automation Magazine* **14**(1), N–A.
- Tapus, A., Mataric, M. J. and Scassellatti, B. (2007), 'Socially assistive robotics', *IEEE Robotics and Automation Magazine* **14**(1), 35.
- Tapus, A., Țăpuș, C. and Matarić, M. J. (2008), 'User—robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy', *Intelligent Service Robotics* **1**(2), 169–183.
- Tax, D. M. and Duin, R. P. (2004), 'Support vector data description', *Machine learning* **54**(1), 45–66.
- Tieleman, T. and Hinton, G. (2012), 'Lecture 6.5-rmsprop', *COURSERA: Neural networks for machine learning*.
- Tompson, J. J., Jain, A., LeCun, Y. and Bregler, C. (2014), Joint training of a convolutional network and a graphical model for human pose estimation, in 'Advances in neural information processing systems', pp. 1799–1807.
- Torta, E., Oberzaucher, J., Werner, F., Cuijpers, R. H. and Juola, J. F. (2012), 'Attitudes towards socially assistive robots in intelligent homes: results from laboratory studies and field trials', *Journal of Human-Robot Interaction* **1**(2), 76–99.
- Toshev, A. and Szegedy, C. (2014), Deeppose: Human pose estimation via deep neural networks, in 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1653–1660.

- Turaga, P., Chellappa, R., Subrahmanian, V. S. and Udrea, O. (2008), 'Machine recognition of human activities: A survey', *IEEE Transactions on Circuits and Systems for Video Technology* **18**(11), 1473–1488.
- Van Den Berg, J., Miller, S., Goldberg, K. and Abbeel, P. (2010), Gravity-based robotic cloth folding, in 'Algorithmic Foundations of Robotics IX', Springer, pp. 409–424.
- Veneman, J. F., Kruidhof, R., Hekman, E. E., Ekkelenkamp, R., Van Asseldonk, E. H. and Van Der Kooij, H. (2007), 'Design and evaluation of the lopes exoskeleton robot for interactive gait rehabilitation', *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **15**(3), 379–386.
- Vu, K.-P. L. and Proctor, R. W. (2011), *Handbook of human factors in Web design*, CRC Press.
- Wada, K. and Shibata, T. (2007), 'Living with seal robots—its sociopsychological and physiological influences on the elderly at a care house', *IEEE Transactions on Robotics* **23**(5), 972–980.
- Wang, R. Y. and Popović, J. (2009), Real-time hand-tracking with a color glove, in 'ACM transactions on graphics (TOG)', Vol. 28, ACM, p. 63.
- Webb, G. I., Pazzani, M. J. and Billsus, D. (2001), 'Machine learning for user modeling', *User modeling and user-adapted interaction* **11**(1-2), 19–29.
- Willimon, B., Birchfield, S. and Walker, I. D. (2011), Model for unfolding laundry using interactive perception., in 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 4871–4876.
- Witten, I. H. and Frank, E. (2005), *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann.

BIBLIOGRAPHY

- Wu, Y. and Demiris, Y. (2010), Towards one shot learning by imitation for humanoid robots, *in* 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 2889–2894.
- Wu, Y. and Huang, T. S. (1999), Vision-based gesture recognition: A review, *in* 'International Gesture Workshop', Springer, pp. 103–115.
- Wu, Y., Su, Y. and Demiris, Y. (2014), 'framework for robot learning by demonstration: Integrating one-shot and incremental learning approaches', *Robotics and Autonomous Systems* .
- Yamakawa, Y., Namiki, A. and Ishikawa, M. (2011), Motion planning for dynamic folding of a cloth with two high-speed robot hands and two high-speed sliders, *in* 'IEEE International Conference on Robotics and Automation (ICRA)', pp. 5486–5491.
- Yamazaki, K., Ueda, R., Nozawa, S., Mori, Y., Maki, T., Hatao, N., Okada, K. and Inaba, M. (2010), System integration of a daily assistive robot and its application to tidying and cleaning rooms, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', pp. 1365–1371.
- Yin, H., Cui, B., Chen, L., Hu, Z. and Zhou, X. (2015), 'Dynamic user modeling in social media systems', *ACM Transactions on Information Systems (TOIS)* **33**(3), 10.
- Yub Jung, H., Lee, S., Seok Heo, Y. and Dong Yun, I. (2015), Random tree walk toward instantaneous 3d human pose estimation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2467–2474.
- Zambelli, M. and Demiris, Y. (2016), Multimodal imitation using self-learned sensorimotor representations, *in* 'IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)'.

Zhang, Z. (2012), 'Microsoft kinect sensor and its effect', *IEEE multimedia*
19(2), 4-10.

BIBLIOGRAPHY