Imperial College of Science, Technology and Medicine
Department of Computing
Hamlyn Centre for Robotic Surgery

Implicit Human-Robot Interfacing in Robotic Surgery

Gauthier Gras

Supervised by Professor Guang-Zhong Yang

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, October 2018

Declaration

This thesis presents original work of the titled author. All material in this dissertation which is not the author's original work has been properly acknowledged and referenced accordingly.

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.

A user interface is like a joke. If you have to explain it, it's not that good.

– Martin Leblanc

Abstract

The surgical environment presents a large number of challenges for surgeons, from the crucial planning and decision-making processes involved, to the technical skills required to complete a surgical task. To tackle these challenges, systems of increasing capabilities have been designed. Surgical robots provide enhanced dexterity under the restrictive conditions of minimally invasive surgery, and advanced imaging and registration modalities allow powerful planning and localisation capabilities for surgery.

However, despite the increasing complexity and capability of modern surgical robot systems, human-robot interfaces have remained largely unchanged. Modern surgical robots possess many degrees of freedom and system-specific dexterous workspaces, yet are still mainly controlled using a static motion scaling master-slave scheme in Cartesian space. Likewise, image guidance information displayed on separate displays force interruptions of the surgical workflow when used, as the surgeon needs to take their attention away from the surgical scene. In order to fully utilise the capabilities provided by these systems, improved user-interfaces taking advantage of the context of the surgical scene need to be developed.

This thesis explores the design of such human-user interfaces for two main cases: interfaces responsible for the control of robotic systems, and surgical navigation interfaces. A key motivation for this work is to generate methods for robotic control and navigation that feel transparent to the user. These interfaces should act implicitly based on the context of the surgical scene, as opposed to requiring explicit user input. Firstly, an implicit approach to control the surgical camera and a focussed energy delivery system is explored, using gaze information and no active gestures. This approach is then expanded for intention recognition, to dynamically adapt motion scaling of master-slave systems. Bayesian approaches to tune these interfaces are also studied, showing the importance of not relying on manual tuning. The methods previously developed for intention recognition are then expanded and applied to augmented reality, to provide optimal display behaviour in robotic applications. Finally, an approach is developed combining augmented reality and force feedback to provide implicit assistance for hand-held robotic instruments. Throughout this work, results from studies conducted for each of the proposed methods demonstrate the capabilities and potential clinical value of context-aware interfaces to improve performance.

Acknowledgements

My time at the Hamlyn Centre has been an incredible experience. Formative, challenging, and at times even surreal, it has allowed me to forge great friendships and learn more about my own drives and goals. I first arrived at the Hamlyn Centre in 2012 to take part in the second year of the MRes course in robotics. Since then, the six years that I have spent at the Hamlyn Centre constitute one of the most enriching and character-defining periods of my life. Asides from technical skills, these years have taught me to manage deadlines, work as part of a team, present my work in front of entire concert halls, and even discuss about research directions with world leaders.

First and foremost, I would like to extend my deepest thanks to Professor Guang-Zhong Yang for allowing me to pursue a PhD in the Hamlyn Centre. His guidance and feedback have been invaluable, while allowing me the freedom to pursue my research topic. His constant efforts to improve the Hamlyn Centre have made it an incredible place to conduct robotics work, and provided me with research opportunities I would never have been able to experience elsewhere.

I have had the chance to be surrounded by extremely bright researches and engineers during my studies. In particular, I would like to express how grateful I am to have been able to conduct my PhD alongside Konrad Leibrandt, Piyamate Wisanuvej, and Petros Giataganas. Working with them has been a complete pleasure, as well as an extremely enriching experience. I have learned so much from them, and hope I was able to give back to them as much as they gave to me.

I would also like to thank Jianzhong Shang, Hani Marcus, Chris Payne, Menglong Ye, Carlo Seneci, and Jindong Liu for their friendship and invaluable contributions to my work. They have taught me the importance of teamwork and have always been there to provide me with useful advice. I would like to further extend these thanks to Valentina Vitiello, Su-Lin Lee, Stamatia Giannarou, George Mylonas, Dan Elson, and Kenko Fujii. They have been incredibly helpful, providing moral support and meaningful insights, both with regards to research considerations as well as for life in general. Special thanks also go to Lin Zhang, Pierre Berthet-Rayne, and Andreas Schmitz, who have all been great fun to work with, as well as highly resourceful colleagues.

I have been privileged to pursue my PhD at the same time as Maura Power, and would like to thank her deeply for her unwavering support and advice. When I

needed help proofreading a paper at 4am, 4 hours before the submission deadline, she would do it. For all your help throughout the years, thank you.

Finally, I would like to acknowledge all the incredible support provided by my amazing mom and dad, and my awesome brothers Beranger and Terence. I would not be where I am without you all. I also want to send my deepest thanks to my grandparents, who have always been amazingly supportive regardless of how good I was at explaining what I was doing. Lastly, I want to dedicate this thesis to my grandfather Yves who passed away abruptly during my PhD. He was a quiet and kind man, and I hope he would be proud.

Contents

Declaration i

Abstract ii

Acknowledgements iv

List of Figures xi

List of Tables xxii

1 Introduction 1

1.1 Research Objectives 3
1.2 Original Contributions 7
1.3 Publications 8

2 Control and Visualisation Interfaces in Robotic Surgery 11

2.1 Introduction 11
2.2 Human-Robot Control Interfaces in Robotic Surgery 12
2.2.1 Grounded Robotic Haptic Master Interfaces 13
2.2.2 Ungrounded Haptic Master Interfaces 20
2.2.3 Gesture and Voice Controlled Interfaces 24
2.2.4 Hands-On Robots, Automated, and Semi-Automated Interfaces 26
2.2.5 Summary 27
2.3 Gaze-Contingent Interfaces in Surgery 27
2.3.1 Gaze-Contingent Robot Control in Surgery 30

2.3.2 Case Study: Gaze Gestures for Human Robot Interaction in Laparoscopic Surgery . . . 34
2.4 Visualisation Interfaces in Robotic Surgery . . . 41
2.4.1 Surgical Navigation Interfaces . . . 41
2.4.2 Augmented Reality Display Interfaces in Surgery . . . 45
2.4.3 Augmented Reality Perception Issues and Display Choices in Surgery . . . 50
2.5 Conclusion . . . 54

3 Gaze-Contingent Control for Focussed Energy Delivery 56
3.1 Introduction . . . 56
3.2 Gaze-Contingent Laparoscope and Laser Control . . . 58
3.2.1 System Overview . . . 58
3.2.2 Interaction Detection . . . 59
3.2.3 Panning Velocity Control . . . 63
3.2.4 Gaze-Centric Stereo Reconstruction . . . 65
3.2.5 Interface Design . . . 66
3.3 Control and Calibration of System Components . . . 66
3.3.1 Calibration of Robotic Arms . . . 68
3.3.2 Camera Calibration and Remote Centre of Motion Considerations . . . 70
3.3.3 Eye Tracker Calibration . . . 72
3.3.4 Laser Calibration . . . 73
3.3.5 Error Minimization . . . 74
3.4 System Evaluation . . . 77
3.4.1 User Study Protocol . . . 77
3.4.2 Results . . . 79
3.5 Discussion and Conclusion . . . 82

4 Intention Recognition for Motion Assistance 84
4.1 Introduction . . . 84
4.2 A Highly Dexterous Robotic System for Surgery . . . 86
4.2.1 Articulated Instruments for Single Port Surgery . . . 86
4.2.2 Highly Redundant Master-Slave System . . . 88

4.2.3 Instrument Kinematics 90
4.2.4 Dexterity Analysis for Task Space Restriction 94
4.3 Gaze-Assisted Adaptive Motion Scaling 97
4.3.1 Limitations of Master Devices for Highly Dexterous Slave Systems 97
4.3.2 Definition of Aims and Behaviour Analysis 98
4.3.3 Instrument Tracking 99
4.3.4 Adaptive Scaling Generation 100
4.3.5 Algorithm Behaviour Analysis 103
4.4 User Studies 104
4.4.1 Experimental Setup 104
4.4.2 Results 106
4.5 Discussion and Conclusion 107

5 User-Specific Adaptation of Motion Behaviour 110
5.1 Introduction 110
5.2 Human-Robot Interface Optimisation 112
5.2.1 Overview of optimisation Techniques and Requirements . 112
5.2.2 Gaussian Processes for Bayesian Learning 113
5.2.3 Gaussian Processes for Preference Relationships 120
5.2.4 Bayesian optimisation 123
5.3 Application of Bayesian Optimisation to Adaptive Motion Scaling 127
5.3.1 System Design and Improvements 128
5.3.2 Experimental Protocol 129
5.3.3 Results 131
5.4 Discussion and Conclusion 135

6 Context-Aware Augmented Reality Display 137
6.1 Introduction 137
6.2 Simulator and System Overview 139
6.2.1 Simulator 139
6.2.2 System Overview 141
6.2.3 Model Inputs and Outputs 142
6.3 Multi Gaussian Process Modelling 145

6.3.1 Gaussian Process Voters with Meaningful Priors 145
6.3.2 Hyperparameter Optimisation 150
6.4 User Study 154
6.4.1 Model Training 154
6.4.2 Experimental Task and Protocol 156
6.4.3 Results 159
6.5 Discussion and Conclusion 162

7 Visual Force-Feedback **164**
7.1 Introduction 164
7.2 System Overview 167
7.3 Augmented Reality Modulation 168
7.4 An Efficient Rendering Pipeline for Stereo Augmented Reality . . 172
7.4.1 Real-Time Pipeline 172
7.4.2 Stereo Rendering 175
7.5 Force-Based Displacement 179
7.5.1 Deformation Modelling 179
7.5.2 Alternative Methods for Deformation Modelling 181
7.6 User Studies 184
7.6.1 Retraction Study 185
7.6.2 Dissection Study 187
7.6.3 Depth Perception Study 188
7.7 Discussion and Conclusion 190

8 Conclusions and Future Research Directions **193**
8.1 Summary of Thesis Achievements 194
8.2 Discussion and Future Research Directions 197

A Theoretical Formulation of the Online Gaze Calibration **201**
A.1 Application of Thin Plate Spline to Gaze Mapping 201
A.2 Thin Plate Spline Mapping Parameter Estimation 203

B Quantification of Calibration Drift **206**
B.1 Experimental Setup 206
B.2 Results 207

C Bayesian Learning **210**

D Intellectual Property Re-use Permissions **215**

D.1 IEEE Permissions 215

D.2 Elsevier Permissions 215

D.3 Springer Permissions 216

D.4 Permissions from Other Publishers 217

Bibliography **218**

List of Figures

2.1 From left to right: Da Vinci X®, Da Vinci SP®, Da Vinci XI®, and surgeon console. ©Intuitive Surgical. 16
2.2 Illustrations of three haptic master devices. (a) Touch™ Haptic Device, ©3D Systems. Formerly Phantom Omni®, ©SenAble. (b) Omega.7®and (c) Sigma.7®, courtesy of Force Dimension, Switzerland. 17
2.3 Examples of haptic master devices that generate commands in joint-space. (a) Master device using a serial-link structure matching the kinematic structure of the controlled instruments [18], ©2017 IEEE. (b) Master device with a parallel structure [17], ©2016 IEEE. 18
2.4 Example joystick/gamepad controllers used in surgical robot systems. (a) Monarch®, ©Auris Health Inc. (b) MMS-II ©2010 IEEE [16]. 21
2.5 Hand-held controllers tracked using inertial sensors and infrared cameras. (a) Oculus Rift®system , ©Facebook Inc. (b) HTC Vive®, ©HTC. 22
2.6 Example of an electromagnetically-tracked master device in the shape of a hand-held gripper, using the Ascension trakStar™ system. (a) and (b): rendering of the master device as used in [29] and [30]. (c): Master device used to control a 6 DoF surgical instrument [47]. 23

2.7 Example of data gloves as control interfaces. (a) CyberGlove®used to assess ergonomics in laparoscopic surgery, ©2010 IEEE [42]. (b) CyberGrasp®tested haptic rendering in teleoperation applications, ©2013 IEEE [48]. (c) CyberForce®used for measuring improvement in post-stroke rehabilitation patients, ©2007 IEEE [49]. 24
2.8 Hand gestures used to request surgical tools from a robotic assistant, ©IEEE 2013 [34]. 25
2.9 (a) Steady-hand robot, ©2007 IEEE [52]. (b) ROBODOC surgical system, ©2007 IEEE [53]. 26
2.10 (a) Tx300 monitor-mounted eye tracker, ©Tobii Technology. (b) SMI Eye Tracking Glasses ©SensoMotoric Instruments (SMI). (c) Retro-fitted eye tracker for the Da Vinci robot, ©IEEE 2015 [59]. 28
2.11 Smart operating room using eye tracking glasses and a Kinect sensor to map the user's gaze to 3D coordinates. A robot-mounted laser is used to highlight the fixated spots. Reproduced from Springer open access [69]. 30
2.12 Gaze-controlled surgical cameras: (a) Using joint control through fixation-activated buttons, ©IEEE 2010 [76]. (b) Using joint control mapped from 2D fixations, reproduced from Springer open access [78]. 32
2.13 (a) The CoBRASurge robotic laparoscope controlled using an S2 eye tracker (Mirametrix Inc.), ©IEEE 2013 [84]. (b) The ARAMIS gaze-contingent camera control system using eye tracking glasses, ©IEEE 2012 [85]. 33
2.14 (a) Gaze-contingent control of a robotic probe using fixations as 3D setpoints, ©IEEE 2008 [86]. (b) Active constraints generated by the surgeon through gaze fixations, reproduced with permission from Springer [87]. 34
2.15 Illustration of the gaze gestures used to enable the (a) *activate camera* mode; and (b) *tilt camera* mode. Picture from the co-authored publication [94]. 36

2.16 Illustration of the different types of control motions for the gaze-gesture robotic laparoscope system. (a) Panning motion. (b) Zoom-in motion. (c) Zoom-out motion. (d) Tilting motion. (e) Stopping motion by fixation of the left hand corner for 750ms. Picture from the co-authored publication [94]. 38
2.17 Operative workflow using the gaze-assisted robotic control. Picture from the co-authored publication [94]. 40
2.18 Intra-operative image guidance of a tracked instrument using a triplanar display. Reproduced with permission from Springer [103]. 42
2.19 Comparison between volume rendering (a), and surface rendering (b). The patient anatomy has been segmented for the surface rendering. Reproduced with permission from the Hepatobiliary Surgery and Nutrition journal [108]. 44
2.20 (a) Augmented reality display using a tablet, reproduced with permission from Karger [113]. (b) Augmented reality display using an opaque head-mounted display, ©IEEE 2009 [115]. 46
2.21 Augmented reality interface using a projector to display the image guidance information directly onto the patient. Reproduced with permission from the Wiley Online Library [120]. 47
2.22 See-through head mounted displays. (a) Google Glass™, reproduced with permission from Elsevier [123]. (b-c) Microsoft HoloLens™, reproduced with permission from Springer [124]. 48
2.23 (a) Virtual mirror used to provide additional viewing angles and improve scene perception. ©IEEE 2009 [115]. (b) Augmented reality overlaying 2D ultrasound information, reproduced with permission from Springer [128]. 50
2.24 Augmented reality overlays using only transparency to merge with the scene. (a-b) DEX-Ray system, reproduced with permission from Neurosurgery [129]. (c) Augmented reality system for the treatment of cerebral arteriovenous malformations, reproduced with permission from Springer [118]. 51

2.25 Augmented reality using depth clues such as occlusion, shadows, and boundary highlighting. (a,b,c) Anatomic mimesis, ©IEEE 2007 [135]. (d) Augmented reality system using volume rendering, reproduced with permission from Annual Reviews [136]. . . . 53
2.26 Augmented reality using distinct filtering and occlusion effects to enhance depth perception. (a) The *Inverse Realism* non-photorealistic rendering, reproduced with permission from Springer [137]. (b) Application in robotic partial nephrectomy, reproduced with permission from Springer [138]. 54

3.1 System overview. (a) Robot 1 holding the stereoscope, (b) Robot 2 holding the laser diode, (c) Tobii X2-60, (d) View of the laser dot on the screen. 58
3.2 Interaction detection state diagram. 60
3.3 Top: blue and red lines respectively represent whether the left and right eye are visible (value of 1) or not (value of 0). Background colors represent the state of the sytem: red = OFF, yellow = PAUSED, green = ON. Center: blue and red lines represent the normalized X and Y screen coordinates of the average gaze point (averaged between the left and right eyes, on-screen values are between 0 and 1, with 0.5 being the center of the screen). Background color represent whether the user is looking inside the screen limits, white: within limits / *Out* check succeeded; blue: outside limits / *Out* check failed. Bottom: blue and red lines same as center figure. Magenta line shows if either one of the two eyes is detected (values same as top figure). Background colors same as top figure. 62
3.4 Illustration of the RCM constraint. (a) Representation of the RCM constraint around the trocar (centre). The red dot represents the initial scope position. (b) Representation of the range of motion of the scope around a fixed RCM. The colour gradient indicates the magnitude of the angle of rotation away from the initial position (illustrated by the red dot). 63

3.5 Generation of velocity vectors based on eye data. The blue ellipse illustrates the boundaries of an example zero velocity zone. The red arrow represents the two-dimensional vector used to generate desired angular velocities. The overlaid heatmap shows typical eye motions in this direction. 64
3.6 Visualisation of the velocity field in (a) standard screen coordinates, and (b) in normalized screen coordinates. 65
3.7 Example activation (a), and deactivation (b) of the system. The circle fades over 800 ms, indicating the system status. 67
3.8 Setup for the calibration of the robotic arms. 68
3.9 Stereoscope rigidly affixed on a robot end effector. The hand-eye transformation ${}^{K}T_{C}$ connects the robot end effector frame K with the camera frame C. 70
3.10 Example eye tracking calibration pattern. 72
3.11 Laser calibration setup. (a) Pivot point for the tool and the laser, (b) Pivoting tool used to determine the base frame coordinates of the pivot point. 74
3.12 Error compensation by projection of the laser on the calibration grid. 76
3.13 Minimisation of the measurement error through reconstruction of the ideally spaced grid points. 77
3.14 Pictures of the system in action: (a) automatic laser targeting and (b) automatic zooming of the laparoscope, based on the gaze-targeted stereo reconstruction. 78
3.15 Experimental results: Time to reach each target point from the previous one. 80
3.16 Experimental results: Total task time, total laser activation time, and tissue-to-camera distance. 80

4.1 Rendering of the 6-DoF highly articulated surgical instrument. Reproduced from co-authored publication with permission from Springer [160]. 87
4.2 Actuation unit and articulated instrument section. 88
4.3 Overview of the proposed adaptive motion scaling system. 89

4.4 Micro-IGES instruments mounted on the robotic arms. 90
4.5 Functional overview of the complete system. Communication and sensor data paths are marked with arrows. 91
4.6 Schematic representing the relationships between the relevant reference frames of the system. 92
4.7 Illustration of the kinematic chain of the virtual 9 DoF robotic manipulator. Featuring dependent joints in neck 1 and neck 2, and a wrist with 2 DoF, where the pitch is actuated by q_7, the yaw is the mean of q_8 and q_9, and the gripper DoF is represented by the angle difference of q_8 and q_9. 93
4.8 Robotic instruments in parallel configuration as used in single port access surgery. 94
4.9 Dexterity $\mathcal{D}$ analysis of the virtual instrument, using 1×10^9 random robot configurations. Red: high dexterity, blue: low dexterity. 95
4.10 Manipulation restriction based on dexterity measure, green: permitted workspace, blue: reachable workspace. 96
4.11 Tool tracking. Left: temporal feature tracking and bounding boxes. Right: salient image gradients. 99
4.12 Input data flow for the generation of the scaling factor. 101
4.13 Evolution of scaling coefficients from input data. Top: from left to right, images of a ring transfer from the right instrument to the left instrument in chronological order. The red dot represents the gaze point. a) Raw X-axis gaze pixel values (blue), filtered X-axis pixel values (red), and left instrument end-effector X-axis pixel values (green). b) validity of fixation data (fixation identified: value of 1, otherwise 0), c) k_d (blue), k_a (magenta), k_s (red), and k_{as} (green). d) V_{hand} X-axis value in mm/s (blue). 103
4.14 Illustration of the different transfer points for the adaptive motion scaling user study. 105
4.15 User study comparing task execution times and number of times the system was clutched across nine participants. 106
4.16 Heatmaps of gaze fixations during a ring transfer task. Left: without adaptive scaling. Right: with adaptive scaling. 107

5.1 Functions drawn from a zero-mean Gaussian process. The dashed red line represents the mean. The green area represents 2 standard deviations from the mean prior. 115
5.2 Functions drawn from a zero-mean Gaussian process posterior. The dashed red line represents the posterior mean, the dashed blue line represents the unknown objective function from which the observations (indicated by black circles) are drawn. The green area represents 2 standard deviations from the posterior mean. . . 117
5.3 Functions drawn from a zero-mean Gaussian Process with an SE kernel of lengthscale $l = 0.2$ (top), and $l = 0.03$ (bottom). 119
5.4 1D Bayesian optimisation. Top: blue curve: true objective function, red curve: predicted mean, Green patch: area representing plus or minus 2 standard deviations from the mean. The bottom blue curve represents the values of the EI acquisition function. The red vertical line shows the optimal next point to sample suggested by the EI acquisition function. 125
5.5 2D Bayesian optimisation of an unknown objective function (top), using a graded approach (left), and a preference approach (right). . 126
5.6 Overview of the bimanual master-slave system comprising: 2 Omega.7 devices (1) and (2), a Tobii eye-tracker (3), two actuation units (4) and (5) with their respective instruments, and an Olympus stereoscope (6). Three Kuka iiwa robotic arms are visible in the background. 128
5.7 Ring transfer task 131
5.8 optimisation study performance results 132
5.9 Coefficient progress for the graded approach. Blue: normalized distance to the optimal parameter set. Red: negative of grade given. The objective distance refers to the normalized Euclidean distance between the current parameter vector and the final parameter vector. 133

5.10 Coefficient progress for the preference approach. Blue: normalized distance to the optimal parameter set. Red dot: sample was not preferred to the previous best sample. Green dot: sample was chosen as the new best sample. The objective distance refers to the normalized Euclidean distance between the current parameter vector and the final parameter vector. 134

6.1 Illustrations of the simulator (a) without augmented reality, (b) with augmented reality displaying a small tumour. 140
6.2 System overview. 141
6.3 Illustration of the three augmented reality display modes with the simulator view used in the tumour contour marking task. (a) No augmented reality. (b) Full augmented reality. (c) Partial augmented reality. 143
6.4 Definition of distances used as inputs in the augmented reality display behaviour model. 1 represents the projection of the tumour centre on the brain tissue surface. 2 represents the suction tool end-effector. 3 represents the 2D gaze point in screen coordinates. 4 represents the cutting tool end-effector. 5 represents the point on the tumour contour closest to the cutting tool end-effector. 144
6.5 Multi Gaussian process model overview. 147
6.6 Gaussian process voter agents for three observations. No hyperparameter optimization. 148
6.7 Ground truth and result of the voting process for the Gaussian process voter agents shown in Figure 6.6. 148
6.8 Gaussian process voter agents for twenty observations. No hyperparameter optimization. 149
6.9 Ground truth and result of the voting process for the Gaussian process voter agents shown in Figure 6.8. 150
6.10 Zoomed out view of the Gaussian process voter agents for twenty observations. No hyperparameter optimization. 151
6.11 Gaussian process voter agents for twenty observations, with hyperparameter optimization. 153

6.12 Ground truth and result of the voting process for the Gaussian process voter agents shown in Figure 6.11. 154
6.13 Visualisation of the uncertainty during model training. The uncertainty is visualised through the colour of the gaze point, located at the centre of the light blue circle. The light blue circles are displayed purely to draw the reader's attention to the correct points in the image. (a) Low uncertainty: the model has been trained to recognize the context of the cutting motion. (b) High uncertainty resulting from an unusual scene configuration. 155
6.14 Visualisation of the marking of the tumour contour (a) for the partial augmented reality display mode, (b) for the full augmented reality display mode. 157
6.15 Visualisation of bleeding spots (in red, labelled 1 and 3 in the figure), and benign spots (in white, labelled 2 in the figure). The light blue circles are displayed purely to draw the reader's attention to the correct points in the image. The spots pulsate according to their own timers, growing larger and shorter periodically. Their shapes differ slightly but they are all roughly circular. 158
6.16 Results of the user study. 160

7.1 From large to small: surgical robots at different scales. (a) Da Vinci XI® and surgical console, ©Intuitive Surgical (b) Micro-IGES single port system, ©2017 IEEE [160] (c) EurEyeCase hands-on robot for eye surgery [189] (d) Hand-held instrument for confocal laser endomicroscopy, ©2011 IEEE [190] (e) Hand-held haptic device for neurosurgery, reproduced with permission from Springer [187]. 165
7.2 Left: surgical setup under a microscope with a blunt dissector with force feedback (left hand), and forceps (right hand) Right: design of the blunt dissector 167
7.3 High-level system overview. 168

7.4 Illustration of the effect of the main visual force feedback parameters: (a) small radius, (d) large radius; (b) low blending coefficient, (e) high blending coefficient; (c) low IR detail level, (f) high IR detail level. 169
7.5 From top to bottom: gradual increase in forces exerted; Left: with plastic membrane to simulate the dura; Right: without the plastic membrane. 171
7.6 Example of the visual force feedback using in vivo data from a porcine brain dissection task. 172
7.7 Real-time augmented reality pipeline. The dotted line represents a similar line of operations being performed for the right stereo image as for the left image. 174
7.8 Stereo off-axis frustum with converging focal planes. The left-shifted frustum (corresponding to the left camera) is shown in blue, and the right-shifted frustum (corresponding to the right camera) is shown in red. The black frustum represents a baseline symmetrical frustum, and is used for geometrical purposes. 176
7.9 Propagation of displacements in the brain for a surface deformation of 2mm. (a) Section of the model. (b) Map of displacements in the section. 180
7.10 Displacements relative to depth (a) Raw data (1 out of 5 points) and polynomial fitting. (b) Comparison between the generalised model and the polynomial fitting. 181
7.11 (a) Tetrahedral mesh of the brain (white), with the tumour (red). (b) Display of the boundary conditions for the FEM simulations. The static regions are in pink, and the dynamic regions are in orange. The red dots represent the nodes on which different forces will be applied. Picture from the co-authored publication [194]. . . 182
7.12 (a) Estimated 3D structure before tool-tissue interaction. (b) Estimated 3D structure during tool-tissue interaction. (c) Deformation heatmap. The colorscale units are in millimetres. Figure re-used from Springer open access co-authored publication [195]. 184
7.13 Illustrations of the three user studies. (a) Retraction study. (b) Dissection study. (c) Depth study. 184

7.14 Primary outcomes of the retraction study. 185
7.15 Forces exerted by the participants during the retraction study. Blue: forces exerted without visual force feedback. Red: forces exerted with visual force feedback. Green: target force threshold. 186
7.16 Primary outcomes of the dissection study. 187
7.17 Primary outcomes of the depth study. 189

A.1 Illustrations of a six-point thin plate spline modelling visualised using TPSDemo[1]. (a) Regularization coefficient $\lambda = 0$. (b) Regularization coefficient $\lambda = 0.575$. 204

B.1 Pixel error obtained using a sequence of gaze points and, from left to right: the latest TPS model, i.e the one generated from that sequence of points; the TPS model generated by the initial calibration procedure; and the TPS model generated by the previous calibration procedure. 207
B.2 Pixel error obtained using a sequence of gaze points and the TPS model generated by the previous calibration procedure for, from left to right: the first two tasks; and the last task. 208

List of Tables

2.1 Surgical robot control interfaces by system, part 1. 14
2.2 Surgical robot control interfaces by system, part 2. 15
2.3 Recent data glove and virtual reality system interfaces with potential for surgical applications. 20
2.4 Summary of advantages and drawbacks of different master-slave control interfaces. The *Gesture* category summarizes visual and haptic gestures, as well as verbal commands. *: limited force feedback capabilities. 27
2.5 Summary of advantages and drawbacks of different augmented reality display interfaces. 1: can be used with an external MIS camera feed, but the transparency is then only a drawback. 2: depends if used with a head-mounted camera video-stream or not. 49

3.1 Criteria for the eye data analysis. 61
3.2 Mean, median, and statistical significance results from the laser ablation study. 81

4.1 Adaptive scaling parameters. The pixel values for d_{pmax} are for a 1920x1080 monitor resolution. The parameters in red are used to tune the system. 102
4.2 Mean, median, and statistical significance results from the ring transfer study. 106

5.1 Mean and median results from the blind study comparing the different optimisation approaches. 132

5.2 p values from Wilcoxon ranksum tests for the different optimisation approaches. Left: values comparing the number of times the system was clutched. Right: values comparing task completion times. 133

6.1 Mean, median, and statistical significance results from the contour marking study. 160

7.1 Mean, median, and statistical significance results from the visual force feedback retraction study. 185
7.2 Mean and median results from the visual force feedback dissection study. 188
7.3 Mean and median success rates from depth study. 189
7.4 p values from Wilcoxon ranksum tests for the depth study. 190

Abbreviations

ARD Automatic Relevance Determination

CDF Cumulative Distribution Function

CPU Central Processing Unit

CT Computed Tomography

DMA Direct Memory Access

DoF Degree of Freedom

EI Expected Improvement

EM Electromagnetic

FEM Finite Element Modelling

GP Gaussian Process

GPU Graphics Processing Unit

HD High Definition (1920x1080 resolution)

HMD Head-Mounted Display

iid Independent Identically Distributed

IR Inverse Realism

MAP Maximum a Posteriori

MB Megabyte

MIS Minimally Invasive Surgery

MRI Magnetic Resonance Imaging

NPR Non-Photorealistic Rendering

OpenCL Open Computing Library

OpenGL Open Graphics Library

OR Operating Room

PDF Probability Density Function

PI Probability of Improvement

RAM Random Access Memory

RBF Radial Basis Function

RCM Remote Centre of Motion

SE Squared Exponential (referring to the kernel)

SVM Support Vector Machine

TPS Thin Plate Spline

UDP User Datagram Protocol

1

Introduction

Surgery is challenging. Accessing the operating site, administering the surgical treatment, and concluding the procedure all carry important risks and require highly skilled personnel to perform. Due to the limitations of surgical tools, surgery is a delicate balancing act between minimising patient trauma and maximising the effectiveness of the procedure. Open surgery approaches provide the most direct means of reaching the operating site, but do so at the cost of substantial patient trauma. This trade-off can result in procedures being postponed or cancelled as the risks to the patient outweigh the expected improvements in quality of life. The advent of minimally invasive surgery (MIS) has significantly improved this trade-off, allowing surgeons to perform what were previously highly invasive open surgery procedures using only a few small incision points. This progress was enabled by the development of MIS instruments such as laparoscopic forceps and cameras capable of being used through surgical trocars, allowing the surgeon to operate inside the human anatomy without having a direct line of sight to the operating site. However, current MIS instruments suffer from severe limitations in terms of dexterity. Constrained by the trocar position, laparoscopic instruments typically consist of a long rigid shaft where the only additional degree of freedom available is a mechanically transmitted rotation along the instrument shaft. Combined with other ergonomic limitations such as the need for a surgical assistant to operate the camera, these constraints significantly limit the scope of procedures that can benefit from MIS approaches. In particular, surgery in long and narrow anatomical pathways such as in the throat or the colon, or where the operative

workspace is highly restricted such as in keyhole neurosurgery, can all be particularly challenging to perform using MIS instruments.

To tackle these challenges, technological efforts in the last decades have focussed on the development of robotic systems for surgery. These systems allow for the design of highly dexterous surgical instruments possessing a large number of degrees of freedom in a small form factor. Wristed instruments mounted on robotic arms, for example, allow surgeons to recover a substantial part of their natural dexterity while performing minimally invasive approaches. Other systems such as robotic endoscopes or snake-like robots allow surgeons to navigate deep inside the patient anatomy through natural orifices. Additionally, these robotic systems provide a natural platform for the aggregation and display of all the information available in the operating room, such as camera feed, image guidance, system and patient status, or force feedback.

While surgical robots provide enhanced capabilities to the surgeon, their additional complexity also introduces new requirements in terms of usability. Unlike traditional surgical tools or MIS instruments, robotic systems typically require dedicated interfaces to control their different components. Highly articulated instruments composed of multiple joints, for example, cannot be expected to meaningfully improve surgical performance if the surgeon is required to control each joint individually. Likewise, a robotic assistant can be counter-productive if it is more difficult for the surgeon to communicate their commands to the robot than to a human assistant. It is entire possible to design robotic systems that possess extremely valuable properties for surgery, while at the same time being completely unusable in practice.

Humans are physically limited in how they can interact with, and operate a robotic system. In addition to those physical limitations, only a finite amount of information can be relayed simultaneously to the surgeon before they saturate. Robotic systems on the other hand, are only limited in their complexity and capabilities by the technology available at the time of their design. To bridge this gap, the complexity of the robotic system must be concealed by the human-robot interface. The more complex the system, the greater the role of the interface.

1.1 Research Objectives

While surgical robot systems have been the focus of substantial research over the last decades, their corresponding human-robot interfaces have not witnessed as much attention. Recent systems as varied as multi-arm laparoscopic robots or concentric tube systems all mainly use master-slave interfaces. For the most part, these interfaces operate similarly to those used in master-slave surgical robots developed in the 1990s, and suffer from the same underlying limitations such as workspace constraints resulting in the need for clutching. Similarly, surgical navigation information is still mainly visualised through separate external displays. Better-integrated display solutions such as augmented reality have been investigated, but also suffer from limitations such as obstruction of the visual field.

A common issue with the aforementioned robot control and visualisation interfaces is their lack of ability to adapt to the current context, instead using a "one size fits all" approach. Master-slave interfaces with a static motion scaling coefficient, for example, need to compromise between the ability to perform high precision motions or to cross the surgical workspace quickly. Likewise, augmented reality interfaces controlled with an on/off switch either substantially obstruct the surgical view, or do not provide image guidance.

In order to alleviate these issues, new methods for human-robot interfacing can be developed making use of scene context to adapt system behaviour. This thesis explores the design of such context-aware interfaces for robot control and surgical navigation. By exploiting natural human behaviour through gaze tracking, and using scene information such as instrument and anatomy tracking, the desired system behaviour can be inferred and the system adapted accordingly. A note should be made here regarding the scope of two terms used throughout this thesis: the *surgical context*, and the *user intention*. In this work, the surgical context considered is limited to the surgical scene, i.e. the state and position of the anatomy and of the surgical instruments during the procedure. Furthermore, the user intention considered is limited to the direct motion or surgical subtask the surgeon is trying to execute with the instruments at any given moment (e.g. reaching motions, grasping). However, the surgical context could be broadly defined to include any factor that has an impact on the outcome of procedure. This could include factors directly related to the procedure, such as the current surgical phase

or pre-operative plan, but could also include indirectly related information. The patient's medical records, the surgeon's level of fatigue, or the level of expertise of the surgical staff all play a significant role in the outcome of the operation. Likewise, the user intention can encompass higher-order concepts than direct surgical subtasks, such as completing particular surgical steps or phases. While these factors can directly impact the surgeon's behaviour during the operation, such as the impact of fatigue on eye tracking behaviour, it is necessary to limit the scope of these considerations for the purpose of this work. Accordingly, the methods presented in this work aim to fit generic surgical contexts in which "standard" conditions could be observed (e.g. no extreme fatigue or unusual operating circumstances). The aims of this thesis are to address the following challenges:

- *Implicit Interfaces:* Develop methods for human-robot interfaces that work implicitly, where the system behaviour is adapted automatically from scene context without requiring explicit input from the surgeon.
- *Robot Control:* Propose implicit control interfaces for the surgical camera and robotic instruments, allowing improved control without relying on an assistant or explicit commands. In particular, this thesis aims to reduce the control issues caused by workspace limitations of master devices, a long-standing problem in surgical robotics.
- *Surgical Navigation:* Propose implicit augmented reality display methods able to maximise the usefulness of the image guidance information, while minimising the visual obstruction to the surgical view.

The thesis is structured as follows:

Chapter 2 presents a review of existing human-robot interfaces. Control interfaces are reviewed first, including a review of master devices and their limitations as well as gesture-based and voice-controlled approaches. This is followed by a dedicated review of gaze-contingent approaches for robotic control in surgery. A case-study application for the gaze-control of a laparoscope is presented, highlighting the usefulness of gaze data for inferring user intention. Finally, a review of augmented reality interfaces for surgical navigation is presented.

Chapter 3 introduces a framework for the implicit control of the surgical camera and a focussed energy delivery system. A noise-resistant interaction detection scheme is presented, allowing activation and control of the system through natural gaze behaviour. Smooth velocity profiles generated from the surgeon's gaze are used to direct the robotic camera in a safe and intuitive manner. By further using gaze tracking to target localised stereo reconstruction, the system is able to accurately determine the user's gaze point coordinates in 3D space. This is used to perform automatic targeting of the focussed energy delivery system, as well as automatic zooming of the surgical camera so that the optimal camera-to-tissue viewing distance is always maintained. The calibration methodology used to obtain millimetre precision for the laser targeting without the aid of visual servoing is described. Finally, a user study validating the system is presented, showing clear improvement with median task completion times under half of those of a manually controlled robotic system.

Chapter 4 presents a bi-manual system using novel self-triangulating 6-degrees-of-freedom (DoF) tools through a flexible elbow, which are mounted on robotic arms. The control scheme for the resulting 9-DoF system is detailed, with particular emphasis placed on retaining maximum dexterity close to joint limits. Most significantly, this chapter introduces the novel concept of gaze-assisted adaptive motion scaling. By combining eye tracking with hand motion and instrument information, the system is capable of inferring the user's destination and modifying the motion scaling accordingly. This safe, novel approach allows the user to quickly reach distant locations while retaining full precision for delicate manoeuvres. The performance and usability of this adaptive motion scaling is evaluated in user studies, showing a clear improvement in task completion speed and in the reduction of the need for clutching.

The adaptive motion scaling presented in chapter 4 possesses a number of core parameters controlling the robot behaviour that are manually tuned. As the adaptive motion scaling dynamically impacts the user's control of the system, it is important that these parameters be selected appropriately to keep the system behaviour as intuitive as possible. **Chapter 5** presents a Bayesian approach to the problem of optimising the adaptive motion scaling in a user-specific manner. Two Bayesian optimisation methods are studied: one in which users are asked to grade robot

behaviour for a given set of parameters, and one where only preference relative to other parameter sets is expressed. The performance of these optimisations is evaluated in blind, randomised user studies, demonstrating that the optimised parameters result in more intuitive system behaviour and better performance relative to the manually tuned ones.

Expanding on the gaze-contingent methods for intention recognition developed in the previous chapters, **Chapter 6** presents a novel approach to automatically adapt the augmented reality display mode based on the context of the surgical scene. Using gaze tracking in conjunction with information from the surgical instruments and the registered anatomy, a multi Gaussian process model is designed to infer the desired augmented reality display mode at any point during the procedure. Additionally, a new augmented reality display mode is introduced in this model, taking advantage of the context information to only display a partial view of the augmented reality when relevant. To validate the presented approach, a detailed simulation of a neurosurgical tumour contour marking task is designed. User studies are conducted to demonstrate the usefulness of the proposed approach compared to manually controlled augmented reality, showing significant improvements in the ability to detect bleeding as well as in task completion time.

Exploring the future of miniaturised robots in surgery, **Chapter 7** studies how the previously discussed context-based augmented reality approaches can be adapted to hand-held robotic instruments. Using a microsurgical blunt dissector equipped with force-sensing capabilities, a novel method simultaneously combining visual force-feedback and image guidance information is presented. Visual force feedback is displayed by modulating an augmented reality overlay. Additionally, critical anatomy beyond the exposed tissue surface is revealed using the augmented reality overlay when the user is exerting large forces within their proximity. The force information is further used to improve the depth perception of the augmented reality by displacing the virtual elements based on the forces exerted. Detailed user studies are performed to assess the efficacy of the proposed method, demonstrating its usefulness for visual force feedback and the importance of motion cues for improved depth perception.

Finally, **Chapter 8** concludes by summarising the key results and findings of this

thesis, and discussing current limitations and future research opportunities.

1.2 Original Contributions

The key contributions of this work include:

- The development of an implicit interaction detection scheme to control a surgical camera, exploiting the synergy between eye tracking and stereo reconstruction to perform high-accuracy 3D gaze tracking, and enabling automatic zooming and laser targeting.
- The design of a highly dexterous robotic system combining the dexterity of single-port instruments with the advantages of external robotic arms.
- The development of a novel gaze-assisted adaptive motion scaling scheme, reducing workspace limitations and the need for clutching of master devices.
- The demonstration of the usefulness of Bayesian optimisation techniques to tune human-robot interfaces, comparing graded and preference-based approaches, and highlighting the pitfalls of relying on manual optimisation.
- The development of a low-latency real-time rendering pipeline for stereo augmented reality.
- The design of a novel gaze-assisted approach to automatically select the optimal augmented reality display mode based on scene context, and the development of a detailed surgical simulator to assess the efficacy of this approach.
- The development of a novel approach combining visual force feedback and image guidance for hand-held microsurgical instruments equipped with force sensing.
- The development of force-based displacement modelling for augmented reality to improve depth perception via motion cues.

1.3 Publications

The following first-author publications are directly related to the work conducted in this thesis:

- **G.Gras**, G-Z.Yang. "Context-Aware Modelling for Augmented Reality Display Behaviour." Robotics and Automation Letters (RA-L) 2019.
- **G. Gras**, C.A. Seneci, P. Giataganas, G-Z. Yang. "Gaze-Assisted Adaptive Motion Scaling Optimization using Graded and Preference Based Bayesian Approaches." IEEE International Conference on Robotics and Automation (ICRA) 2018.
- **K. Fujii - G. Gras**, A. Salerno, G-Z. Yang. "Gaze gesture based human robot interaction for laparoscopic surgery." Medical Image Analysis (MedIA) 2018. **Joint first author**
- **G. Gras**, K. Leibrandt, P. Wisanuvej, P. Giataganas, C.A. Seneci, M. Ye, J. Shang, G-Z. Yang. "Implicit Gaze-Assisted Adaptive Motion Scaling for Highly Articulated Instrument Manipulation." IEEE International Conference on Robotics and Automation (ICRA) 2017.
- **G.Gras**, G-Z.Yang. "Intention Recognition for Gaze Controlled Robotic Minimally Invasive Laser Ablation." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2016.
- **G. Gras**, H.J. Marcus, C.J. Payne, P. Pratt, G-Z. Yang. "Visual force feedback for hand-held microsurgical instruments." International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2015. **Single track oral talk presentation**

The following co-authored publications are relevant to the work presented in this thesis:

- B-R. Pierre, **G. Gras**, K. Leibrandt, P. Wisanuvej, A. Schmitz, C.A. Seneci, G-Z. Yang. "The i^2 Robotic Platform for Endoscopic Surgery." Annals of Biomedical Engineering (ABME) 2018.

- B-R. Pierre, K. Leibrandt, **G. Gras**, P. Fraisse, A. Crosnier, G-Z. Yang. "Inverse Kinematics Control Methods for Redundant Snake-Like Robot Teleoperation during Minimally Invasive Surgery." Robotics and Automation Letters (RA-L) 2018.

- M. Tonutti, **G. Gras**, G-Z. Yang. "A machine learning approach for real-time modelling of tissue deformation in image-guided neurosurgery." Artificial Intelligence in Medicine 2017.

- P. Wisanuvej, **G. Gras**, K. Leibrandt, P. Giataganas, C.A. Seneci, J. Liu, G-Z. Yang. "Master Manipulator Designed for Highly Articulated Robotic Instruments in Single Access Surgery." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017.

- C.A. Seneci, **G. Gras**, P. Wisanuvej, J. Shang, and G-Z. Yang. "3D printing of improved needle grasping instrument for flexible robotic surgery." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2017.

- K. Leibrandt, P. Wisanuvej, **G. Gras**, J. Shang, C. Seneci, V. Vitiello, G-Z. Yang. "Effective Manipulation in Tight Spaces of Highly Articulated Robotic Instruments for Single Access Surgery." Robotics and Automation Letters (RA-L) 2017.

- J. Shang, V. Vitiello, K. Leibrandt, C. A. Seneci, P. Wisanuvej, J. Liu, P. Giataganas, **G. Gras**, J. Clark, A. Darzi, G-Z. Yang. "A Single-Port Robotic System for Transanal Micro-Surgery – Design and Validation." Robotics and Automation Letters (RA-L) 2017.

- S. Giannarou, M. Ye, **G. Gras**, K. Leibrandt, H.J. Marcus, G-Z. Yang. "Vision-based deformation recovery for intraoperative force estimation of tool–tissue interaction for neurosurgery." International journal of computer assisted radiology and surgery 2016. **Best paper award.**

- C.J Payne, **G. Gras**, M.R. Hughes, D. Nathwani, G-Z. Yang. "A handheld flexible mechatronic device for arthroscopy." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2015.

- **G. Gras**, V. Vitiello, G-Z. Yang. "Cooperative control of a compliant manipulator for robotic-assisted physiotherapy." IEEE International Conference on Robotics and Automation (ICRA) 2014. **Best student paper finalist.**

2

Control and Visualisation Interfaces in Robotic Surgery

2.1 Introduction

Human-robot interfaces are a crucial component of surgical robots. These interfaces can be as simple as a few buttons or a monitor, to as complex as entire robots of their own right, and are essential for the surgeon to be able to interact with the robotic system. Two categories of human-robot interfaces are studied in this thesis: interfaces related to the control of the surgical robot, and visualisation interfaces related to surgical navigation. The quality and design choices involved in the making of these interfaces play a key role in the final performance of the entire system. Robot control interfaces are responsible for i) accurately relaying the surgeon's intended commands to the robot, and ii) permitting seamless cooperation between the operator and the robot. Similarly, visualisation interfaces are responsible for presenting information from the system and the surgical scene to the surgeon such that they are accurately understood, and minimizing the distraction caused while doing so. Poorly designed control interfaces can result in robotic systems that are challenging and potentially dangerous to use, even if the surgical robot itself has advanced capabilities. Likewise, poor visualisation interfaces create clutter and can even mislead the surgeon, as well as reducing surgical performance through occlusion of the visual field or causing interruptions of the

surgical workflow.

To take full advantage of modern robotic systems, effective human-robot interfaces must also be designed. As such, the advent of robotics in the operating theatre has also prompted the development of new interfacing techniques. This chapter presents an overview of existing human-robot interfaces in the context of surgery. Robot control interfaces are reviewed first, from the widely-used master consoles to more recent solutions such as gamepads and data gloves. The second section is dedicated to a separate review of gaze-contingent control interfaces in surgery, a technology utilized substantially throughout this thesis for its capacity to help infer user intention. Finally, the third section reviews interfaces related to the visualisation of image guidance information, with a strong focus on augmented reality.

2.2 Human-Robot Control Interfaces in Robotic Surgery

The control interface of a surgical robot consists of a set of components and methods used by the surgeon to control the robot. The function of a robot control interface can be segmented into two processes. An acquisition process, responsible for physically interacting and acquiring information from the surgeon; and a transformation process, responsible for transforming the information obtained by the acquisition process into commands compatible with the surgical robot.

By definition, some hardware components are necessary for the system to be able to interact with a user. The components of the acquisition process will typically present themselves in the form of a device, or a set of devices, without which the system cannot be used. Conversely, the transformation process describes the logic used to control the robot, and can sometimes be performed entirely in software. Both of these components play a fundamental role in how to interact with the robotic system. To provide the reader with a broad overview of human-robot control interfaces in surgery, Tables 2.1 and 2.2 list a number of surgical robots and their respective interface types. Note that the purpose of this list is not to provide an exhaustive catalogue of the different existing surgical robots. Instead, these tables display a selected set of surgical robot systems to provide a compre-

hensive overview of the different human-robot control interfaces that can be found in commercial and research-prototype surgical robots. For reviews focused on a more exhaustive list of existing surgical robot systems, the interested reader is referred to [1, 2, 3, 4].

2.2.1 Grounded Robotic Haptic Master Interfaces

One of the most widely used control paradigms for surgical robots is the master-slave paradigm. Under this paradigm, surgical robot systems are composed of two main subsystems: a master subsystem which the surgeon uses to issue their commands, and a slave subsystem which physically interacts with the patient following the surgeon's commands. This system architecture presents several key advantages for surgical robots. In such a configuration the slave typically only acts when directly commanded to by the surgeon, allowing the surgeon to always remain in control of the procedure while still benefiting from robotic advantages such as tremor reduction and motion scaling. While it is possible to blur the line between direct master-slave control and semi-automation depending on the type of interface, in this thesis semi-automation is defined as starting from the moment where surgical subtasks more complex than brief motions are automated [5]. The physical separation between the master and slave subsystems also allows for larger and more elaborate human-robot interfaces, whereas otherwise the footprint of those interfaces might prevent their use at the operating table. Additionally, the separation between slave and master can be advantageous to perform long distance teleoperated surgery.

A particular type of master interface that has emerged as highly successful over the last decades are grounded haptic master interfaces. In these systems, the master subsystem is itself a robotic device that the surgeon holds with their hands. These haptic master devices allow a highly accurate capture of the surgeon's hand motions, while also providing the possibility for haptic feedback. The grounded aspect of these systems comes from the design choice of the master interface to be part of a master workstation, or master console, in part due to the size and connectivity requirements of the haptic devices. The highly successful Da Vinci system from Intuitive Surgical, illustrated in Figure 2.1, is an example of such a system. The use of such consoles allows advanced functionality to be integrated

System Name	Institute Company	Country	Interface	Refs.
NeuroArm	University of Calgary	Canada	Haptic master interface. Cartesian-space commands.	[6] [7]
Da Vinci	Intuitive Surgical	USA	Haptic master interface. Cartesian-space commands.	[8]
Raven	Applied Dexterity	USA	Haptic master interface. Cartesian-space commands.	[9] [10]
Senhance	Transeterix	USA	Haptic master interface, Cartesian-space commands. Force feedback with force sensing at tool base. Eye tracking for camera control.	[11] [12]
ZEUS/AESOP	Intuitive Surgical	USA	Haptic master interface, Cartesian-space commands. Voice controlled camera-holder.	[13]
Monarch	Auris Health	USA	Haptic master interface using a gamepad controller. Commands presumably in Cartesian-space.	[14]
Sensei X	Hansen Medical	USA	Haptic master interface, Cartesian-space commands. Includes force feedback.	[15]
MMS-II	T.U. Munich	Germany	Haptic master interface using a joystick controller. Cartesian-space commands.	[16]
FLEXMIN	T.U. Darmstadt	Germany	Haptic master interface, joint-space commands.	[17]
Micro-IGES	Imperial College London	UK	Haptic master interface, joint-space commands.	[18]
Niobe	Stereotaxis	USA	Mouse and keypad master interface.	[19]
Steady hand	Johns Hopkins University	USA	Hands-on robot.	[20] [21]
Preceyes Surgical System	Preceyes B.V.	Netherlands	Haptic master interface with kinematic matching between master and slave devices.	[22]

Table 2.1: Surgical robot control interfaces by system, part 1.

System Name	Institute Company	Country	Interface	Refs.
Acrobot	Stryker	USA	Hands-on robot.	[23]
Robodoc	Integrated Surgical Systems	USA	Automated: pre-operative computer interface.	[24] [25]
Cyberknife	Accuray Incorporated	USA	Automated: pre-operative computer interface.	[26] [27]
MicroHand A	Tianjin University	China	Electromagnetically-tracked master interface. Cartesian-space commands.	[28]
i2Snake	Imperial College London	UK	Electromagnetically-tracked master interface. Cartesian-space commands.	[29] [30]
Unnamed G.I robot	Nanyang Technological University	Singapore	Exoskeleton master interface. Joint-space commands.	[31]
Unnamed endonasal robot	Vanderbilt University	USA	Master interface based on hand tracking.	[32]
Unnamed radiofrequency ablation robot	National University of Singapore	Singapore	Master interface based on hand tracking and hand gestures.	[33]
Robotic Scrub Nurse	Purdue University	USA	Semi-automated control interface based on hand gestures.	[34]
Unnamed MIS robot	Osaka Institute of Technology	Japan	Master interface based on foot gestures.	[35]
ViKY	INSERM	France	Master interface based on voice control.	[36]

Table 2.2: Surgical robot control interfaces by system, part 2.

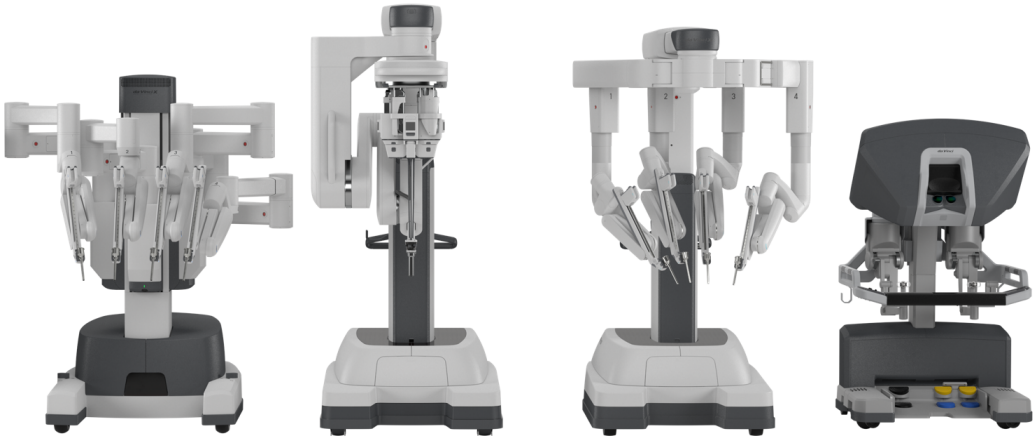


Figure 2.1: From left to right: Da Vinci X®, Da Vinci SP®, Da Vinci XI®, and surgeon console. ©Intuitive Surgical.

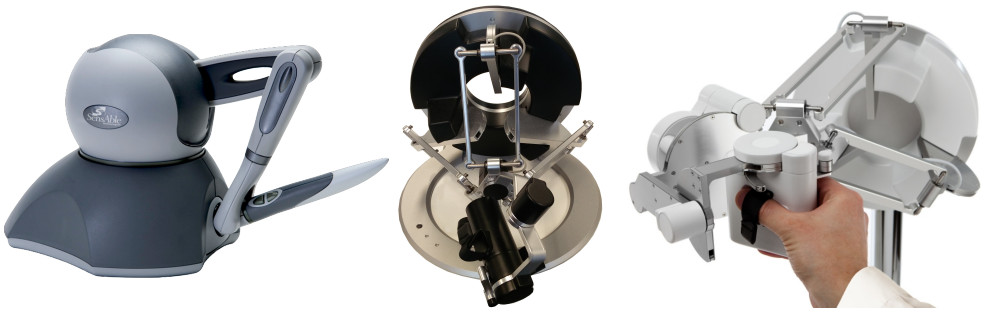
in the master interface, such as an immersive stereo vision and additional control options via pedals and buttons.

Several key design choices regarding the master haptic device impact how the surgeon controls the slave robot. While these can depend strongly on the type of master interface and slave robot, three main categories can be distinguished: position-controlled haptic interfaces, joint-controlled haptic interfaces, and hybrid position-velocity controlled haptic interfaces.

Position-Controlled Haptic Interfaces

Haptic devices in master-slave systems are used to control the surgical robot. In a large number of systems [13, 8, 6, 12, 10], the robotic instruments are controlled in Cartesian space, by specifying a desired position and orientation for the end-effector of each surgical instrument. A main advantage of this approach is that the master haptic device can have any architecture as long as it is able to output an end-effector pose. This means that the master haptic device can be designed to be ergonomically friendly, independently of the robotic instruments. Additionally, this makes it possible for different master haptic devices to be used interchangeably with the same slave system. Figure 2.2 shows three widely used haptic master devices that output commands in Cartesian space.

The Touch Haptic Device consists of a pen-like end effector that can be grasped by the user. Two buttons allow the user to relay additional commands to the slave



(a) (b) (c)

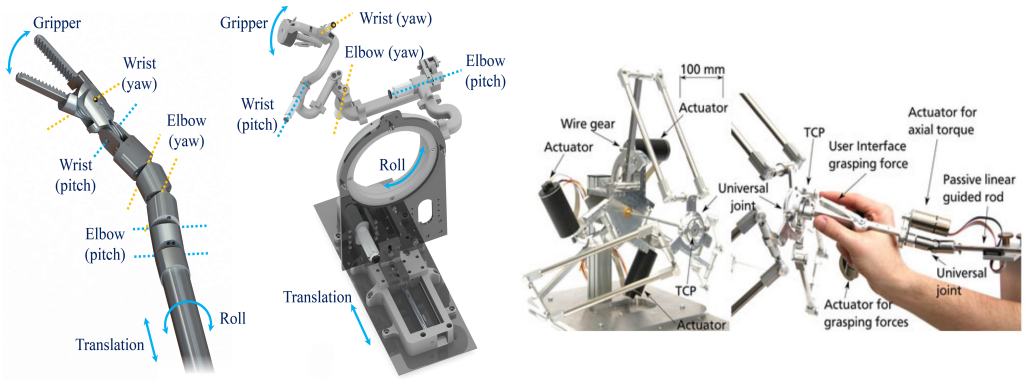
Figure 2.2: Illustrations of three haptic master devices. (a) Touch™ Haptic Device, ©3D Systems. Formerly Phantom Omni®, ©SenAble. (b) Omega.7®and (c) Sigma.7®, courtesy of Force Dimension, Switzerland.

system. The Omega.7 and Sigma.7 both possess a gripper-like end effector, more similar to surgical grippers. The Omega.7 allows for force feedback on all translational degrees of freedom, while the Sigma.7 allows for force feedback on all degrees of freedom, including the rotation.

Joint-Controlled Haptic Interfaces

While position-controlled haptic interfaces are highly versatile and ergonomic, they suffer from some limitations. A key limitation of these interfaces stems from the generic nature of the commands they issue. Cartesian end-effector commands are useful for fully describing the desired pose of the surgical instruments' end-effectors, however they cannot be used to specify additional information regarding the desired robot configuration. This can be disadvantageous in redundant robotic systems, where the additional degrees of freedom cannot be controlled directly by the user. Instead, the user must rely entirely on the programmed kinematic behaviour to select the appropriate joint positions. Additionally, while haptic devices such as the Omega.7 are able to relay haptic feedback to the user, that haptic feedback is also constrained to be generic, such as forces applied on the end-effector.

A solution to these issues is to design a master haptic interface that sends commands in joint space. This approach imposes strict limitations on the structural design of the master interface, as it must sufficiently match the structure of the controlled instruments to be intuitively usable. Figure 2.3 shows two such interfaces, used to control redundant surgical instruments for natural orifice surgery.



(a) (b)

Figure 2.3: Examples of haptic master devices that generate commands in joint-space. (a) Master device using a serial-link structure matching the kinematic structure of the controlled instruments [18], ©2017 IEEE. (b) Master device with a parallel structure [17], ©2016 IEEE.

These devices allow the robotic instruments to be controlled without requiring an inverse kinematic solver, as the joint values of the master device are mapped to the joint values of the instruments. This makes it possible for the user to more easily understand the kinematic configuration of the robotic instruments. Additionally, joint-specific force feedback can be used to convey the joint limits of the robotic instrument to the user, a valuable property for highly dexterous minimally invasive robots working in tight spaces.

The main limitation of these interfaces is that their ergonomics are mainly dictated by the kinematic structure of the robotic instruments, making them impractical to use in a number of cases. Additionally, all instruments used during a procedure must have a similar kinematic structure, as otherwise the master device will not be able to control the robot properly after instrument-swapping.

Hybrid Position-Velocity Haptic Controlled Interfaces

A final key limitation of position-controlled haptic interfaces is linked to their restrictive workspace. Due to their rigid design, haptic interfaces have a well defined, and often quite limited, workspace. This workspace dictates how far the user can move their hands while grasping the master device before reaching the limits of how far the device can go. Due to motion scaling, it is entirely possi-

ble for the user to reach the end of the master device workspace before finishing their desired motion. Note that this problem arises in nearly all master-slave haptic devices. Grounded haptic interfaces, however, typically suffer from smaller workspaces than other master-slave interfaces, accentuating this issue.

The most widespread method to address this problem is to use a footpedal, which when pressed allows the user to re-centre the master device in a process called "clutching", similarly to re-centring a computer mouse on a mousepad. However, this process can be highly disruptive, particularly in systems with a large workspace but where really low motion scaling factors are desired. A solution to this problem for position-controlled haptic interfaces consists in defining velocity zones in the master device workspace [37, 38, 39]. When the end-effector of the master device is brought into these velocity zones, the system adds a velocity component to the motion commanded by the master device. As long as the end-effector remains in this zone, even if immobile, the commanded robotic instrument will keep translating in the direction specified by the velocity.

To retain intuitive control of the slave robot, a hybrid position-velocity control with variable stiffness is used for the master device. In the "Bubble" Technique [38], the standard position-controlled mode is used within a large spherical space centred on the master device origin. The space outside of this sphere is a velocity zone, where the velocity is a function of the distance from the sphere edge. The velocity component drops down to zero once the master device end-effector leaves the velocity zone. Additionally, the haptic device exerts a force on the user's hand when inside the velocity zone, pulling them back to the position-controlled workspace with a force direction opposite to the velocity vector. This allows the velocity zones to be perceptible to the user, significantly improving the ergonomics of the interface [37].

The main drawback of this approach, however, is the loss of precision incurred in the velocity zones. Even with smoothly evolving velocity zones and hysteresis around the boundary, it can be challenging for a user to achieve high precision using a hybrid position-velocity haptic interface when working at the edge of the master device workspace, thus not truly resolving the clutching issue. Finally, the "automatic" motion of the system when the master device is immobile in a velocity zone is a break from the master-slave paradigm, introducing an additional risk factor.

2.2.2 Ungrounded Haptic Master Interfaces

System Name	Institute Company	Country	Description	Refs.
Carbon Nanotube Glove	National Institute of Advanced Industrial Science and Technology	Japan	Master interface based on hand tracking and hand gestures.	[40]
CyberGlove CyberTouch CyberGrasp CyberForce	CyberGlove Systems LLC	USA	Gloves for hand motion tracking. CyberTouch includes tactile feedback. Cybergrasp is an exoskeleton glove with grasping haptic feedback. CyberForce is an exoskeleton glove with grasping and full-hand haptic feedback.	[41] [42]
HaptX VR Glove	HaptX	USA	Glove for hand motion tracking. Includes haptic and tactile feedback.	[43]
Manus VR Glove	Manus VR	Netherlands	Glove for hand motion tracking. Includes vibration feedback.	[44]
SteamVR	HTC	Taiwan	Untethered hand-held controller used in the HTC Vive virtual reality system.	[45]
Oculus Rift Controller	Oculus VR	USA	Untethered hand-held controller used in the Oculus Rift virtual reality system	[46]

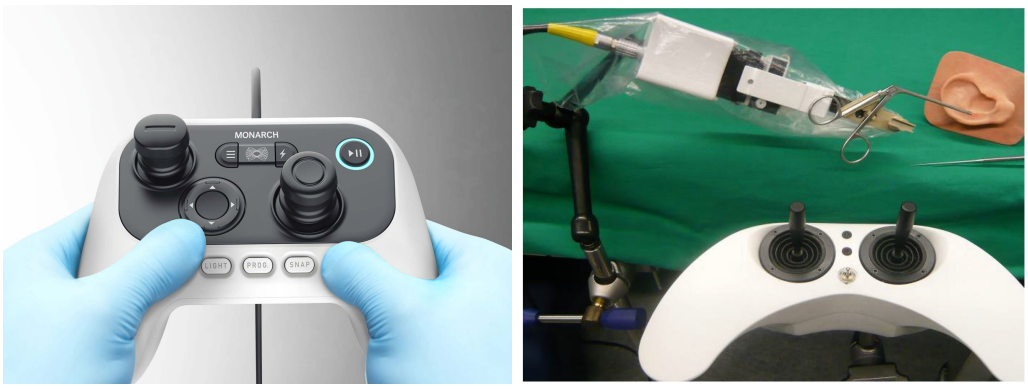
Table 2.3: Recent data glove and virtual reality system interfaces with potential for surgical applications.

While the grounded robotic master interfaces presented in section 2.2.1 have enjoyed widespread popularity for master-slave surgical robots, more recent haptic technologies have emerged as strong alternatives. This includes technologies applied to existing surgical robot systems, such as gamepads, and are also listed in tables 2.1 and 2.2. Other technologies, summarized in table 2.3, present attractive features for use in robotic surgery, but to our knowledge have not yet been tested in non-simulation based surgical applications. These alternative haptic interfaces are reviewed below.

Joysticks and Gamepads

Joysticks and gamepads provide more lightweight, mobile alternatives to traditional grounded haptic interfaces. These haptic interfaces have a much smaller footprint than grounded haptic interfaces, and can be moved around the operating theatre if necessary thanks to unobtrusive tethering or wireless capabilities. While they can provide haptic feedback to the user, this feedback will typically come in the form of vibrations as opposed to targeted forces. The Monarch Platform from Auris Health, Inc. uses a gaming console-like controller illustrated in Figure 2.4(a) to navigate its endoscopy system, while the MMS-II shown in Figure 2.4(b) uses a dual-joystick system assembled from Megatron components (Megatron Elektronik GmbH, Germany).

Due to their design, joysticks and gamepads typically use a velocity-based control to generate commands for the robot. While this means they avoid clutching issues, it also does mean high positional precision can be harder to achieve relative to robotic haptic devices. Due to decades of product-testing in the consumer market, however, the ergonomics of these devices are well understood and compensated for. The key drawback of these interfaces is that both hands are usually required to operate a single joystick/controller, itself designed to control a single device (e.g. left thumb controls the position while right thumb controls the orientation). This prevents such interfaces from being used with systems that require two or more instruments to be controlled simultaneously.



(a) (b)

Figure 2.4: Example joystick/gamepad controllers used in surgical robot systems. (a) Monarch®, ©Auris Health Inc. (b) MMS-II ©2010 IEEE [16].

Inertial and Vision-Based Haptic Interfaces

The emergence of virtual reality as a consumer product has precipitated the development of haptic interfaces used to interact with those systems. Systems such as the Oculus Rift and HTC Vive, illustrated in Figure 2.5, combine infrared tracking with inertial sensors to provide accurate, drift-free tracking of haptic devices.



(a) (b)

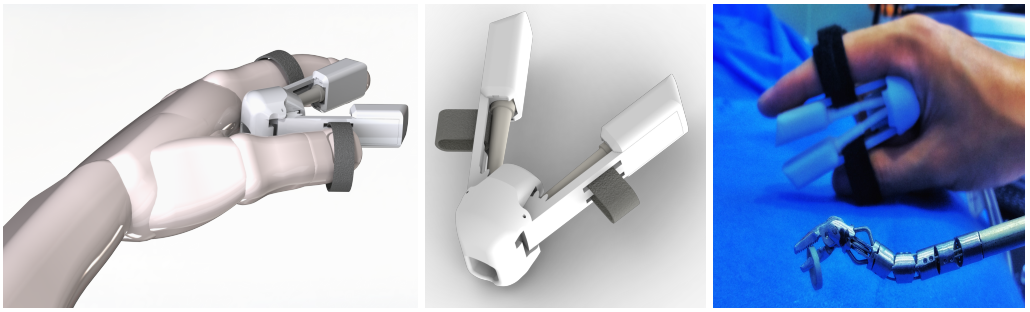
Figure 2.5: Hand-held controllers tracked using inertial sensors and infrared cameras. (a) Oculus Rift®system , ©Facebook Inc. (b) HTC Vive®, ©HTC.

While not as precise as robotic master devices, these devices offer a large workspace and substantial freedom of movement to the surgeon. Their main drawbacks are the requirement for a line of sight to the tracking cameras, and the lack of capability for force feedback other than vibrations. Additionally, prolonged surgeries would require an arm-rest support to prevent exhaustion.

Electromagnetic Haptic Interfaces

Unlike vision-based interfaces, electromagnetic (EM) trackers do not require line-of-sight to their markers. EM markers are lightweight and their small dimensions make them easy to integrate in hand-held devices, as illustrated in Figure 2.6. This makes them an attractive alternative to vision-based haptic interfaces in crowded operating theatres, where maintaining line of sights to visual trackers would be challenging.

Like vision-based and inertial interfaces, EM haptic interfaces do not by default support the possibility for force feedback other than vibrations, and require an arm-rest support for long procedures. Unlike vision-based and inertial inter-



(a) (b) (c)

Figure 2.6: Example of an electromagnetically-tracked master device in the shape of a hand-held gripper, using the Ascension trakStar™ system. (a) and (b): rendering of the master device as used in [29] and [30]. (c): Master device used to control a 6 DoF surgical instrument [47].

faces, however, EM interfaces typically have shorter workspaces and suffer from distortions due to nearby metallic objects.

Data Gloves and Exoskeletons

Data gloves are not new, with models such as the CyberGlove (CyberGlove Systems LLC) having existed since the 1990s. However, they have experienced a recent surge thanks to the growing popularity of virtual reality systems. Current generations of data gloves are capable of precise finger tracking [40] and force feedback [48], as well as tactile feedback for models such as the CyberTouch and HaptX Glove™. Figure 2.7 illustrates different data gloves and exoskeletons.

While exoskeleton interfaces such as the CyberGrasp may seem closer to grounded robotic haptic interfaces than to wireless game controllers, they are classified here as ungrounded due to their wearable nature. In addition to the multiple types of feedback available through these interfaces, one of their main advantages is the capability to track individual fingers, allowing for more advanced modes of interaction with the robotic system. A limitation of such systems is that the setup time for the gloves can be non-negligible. Older models could require substantial calibration procedures, prohibitive for use in surgery. However newer models and the advent of consumer-ready products should improve this, as has been the case for vision-based systems.



(a) (b) (c)

Figure 2.7: Example of data gloves as control interfaces. (a) CyberGlove®used to assess ergonomics in laparoscopic surgery, ©2010 IEEE [42]. (b) CyberGrasp®tested haptic rendering in teleoperation applications, ©2013 IEEE [48]. (c) CyberForce®used for measuring improvement in post-stroke rehabilitation patients, ©2007 IEEE [49].

2.2.3 Gesture and Voice Controlled Interfaces

The precision and safety requirements of surgery make haptic master interfaces particularly well suited to control robotic instruments. Current non-haptic interfaces rely purely on gaze tracking, visual tracking, voice recognition, or other gesture recognition. Eye tracking techniques can be leveraged to great effect in robotic surgery and are discussed separately in section 2.3.

Purely vision-based control interfaces offer similar advantages as hybrid inertial-vision-based haptic interfaces presented in section 2.2.2, while allowing the user's hands to be free if the tracking allows it. In [32], a concentric tube system for endonasal surgery is controlled using a Leap Motion tracker. In [33], hand gestures detected using a Microsoft Kinect are used to control both a robot and an augmented reality display for percutaneous needle insertion. In [34], hand gestures detected by a Microsoft Kinect, illustrated in Figure 2.8, are used to control a robotic nurse.

The key limitation of pure vision-based master interfaces is accuracy. Unlike in hybrid inertial/vision-based haptic interfaces where inertial sensors readings are combined with visual tracking to provide precise position and orientation measurements, any limitation in the visual tracking algorithm or obstruction of the visual field will result in poor interface accuracy. A study conducted in [50] simulated a surgical positioning task comparing the Leap Motion and the Microsoft

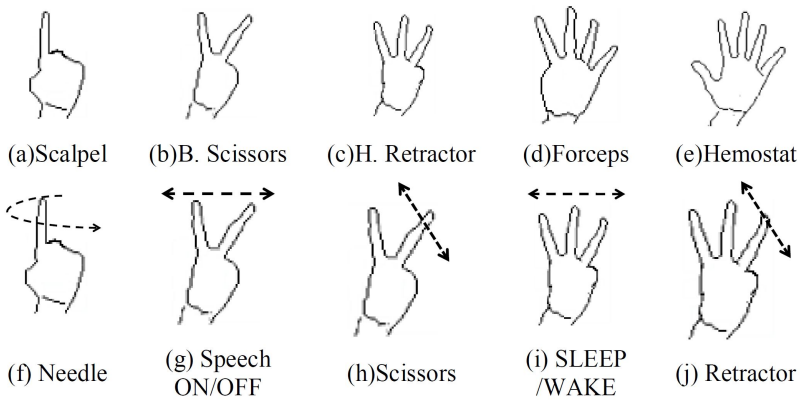


Figure 2.8: Hand gestures used to request surgical tools from a robotic assistant, ©IEEE 2013 [34].

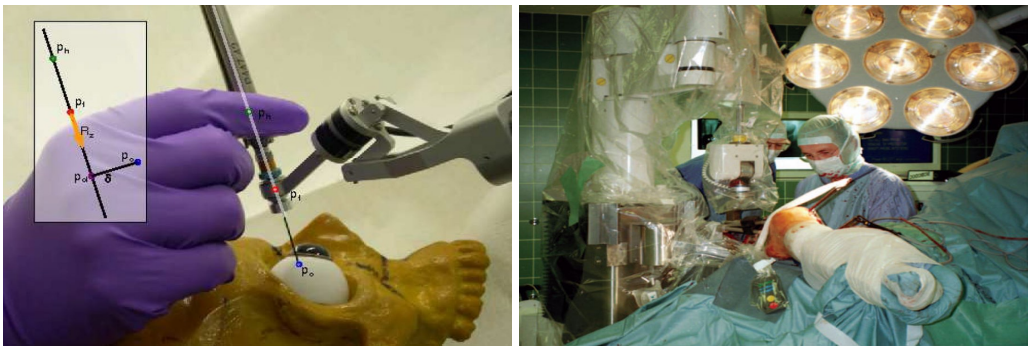
Kinect. The reported conclusion is that, with an average accuracy of 6.2mm, neither system was sufficiently precise or robust over a workspace that would justify their use over a robotic haptic master device. Note that systems relying purely on detected gestures, such as [34], are more concerned with the gesture recognition accuracy than with the tracking precision accuracy. While gesture-based approaches are ideal for numerous high-level commands, they are ill-suited for the real-time positional control of a surgical instrument due to their discrete nature and the need for each gesture to be clearly, individually identified.

Other options include voice-controlled interfaces, such as for camera control in the AESOP [13] and ViKY [36] systems. As with gesture-based approaches, the key limitation of these interfaces is their limited capability for real-time instrument control. While suitable for occasional adjustments of the camera position or angle, these methods are not adapted to the real-time control of a surgical instrument, or fluid camera tracking. As shown in [51], this causes the surgeon to experience a definite loss of comfort and time when relying on these voice-controlled interfaces. Finally, a gesture-based interface using foot motions was developed in [35] to provide continuous traction with a third arm equipped with surgical forceps. While interesting, this approach suffers from the same limitations as mentioned previously, as well as the fact that footpedals are already used extensively in robotic systems.

2.2.4 Hands-On Robots, Automated, and Semi-Automated Interfaces

While master-slave systems make up a large portion of existing surgical robots, numerous other systems exist that do not follow that paradigm. These can be roughly classified between hands-on system, and automated or semi-automated systems. Hands-on robots such as the Steady Hand [20] (Figure 2.9(a)) provide the user with a collaborative robotic interface that assists the motion. The Acrobot surgical system [54] similarly allows a hands-on collaborative interface through the use of virtual fixtures. These systems provide highly useful functionality through collaborative interfaces. The main drawback of these hands-on interfaces is that they are primarily aimed at open surgery procedures, offering limited usefulness for laparoscopic approaches compared to master-slave systems with articulated robotic instruments.

Automated and semi-automated systems such as the ROBODOC surgical system [24] (Figure 2.9(b)) or the Cyberknife [26] also offer useful functionality, for bone drilling and non-invasive radiation therapy respectively. While these systems are highly interesting from a robotic point of view, their human-robot interfaces mainly reduces to the software used to control their automated functionality. Even though these types of interfaces are interesting in their own right, they are considered outside the scope of this work and thus not reviewed.



(a) (b)

Figure 2.9: (a) Steady-hand robot, ©2007 IEEE [52]. (b) ROBODOC surgical system, ©2007 IEEE [53].

2.2.5 Summary

Table 2.4 presents a qualitative summary of advantages and drawbacks of the principal master-slave robot control interfaces reviewed.

Interface Type	Workspace	Movement in the OR	Accuracy	Expressiveness	Force Feedback	Footprint
Haptic Position	- -	✗	+ +	-	✓	- -
Haptic Joint	+	✗	+ +	+ + +	✓	- -
Haptic Velocity	+	✗	+	-	✓	- -
EM Tracking	+ +	+	+	+	✗	+
Visual Tracking	+ +	+ +	- -	-	✗	+ +
Inertial & Visual Tracking	+ +	+ +	+	-	✗	+ +
Gestures	+	+	-	- -	✗	+
Glove	+ +	+	+	+	✓*	+
Exoskeleton	-	- -	+ +	+ +	✓	- -
Joystick	-	-	+	+	✓*	+

Table 2.4: Summary of advantages and drawbacks of different master-slave control interfaces. The *Gesture* category summarizes visual and haptic gestures, as well as verbal commands. *: limited force feedback capabilities.

2.3 Gaze-Contingent Interfaces in Surgery

The precision required to perform surgery makes haptic interfaces particularly well suited for the control of robotic instruments. However, as discussed previously in section 2.2, haptic interfaces present ergonomic limitations, such as the need to clutch due to workspace issues or to control the surgical camera. While some solutions presented previously in section 2.2 have been brought forward to alleviate these issues, such as velocity zones or voice control, these have shown limited effectiveness in practice [51].

An alternative approach consists in tracking the user's gaze to provide new control solutions. Haptic or voice-based interfaces require purposeful, dedicated



(a) (b) (c)

Figure 2.10: (a) Tx300 monitor-mounted eye tracker, ©Tobii Technology. (b) SMI Eye Tracking Glasses ©SensoMotoric Instruments (SMI). (c) Retro-fitted eye tracker for the Da Vinci robot, ©IEEE 2015 [59].

actions to be taken by the user, such as hand motions or uttering vocal commands. Interfaces using information extracted from natural eye behaviour, however, do not require any explicit actions from the user. Natural gaze behaviour constantly explores the visual field and focuses on areas of interest [55, 56]. This allows gaze-contingent human-robot interfaces to infer useful information about the user's intention, and take advantage of that knowledge to direct the robotic system.

While intrusive eye tracking technology exists, such as using lenses [57], most eye tracking systems used in practice rely on vision-based trackers. A typical approach consists in using infrared light to illuminate the eyes, generating reflections on the boundaries of the lens and cornea. Images captured from this process, referred to as Purkinje images, can be processed to detect both the centre of the pupil and the resulting glint. The corresponding vector can then be mapped to a unique gaze direction [58, 59].

To obtain a high accuracy measurement of the user's 2D gaze point on a specific surface, the eye tracking device must be calibrated to that surface. As such, eye tracking systems in which the camera has a static and well calibrated position relative to the display surface are able to yield much higher accuracy than systems where these positions can change between uses. Example eye tracking devices are shown in Figure 2.10. A common configuration is to rigidly mount an eye tracker on a monitor, such as in Figure 2.10(a). The key advantage of this configuration is the reliable calibration of the system, as well as the relative ease of use of this medium in the operating room (OR). Additionally, comparatively

larger eye trackers can be mounted under the monitor, allowing high framerate and a large detection field for the surgeon's eyes. Alternatively, miniaturised eye trackers fitted inside glasses can be used, as illustrated in Figure 2.10(b). This approach allows eye tracking to be used in a completely portable fashion, removing any restrictions on the users' freedom of movement. However, the trade-off for this portability is that the recovered 2D gaze point coordinates are with reference to the image seen on the surface of the glasses. Since images cannot typically be displayed on the surface of the glasses, this approach makes it much more challenging to get high-accuracy gaze information from an interactive display. As a result, this approach is more suitable for applications such as assistive robots [60, 61], where the user is expected to be mobile and eye tracking is used for passive information gathering. Lastly, eye trackers may be directly integrated within the console of a robotic system. This approach has been presented with the Da Vinci console, such as in [58] and [59] (illustrated in Figure 2.10(c)).

In medical applications outside of the operating theatre, assistive robots have been substantially studied for both general patient-care applications [62, 63, 64] and rehabilitation applications [65, 66, 67]. Additionally, with the popularisation of robotics over the last decade, assistive robot applications for domestic tasks have also gained momentum. In these applications, eye tracking can provide valuable insights to improve the quality of the robotic assistance. This can be achieved by factoring the user's intention into the decision process, or by using gaze fixation to trigger assistance. In [60], a framework using eye tracking to improve human-robot interaction in activities of daily living through assistive robots is presented. When using stereo eye tracking, i.e. the tracking of both eyes with two separate images, it is possible to find the 3D position of the focused object by estimating the convergence point of both eyes. Often times an additional non-linear calibration step is required to improve the accuracy of the mapping. In [68], 3D gaze tracking with a neural network calibration is used to train assistive robots by mimicking human visuomotor function for grasping tasks. The FreeView system presented in [61] also uses 3D gaze tracking with a neural network calibration, this time to trigger pre-trained pick and place tasks automatically, or to directly control the robot's end-effector.

Assistive robots for the operating room have also been studied. Preliminary work for two gaze-contingent robotic assistant frameworks capable of identifying

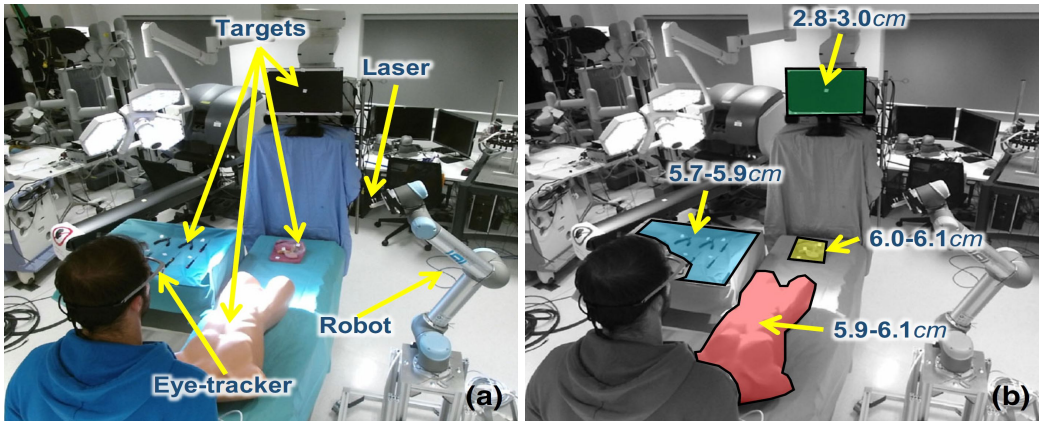


Figure 2.11: Smart operating room using eye tracking glasses and a Kinect sensor to map the user's gaze to 3D coordinates. A robot-mounted laser is used to highlight the fixated spots. Reproduced from Springer open access [69].

surgeon interaction with elements of the operating room are presented in [70] and [71]. A system presented in [69] introduces a novel way for the surgeon to interact with the surgical staff. A laser-pointer mounted on a robotic arm is used to project the surgeon's gaze fixation point in 3D space, allowing the surgical staff to visualise what the surgeon is interested in. The system, shown in Figure 2.11, uses a Kinect sensor to co-register the scene with the SLAM-based 3D reconstruction obtained using the camera mounted on the eye tracking glasses. While these approaches all show interesting concepts, current work on gaze-contingent assistive robots is still in large part in a preliminary stage, particularly regarding applications in the operating room.

The following section offers a review of gaze-contingent methods specifically applied to robotic surgery. Lastly, a case study of gaze-contingent laparoscopic control from a co-authored publication is presented, to introduce important concepts and motivations for the work presented in this thesis.

2.3.1 Gaze-Contingent Robot Control in Surgery

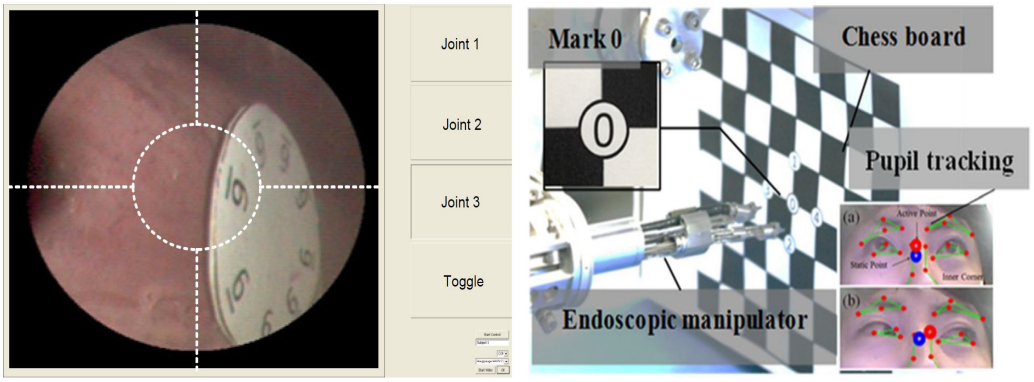
One of the most widely reported applications of gaze-contingent control in surgery is for the control of the surgical camera. Laparoscopic cameras typically have a limited field of view, and so must be frequently repositioned to allow for good visual coverage of the surgical scene. As such, it is difficult for a surgeon to perform

a bimanual task without external assistance to move the camera. This task is usually performed by a surgical assistant via verbal communication. However studies have shown that communication failures between the surgeon and the assistant can be cognitively taxing as well as dangerous to the patient [72, 73].

Different means of active robotic camera control have been explored, such as head-mounted tracking [74], verbal control [51], and manual control [75]. While these systems allow the surgeon to keep both hands free they present a relatively large footprint in an already crowded operating theatre, and require direct surgeon control to function.

An attractive alternative is to use eye tracking data to control the movement of the camera, typically by re-centring the camera around areas of interest. In [76] an articulated mechatronic laparoscope is controlled using a joint-controlled gaze-contingent method. This technique uses an eye-position based proportional controller to move one robot joint at a time, as illustrated in Figure 2.12(a). However this system does not provide task space control of the robot, and an approximate joint mapping is used instead. An intention recognition scheme using a Support Vector Machine (SVM) classifier to detect gaze commands from pupil tracking is presented in [77]. This intention recognition scheme is then used along a probabilistic neural network to control a robotic laparoscope [78, 79]. The approach presented in [77] is interesting because it tries to classify gestures as intentional or not by using a classifier directly from raw pupil tracking motion. However the laparoscope control presented in [78], shown in Figure 2.12(b), is only capable of motions between pre-defined points of interest. Similarly to [76], an approximate joint mapping is used to translate the 2D gaze into motion commands. Additionally, neither approach supports zooming functionality, i.e. the translation of the camera along its optical axis, a crucial feature for surgery as noted in [78]. Finally, note that in [78] the entire robot sheath moves with the camera, including the instruments. This would make it challenging to use in practice, as the instruments would need to be clear from any anatomy before each motion.

An alternative approach for differentiating intentional gaze gestures is through the detection of gaze fixations. A widespread approach to identify these fixations is presented in [80], through a threshold dispersion technique. Methods to select the optimal threshold value are discussed in [81]. Another approach with similar reported performance is reported in [82], instead using a two-step spa-

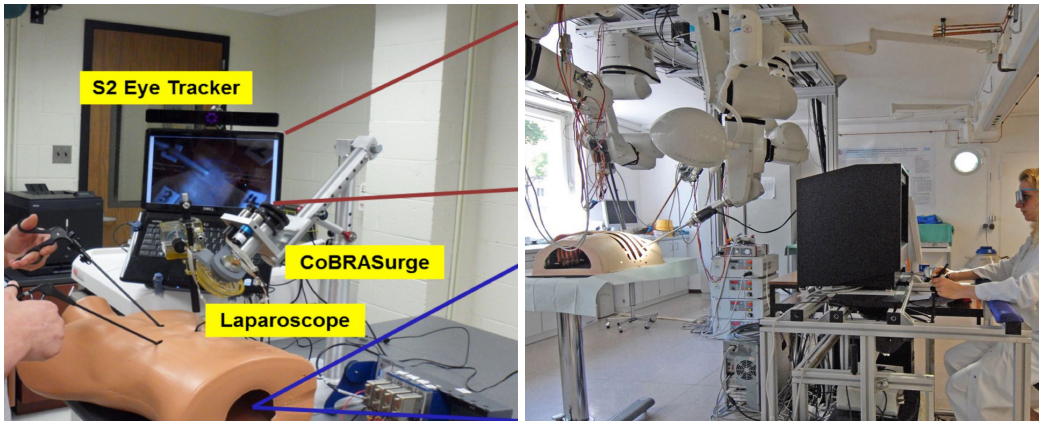


(a) (b)

Figure 2.12: Gaze-controlled surgical cameras: (a) Using joint control through fixation-activated buttons, ©IEEE 2010 [76]. (b) Using joint control mapped from 2D fixations, reproduced from Springer open access [78].

tial dispersion threshold. Yet another approach is presented in a gaze-contingent laparoscope control system using the CoBRASurge robot [83], illustrated in Figure 2.13(a). In the approach presented in [83], a fuzzy-logic inference engine is used to infer the user's intended destination. In an earlier approach, a dwell-time gaze fixation method from [80] had been implemented with the same system [84]. While the authors reported improvements using their fuzzy-logic inference engine, it should be noted that both iterations of their system controlled the robot using whole trajectories sequentially, moving from gaze fixation to gaze fixation. The discrete nature of these motions meant that the system relied on precise areas of interest to be detected, and relatively lengthy gaze fixation detection times were used (median times of 1.367s with their fuzzy logic). These slow response times and discrete motions tend to make for a frustrating system behaviour, particularly when undesired scope motions can be generated inadvertently if the user temporarily focuses on a different area of the surgical scene.

The last laparoscope gaze-contingent control approach reviewed here is the ARAMIS system, presented in [85]. This system uses custom eye tracking glasses in conjunction with a custom stereo display equipped with infrared light emitting diodes. The system, shown in Figure 2.13(b), uses a ceiling-mounted robot to control the camera. A wide-angle scene camera facing towards the display is used to track the user's head position, and markers are used to register the display with

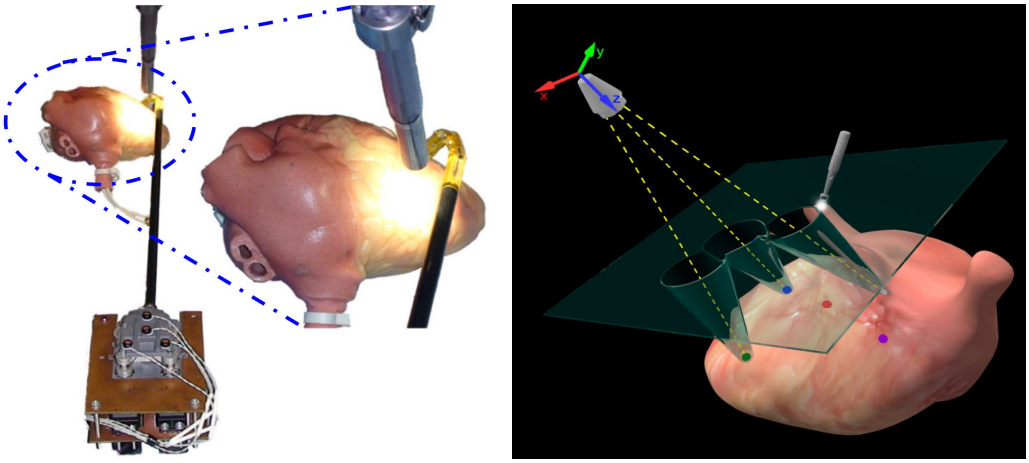


(a) (b)

Figure 2.13: (a) The CoBRASurge robotic laparoscope controlled using an S2 eye tracker (Mirametrix Inc.), ©IEEE 2013 [84]. (b) The ARAMIS gaze-contingent camera control system using eye tracking glasses, ©IEEE 2012 [85].

the camera. A velocity-based Cartesian control mode is used, which should result in smoother and more intuitive robot motions than the other methods reviewed. However the system makes no mention of any intention recognition scheme other than for the identification of dwell-time gaze fixations, an important limitation in the usability of the system. Additionally, as with all the other methods reviewed, no zooming functionality is available.

While a large part of the literature on gaze-contingent applications in surgery is focused on camera control, other applications have also shown promising results. In [88], a novel eye tracker calibration compensating for head motions is introduced, and used to perform gaze-control of an ultrasound interface. In [58], a custom-made binocular eye tracker is integrated inside a surgical console. The 3D gaze fixation points recovered allow for accurate depth reconstruction, and subsequent motion stabilisation of the robotic system. In [86], a similar approach is used to generate commands in 3D space from the user's gaze fixations. These commands are then used to direct a 4DoF robotic probe to target locations, as shown in Figure 2.14(a). The same approach of generating 3D commands from the users' gaze fixations is used in [87] to generate virtual constraints. As illustrated in Figure 2.14(b), 3D gaze fixations are used to generate zones of active constraints, preventing the surgical instruments to reach into these areas. Lastly,



(a) (b)

Figure 2.14: (a) Gaze-contingent control of a robotic probe using fixations as 3D setpoints, ©IEEE 2008 [86]. (b) Active constraints generated by the surgeon through gaze fixations, reproduced with permission from Springer [87].

haptic guidance from 3D gaze fixations are also presented in [59], guiding the user towards fixated targets.

2.3.2 Case Study: Gaze Gestures for Human Robot Interaction in Laparoscopic Surgery

As described previously in section 2.3.1, existing gaze contingent camera systems only have a panning control of the laparoscope. The lack of zoom and tilt control complicates effective navigation during surgery. The system proposed by Noonan [76], for example, requires a foot-pedal which the user needs to press to activate. The need to introduce additional hardware such as foot-pedals can lead to instrument clutter in an already complex environment. Furthermore, existing eye-controlled platforms often use dwell time on fixed regions to indicate a user's intention, which can be difficult to use in practice.

In order to overcome these problems, an alternative gaze contingent robotic camera control system was initially proposed by Fujii in [89], where the camera is activated via real-time gaze gestures rather than an external switch. Through the use of multiple gestures, which are statistically learned and can map to specific camera control commands, the system is capable of different camera control

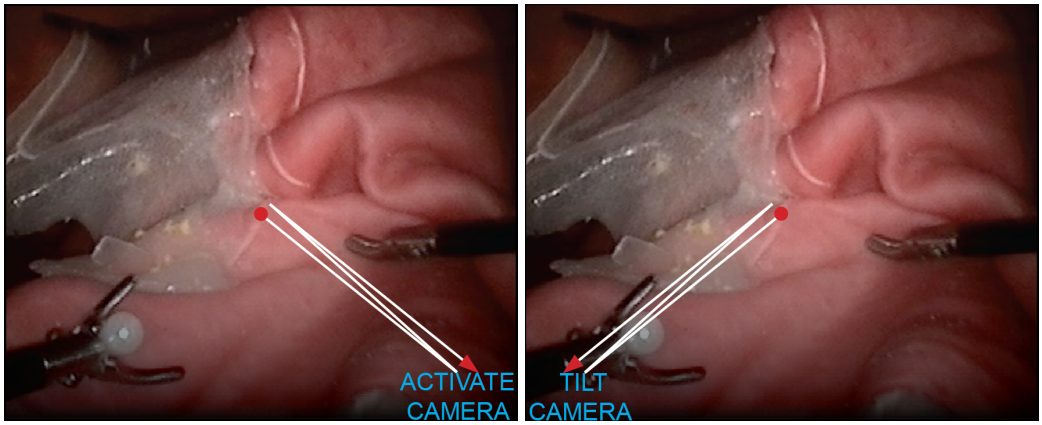
modes such as panning, zooming, and tilting. The system also incorporates a novel online calibration algorithm for the gaze tracker overcoming the need of an explicit offline calibration procedure. The gaze gesture based human-computer interaction method differs from previous gaze based interaction methods such as the eye mouse [90], which uses dwell time to convey user intention of mouse clicking, or the Manual And Gaze Input Cascaded (MAGIC) pointing method [91], which moves the cursor position in close proximity to the target location, but relies on the user to convey their intention with a small manual cursor movement and mouse click. The work presented here builds on the "Perceptual Docking" paradigm introduced by [92], and extends initial work presented in [89]. The key novelties of that work presented in this review are: i) the capability to pan, zoom, and tilt the camera (rotation around the laparoscope's longitudinal axis); and ii) the ability to seamlessly switch between panning and zooming control by using the distance from the user's face to convey the intention to zoom in or out.

While this work is described in detail in [93], and as part of co-authored work in [94], it provided strong insights motivating part of the work that follows in this thesis. As such, and as a means of introducing key techniques related to gaze-contingent control, the methods and limitations of that work are briefly reviewed below.

Gaze Gesture Detection

Gaze gestures are based on fast eye movements, i.e. saccadic movements rather than gaze fixations. They consist of a predefined sequence of eye movements. Gaze gestures can be single-stroke [95, 96] or multi-stroke [97]. When the user performs the intended unique sequence of saccadic eye movements, a specific command is activated. Gaze gestures have previously been used in applications such as eye typing [98], Human-Computer Interaction, mobile phone interaction [99] and gaming [100].

The human eye is normally used for information gathering [55] rather than to convey intention to control external devices. As such, the main challenge of using gaze gestures is distinguishing natural gaze patterns from intentional gaze gestures with high accuracy and precision. To this end, pattern recognition methods are necessary to learn these gaze gestures. The system uses gaze gesture recognition



(a) (b)

Figure 2.15: Illustration of the gaze gestures used to enable the (a) *activate camera* mode; and (b) *tilt camera* mode. Picture from the co-authored publication [94].

based on Hidden Markov Model (HMM) to learn multiple input commands from a surgeon in order to convey the desired camera control mode.

Two possible gaze gestures are introduced to control the camera: *activate camera* (for pan and zoom) and *tilt camera* (for rotation). The *activate camera* gaze gesture is illustrated in Figure 2.15(a). It is defined by the following three-stroke sequence of eye movements: gaze at the centre of the screen, then to the bottom right corner, then back to the centre, and finally back to the bottom right corner. The *tilt camera* gaze gesture is similar to the *activate camera* gaze gesture but the user is instead required to look at the bottom left corner of the screen, as shown in Figure 2.15(b). Gaze gestures oriented towards the corner of the screen are chosen to prevent obstruction of the camera view, minimise the amount of screen space necessary, and reduce detection of involuntary gaze gestures. Text labels were placed at the bottom left and right corners of the screen to indicate the mode to be activated by performing a gaze gesture in that direction.

Once a gaze gesture is identified, the robotic arm is activated and the user is able to control the laparoscope with his gaze. The user can also deactivate the gaze contingent laparoscopic control via gaze gesture. It is composed of two processes: a gaze gesture recognition process and a robot control process. The gaze gesture recognition process analyses inbound gaze data and identifies whether a gesture has been performed. The robot control process uses gaze data, i.e. where the user

is looking at, to generate the robot trajectory. Finally, the user can stop the robotic camera control by fixating the *stop camera* text present at the bottom left corner of the screen during robotic control.

The gaze data generated by the eye tracker is first passed through a median filter with a 150ms time window to reduce the noise inherent to eye data. The filtered data is then used in a gesture segmentation algorithm. Since the proposed gaze gestures consist of three fast eye movements, i.e., saccades, the gesture segmentation algorithm was designed to detect three such sequential saccades. These saccades are detected by searching for two sequential gaze data samples which exceed a $300^{\circ}s^{-1}$ velocity threshold. In this context, the term "stroke" is also used to describe a saccade. A velocity threshold saccade detection technique [80] was implemented. Additional temporal constraints were added to ensure that the gaze gestures were not excessively long (each gaze stroke was required to be less than 750ms), allowing segmentation of continuous and user-intended gaze gestures. This segmentation process outputs a stream of strokes which is analysed by the gesture recognizer in a sliding window fashion.

The xy coordinates from the segmented trajectories of a potential gaze gesture are then clustered using a pre-trained k-means algorithm. Each cluster's symbol number, centroid coordinates, and radius are used collectively to create a discrete codebook that captures the relevant features of the gaze gestures, i.e. the xy coordinates. Each potential gaze gesture sequence is encoded using the codebook, where symbol numbers are assigned to each observation (xy coordinate) by using the distance between the observation and the centroid of each cluster, provided that it is within the defined radius. In order to recognise the segmented potential gaze gestures, two *left-to-right* HMMs were used for each camera control activation mode. Unlike in [96], gaze patterns are not analysed just to identify the nature of the gaze data, i.e. whether it constitutes a continuous motion, fixation or a stroke, but to classify which kind of gesture is being performed. The *activate camera* gaze gesture is modelled by a first HMM and enables panning and zooming control of the laparoscope. The *tilt camera* gaze gesture is modelled by a second HMM and enables rotation around the laparoscope's longitudinal axis.

Interface Design

The camera is initially stationary while the system waits for a gaze gesture input from the user. The user has the option to control the camera via *activate camera* or *tilt camera* modes. The *activate camera* mode enables panning or zooming, and is activated by a gaze gesture. Switching between panning and zooming is enabled with a movement of the head forward or backward, as illustrated in Figure 2.16. In the *tilt camera* mode, which is activated by a different gaze gesture, the system allows the camera to rotate the view around the laparoscope's longitudinal axis.

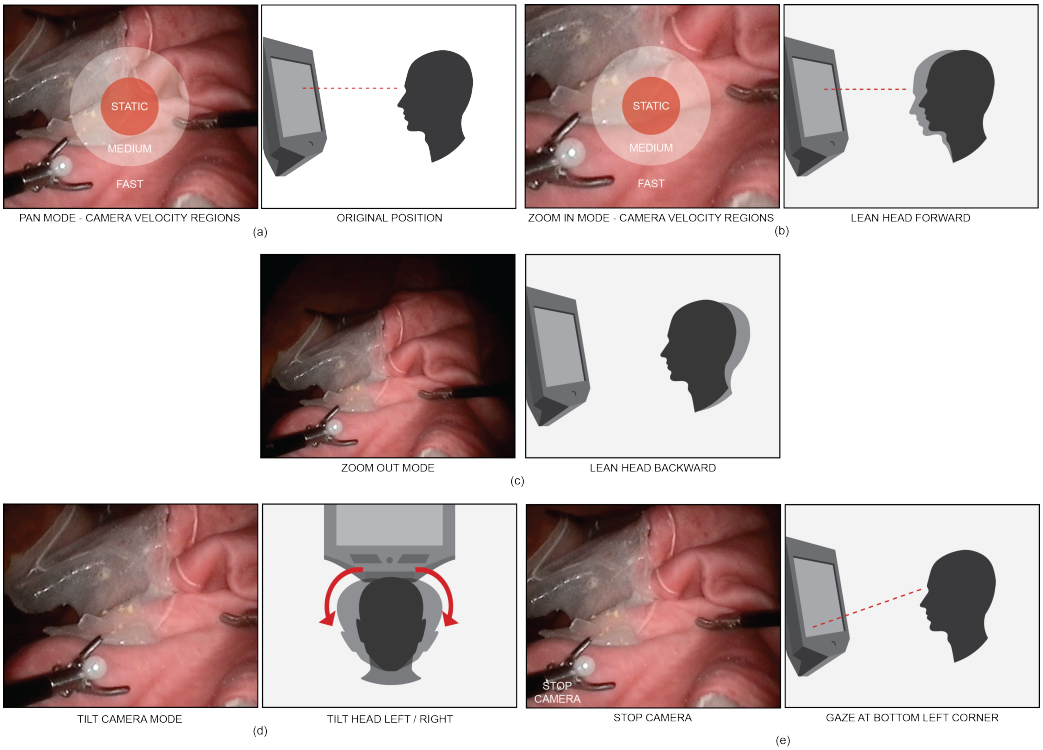


Figure 2.16: Illustration of the different types of control motions for the gaze-gesture robotic laparoscope system. (a) Panning motion. (b) Zoom-in motion. (c) Zoom-out motion. (d) Tilting motion. (e) Stopping motion by fixation of the left hand corner for 750ms. Picture from the co-authored publication [94].

Upon activation of the *activate camera* mode, the distance of the user's head relative to the screen is saved as the *original distance*. The panning motion is available in the *activate camera* mode as long as the distance from the user's head to the screen is within 5cm of the *original distance*. During this panning mode, the

camera will pan following the vector direction from the screen center to the gaze point. This vector is used to generate a target velocity for the robot, depending on distant from the screen centre the gaze point is. The screen area is separated into three radial speed regions as shown in Figure 2.16(a). Gazing within each region moves the camera at a different velocity accordingly. If the surgeon's gaze point is within the central screen region, the camera remains stationary allowing for a stable view whilst performing tasks. Otherwise, the camera moves at a velocity of $16.9mms^{-1}$ and $23.3mms^{-1}$ respectively for the regions of medium and fast velocities.

The zoom functionality is enabled during the *activate camera* mode when the user moves their head by a distance greater than 5cm from the *original distance*. The system will continuously zoom in if the user moved their head closer to the screen, or continuously zoom out if the user moved their head away from the screen, as illustrated in Figure 2.16(b) and 2.16(c) respectively. The zooming motion continues until the user's head comes back within 5cm of the *original distance*. For reasons of safety and accuracy, simultaneous panning motions are only allowed while zooming in, not when zooming out.

Upon activation of the *tilt camera* gaze gesture, the initial left and right eye positions are recorded and the horizon going through the left and right eye is calculated and stored as the *base line*. The horizon is continuously tracked by updating the position of the left and right eye. The angle formed by the current horizon and the *base line* is used to control the camera tilt. The camera can then be tilted left or right by the user respectively tilting their head left or right by more than $\pm 15^{\circ}$. Like in the zooming mode, the robot will continuously tilt the camera in the direction indicated by the user's head until the angle formed by the current horizon and the base line becomes less than $\pm 15^{\circ}$.

Guidance text is overlaid onto the camera view, and the camera can also be stopped by fixating the *stop camera* text at the bottom left hand corner of the screen (Figure 2.16(e)). The *stop camera* command is identified by detecting dwell-time gaze fixations of at least 750ms in that region of the screen. In order to address the potential uncertainties when the eye tracker loses tracking of the user's eyes, a safety mechanism is introduced where the robotic system immediately stops under lost gaze tracking circumstances. On re-detection of the user's gaze, the robotic laparoscope resumes with the same control mode as before the tracking

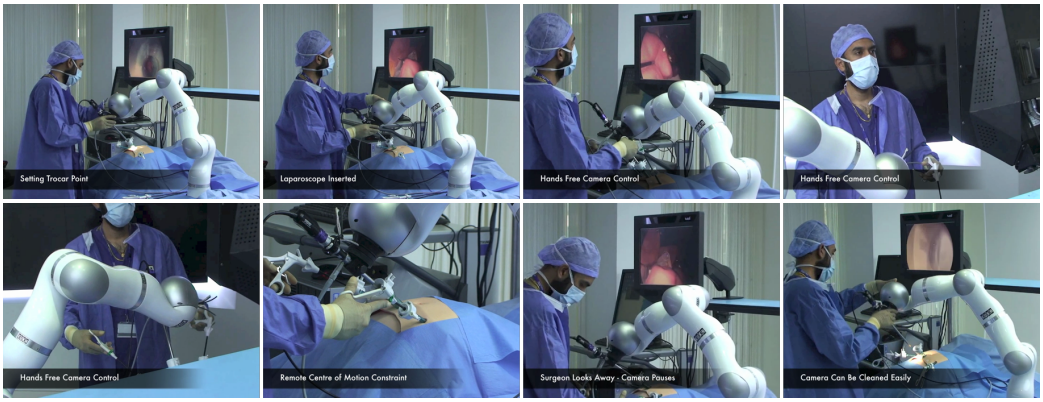


Figure 2.17: Operative workflow using the gaze-assisted robotic control. Picture from the co-authored publication [94].

was lost. The operative workflow is illustrated in Figure 2.17, where a surgical resident performs a lesion removal task on an upper gastrointestinal phantom.

Discussion

Validation of the system showed that HMMs are effective in recognising gaze gestures and that user intention can be separated from unintentional eye movements, providing a means to communicate with the robotic laparoscope. Additionally, a comprehensive usability study demonstrated that once the group of surgeons learnt how to use the system, they were able to perform a surgical navigation task quicker, with a superior camera and instrument efficiency when compared to instructing a camera assistant or when using a pedal activated control scheme. Overall, this work demonstrated the potential of gaze contingent laparoscopic control to allow the surgeon to rapidly execute a bimanual task without requiring a camera assistant.

However, a number of important limitations can also be noted. The results presented in [94] show that the Cartesian-space velocity-based control of the laparoscope was a substantial improvement over previous joint-space solutions such as in [76]. Nonetheless, user feedback suggested that the discrete velocity zones of the panning mode could feel unintuitive, generating rapid variations in robot behaviour. Surgical feedback also included the desire to be able to manually tune the speed of the robot, in order to achieve a pace they are comfortable with. As such, providing a means to achieve different velocities in a safe manner could pos-

itively impact the performance of a gaze-contingent laparoscope control scheme.

Additionally, the very nature of gaze gestures lends itself to criticism. Normal human eye behaviour consists of using sight as an input, to explore and investigate the surrounding world. However, gaze gestures require the user's eyes to actively perform a combination of predefined movements, which can feel unnatural and tiring. Gaze gestures require the surgeon to use their eyes as outputs, similarly to pressing a button with a finger of their hand. This means that while gaze gestures are successful at effectively creating a third hand, actioned with the eyes and thus allowing the user to perform bimanual tasks, they come at the cost of unnatural eye behaviour. A related issue is that using gaze gestures still creates an interruption of the workflow since the user has to temporarily take their attention away from what they are doing to perform the desired gesture. Furthermore, activating a gaze gesture while in a camera control mode can make the system move in undesired ways. These factors are key in motivating an implicit approach for gaze-contingent control, making use of the scene context where possible to provide a seamless interface.

2.4 Visualisation Interfaces in Robotic Surgery

Visualisation interfaces play a crucial role in robotic surgery, relaying critical information about the surgical scene and information about each of the systems used for the procedure. Similarly to robot control interfaces, visualisation interfaces cover a broad range of systems and methods. As the visualisation interfaces developed in this thesis focus on context-aware augmented reality behaviour for surgical navigation, we limit the scope of the following review to visualisation interfaces related to intra-operative image guidance, particularly augmented reality.

2.4.1 Surgical Navigation Interfaces

Image-guided procedures are surgical procedures in which pre-operative and/or intra-operative images are used to guide the surgeon as they move their instruments through the patient anatomy. Similarly to using a Global Positioning System (GPS) while driving, image guidance can assist the surgeon by providing both a roadmap for the procedure, and real-time guidance during the operation.

This can be especially useful in surgical procedures with limited visibility such as laparoscopy or endoscopy [101], or when dealing with highly delicate anatomy [102]. In neurosurgery, image guidance allows surgeons to limit collateral damage to brain tissue during the operation by guiding them efficiently to the operating site. When performing tumour resection, surgical navigation can help the surgeon observe small positive margins around the tumour, limiting unnecessary cutting while still removing the entirety of the tumour.

Perhaps the most common way to visualise surgical navigation information is through the use of tri-planar displays [104, 102, 103]. This technique, illustrated in Figure 2.18, consists in showing where the end-effector of the tracked surgical instrument is located relative to the imaging information available. Tri-planar displays are a natural evolution of medical image visualisation, as pre-operative im-

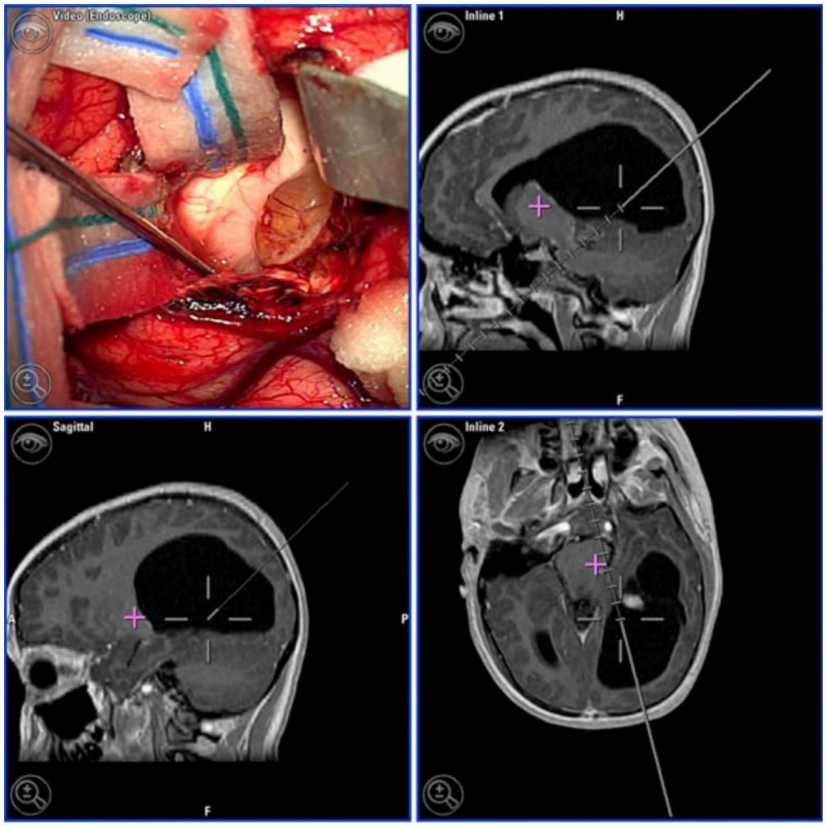


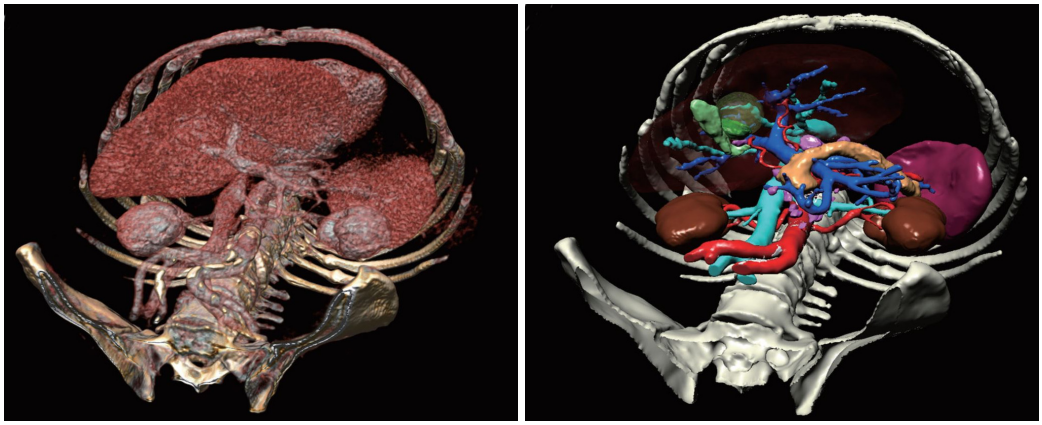
Figure 2.18: Intra-operative image guidance of a tracked instrument using a tri-planar display. Reproduced with permission from Springer [103].

ages from Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) scans are typically visualised using 2D slices. Surgeons are thus well trained to understand these displays, making them easy to integrate within the surgical workflow. Figure 2.18(top left) shows the end effector of the tracked surgical instrument. The tri-planar display (Figure 2.18(top right, bottom left, bottom right)) shows the position of this instrument from three different orthogonal views.

The three-way-split 2D nature of the tri-planar displays presents some disadvantages as a visualisation interface. While surgeons are trained to understand the 2D views used in CT and MRI scans, this interface is not optimal for the display of real-time 3D information. When viewing scans pre-operatively, surgeons will scroll back and forth between the different parallel images of the scan along the scanning direction, granting them a strong spatial understanding of the anatomy. However, relying on a single set of three orthogonal planes is much less effective to represent a 3D volume, leaving a large portion of the space effectively invisible. While it is possible for the surgeon to scroll through the pre-operative images during the procedure, this further increases the disruption to the operation caused by needing to look away from the operating site and check the three different displays.

An alternative to tri-planar displays is to use interfaces that allow the surgeon to directly visualise the 3D volume. Two main methods are used in surgery: volume rendering, and surface rendering. Volume rendering uses the CT or MRI scans directly to create a 3D projection of the entire volume, using techniques such as ray-tracing [105]. Transfer functions are used to specify the visualisation properties of the volume, based on the properties of each voxel [106]. Figure 2.19(a) shows an example of volume rendering obtained from an abdominal scan. A key advantage of volume rendering is that it allows the whole volume to be visualised directly from the scans, without requiring pre-processing. However, the quality of the visualisation depends on the transfer function selected. Automatic selection of transfer functions is still an active area of research, and so great care must be taken to design an appropriate transfer function for the particular volume to visualise [107]. Additionally, volume-rendering methods tend to be computationally expensive, with a significant trade-off between visualisation resolution and rendering speed.

Surface rendering does not directly use the scanned dataset. Instead, meshes



(a) (b)

Figure 2.19: Comparison between volume rendering (a), and surface rendering (b). The patient anatomy has been segmented for the surface rendering. Reproduced with permission from the Hepatobiliary Surgery and Nutrition journal [108].

consisting of geometric primitives such as triangles must first be segmented from the dataset before being rendered. These meshes represent the boundaries between anatomical structures, such that each structure of interest will have a mesh delineating its outline. The obvious drawback of this approach is the time required for the pre-processing step necessary to generate the meshes. However, once complete, the surface rendering itself is much less computationally intensive than volume rendering, and can be performed in real-time without compromises on quality. Additionally, the required segmentation step allows for the clear delimitation of anatomical structures, useful to highlight precise tumour margins for surgical navigation for example. Software methods also exist capable of facilitating and even partially automating the segmentation process [109]. Figure 2.19(b) shows the surface rendering corresponding to Figure 2.19(a).

When used individually, such as in [110], both surface rendering and volume rendering are forms of virtual reality. These techniques provide the surgeon with a purely virtual environment in which they are able to visualise the 3D volume. This still requires the surgeon to take their attention away from the surgical scene to benefit from the image guidance. To overcome this issue, augmented reality techniques can be used by merging the virtual environment with real-time images from the surgical scene.

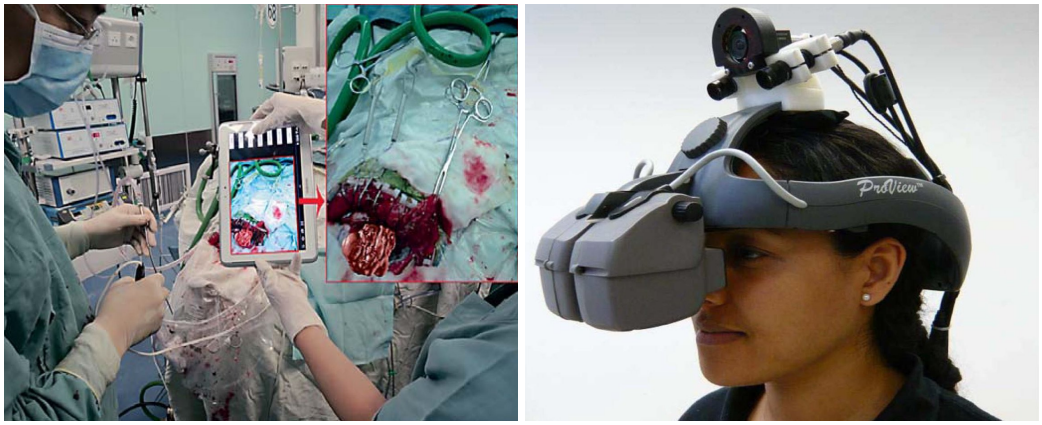
2.4.2 Augmented Reality Display Interfaces in Surgery

Augmented reality consists in overlaying images from a virtual scene onto live surgical images. The content of the virtual scene is used to provide surgical navigation information to the surgeon, and can be obtained using surface or volume rendering as described in section 2.4.1. Unlike in virtual reality, the information is fully integrated into the current surgical view, effectively enhancing the surgical images with image guidance. For the virtual elements to be overlaid correctly onto the surgical image, the position of the anatomy must be known relative to the display camera point of view. Well-known registration and calibration procedures exist capable of yielding high accuracy for static environments, however challenges arise when dealing with heavily deformable environments such as in laparoscopy. This can be particularly problematic when relying on pre-operative datasets for image guidance that are not updated throughout the procedure. Solving these registration issues is still an active field of research, but reliable solutions such as intra-operative ultrasound or MRI do exist. For more information on registration techniques the interested reader is referred to [111, 112].

To use augmented reality, a suitable display interface capable of showing the combination of virtual and real images must be used. These can be broken down into two categories: opaque interfaces in which the virtual and real images are merged through processing and displayed on an opaque surface, and transparent interfaces in which the virtual and real images are optically merged using transparency.

Opaque Interfaces

Like virtual reality, it is possible to display augmented reality on any screen or monitor. Images acquired from the surgical scene can be merged with the virtual images in software, and then displayed on a screen. While seemingly simple, this solution offers several important advantages. Firstly, this method can work with any system where the images from the surgical scene can be obtained digitally. This makes it a suitable choice for displaying augmented reality in minimally invasive surgery. Another advantage of opaque displays is that the software processing of the real and virtual scenes can be leveraged to great effect to improve the realism of the virtual elements shown with the augmented reality, as discussed



(a) (b)

Figure 2.20: (a) Augmented reality display using a tablet, reproduced with permission from Karger [113]. (b) Augmented reality display using an opaque head-mounted display, ©IEEE 2009 [115].

in detail later in section 2.4.3.

The opaque nature of the display does not impose that a static screen be used. Portable opaque displays such as tablets have proven to be an effective and popular way to display augmented reality in surgery, due to their low-cost and small footprint. Although they introduce another source of registration errors, tablets and smartphones can be used to simulate see-through displays by using augmented reality with the video feed from their camera. In [113], a tablet-displayed augmented reality system was used for navigation in neurosurgery, using the tablet's own camera. The system, shown in Figure 2.20(a), was trialled in both a pre-clinical and clinical study with positive results. In [114], a visual SLAM approach was used to provide marker-less augmented reality using a tablet, although the reported accuracy of 5mm is not suitable for delicate applications such as neurosurgery.

Opaque head-mounted displays (HMD) such as displayed in Figure 2.20(b) can also be used to display augmented reality. The recent introduction of HMDs to the consumer market with gaming systems such as the HTC Vive and the Oculus Rift make them attractive options for surgery. Opaque HMDs can be used in surgery with a head-mounted camera, such as in [115] and [116], or with a video stream from another source such as a laparoscope. However, due to their head-



Figure 2.21: Augmented reality interface using a projector to display the image guidance information directly onto the patient. Reproduced with permission from the Wiley Online Library [120].

mounted nature, the latter option may cause nausea if the laparoscope's movements do not match the surgeon's head motions. Additionally, because opaque HMDs completely occlude the surgeon's sight they can feel quite restrictive for the surgeon.

Transparent Interfaces

The overlay-driven nature of augmented reality lends itself well to display interfaces naturally able to exploit transparency. A first example of this is the injection of virtual images through the optical eyepiece of a surgical microscope. This approach is particularly attractive in neurosurgery such as in [117, 118, 119], where surgical microscopes are commonly used. Using a stereotactic frame to maintain the patient's head still, this approach makes it possible to achieve high accuracy surgical navigation while using the same display interface as in non non-guided procedures.

A very different approach consists in directly projecting the virtual images onto the patient anatomy [120, 121, 122]. This approach, illustrated in Figure 2.21, allows the surgeon complete freedom of movement around the operating



(a) (b) (c)

Figure 2.22: See-through head mounted displays. (a) Google Glass™, reproduced with permission from Elsevier [123]. (b-c) Microsoft HoloLens™, reproduced with permission from Springer [124].

room. Additionally, this approach allows all of the surgical staff to view the augmented reality. The accurate projection of this augmented reality is highly challenging, however, given that the display surface is the patient itself. While normally changes to the anatomy during the procedure can cause a loss of registration accuracy, in a projection-based approach those changes will also affect the surface on which the augmented reality is displayed. Accounting for these changes requires accurate 3D reconstruction, which will introduce a new substantial source of display error in the system. As such, projection approaches are best suited to the initial stages of the procedure, such as the placement of surgical trocars.

Similarly to opaque HMDs, transparent HMDs can be used to display augmented reality. However, unlike the former, transparent HMDs do not require an additional head-mounted camera to be used in the operating room, and do not obstruct the surgeon's sight other than for the partially opaque virtual elements shown. Additionally, the recent development of novel transparent HMDs such as Google Glass™ and the Microsoft HoloLens™, have made this technology a highly interesting candidate for displaying augmented reality in surgery. In [124], the HoloLens™ was used to provide image guidance for navigation and dissection of pedunculated flaps during a preliminary reconstructive surgery case study, as shown in Figure 2.22(b) and 2.22(c). In [125], the HoloLensTM was used to provide guidance during endovascular phantom studies, showing the potential of

Interface Type	Display Accuracy	Freedom of Movement in the OR	View Obstruction	Suitability for MIS	Requires Tracking
Static Screen	+ +	-	+	+ +	✗
Tablet	-	+ +	+ +	-	✓
Opaque HMD	+ +	+ +	- -	+ +	[2]
Microscope injection	+	-	- -	✗	✗
Projection	- -	+ + +	+ +	✗	✓
Transparent HMD	-	+ +	+ +	- - [1]	✓

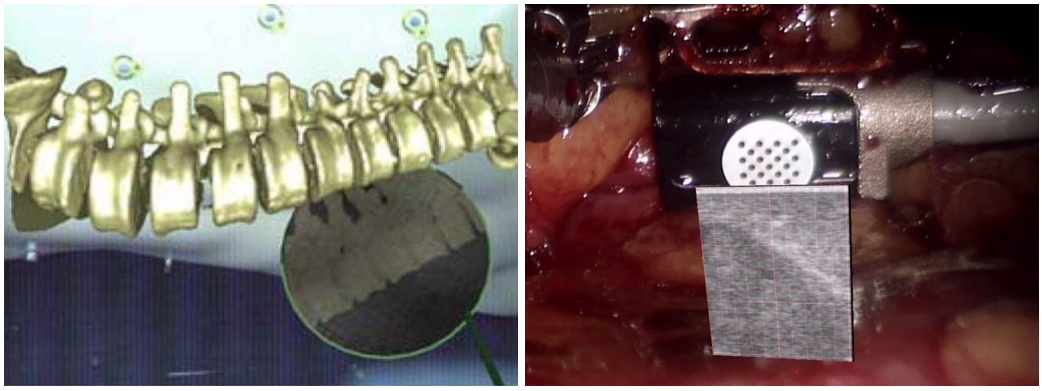
Table 2.5: Summary of advantages and drawbacks of different augmented reality display interfaces. 1: can be used with an external MIS camera feed, but the transparency is then only a drawback. 2: depends if used with a head-mounted camera video-stream or not.

augmented reality to improve surgical workflow. The HoloLensTM was also used in [126] as an alternative to expensive near infrared fluorescence augmented reality systems. While Google Glass^{TM}, shown in Figure 2.22(a), also presents a transparent display for augmented reality, the location of that display in the top right corner makes it difficult to use for surgical navigation. It is instead more useful as a heads up display to show patient vitals or telementoring during surgery [127, 123]. However, due to their transparent nature, HMDs are even less suited to displaying augmented reality for minimally invasive surgery, as the natural transparency of these interfaces is instead purely distracting.

Table 2.5 presents a qualitative summary of the advantages and drawbacks of the opaque and transparent interfaces reviewed above.

Virtual Interfaces

The opaque and transparent interfaces mentioned previously categorize physical interfaces used to display augmented reality. However, it is entirely possible to simulate other display interfaces virtually. These virtual interfaces are not true display interfaces, and still require the use of one of the interfaces mentioned previously, but can be used to generate useful effects. An example of this is display of a virtual mirror in [115], illustrated in Figure 2.23(a). The system simulates a



(a) (b)

Figure 2.23: (a) Virtual mirror used to provide additional viewing angles and improve scene perception. ©IEEE 2009 [115]. (b) Augmented reality overlaying 2D ultrasound information, reproduced with permission from Springer [128].

virtual mirror attached to the extremity of a tracked rigid instrument, able to display the virtual anatomy from different angles. Another virtual interface is used in [128] to display the 2D image from an ultrasound probe. As shown in Figure 2.23(b), the 2D image is displayed at the correct position under the tracked probe, allowing the intra-operative image guidance from the ultrasound probe to be displayed meaningfully in the surgical image.

2.4.3 Augmented Reality Perception Issues and Display Choices in Surgery

The display interfaces presented previously in section 2.4.2 dictate the physical manner in which the augmented reality can be displayed. However, even after choosing a display interface and virtual elements to show, crucial design choices still need to be made regarding how and when to display those elements. For a large number of augmented reality systems in the literature, the virtual elements are simply shown as a semi-transparent overlay on top of the surgical image. Figure 2.24 illustrates the resulting augmented reality obtained using this approach, for two often cited systems found in the literature [129, 118].

While the virtual elements may in fact be positioned correctly on the surgical image, it is immediately obvious that the virtual elements shown in Figure 2.24(a)

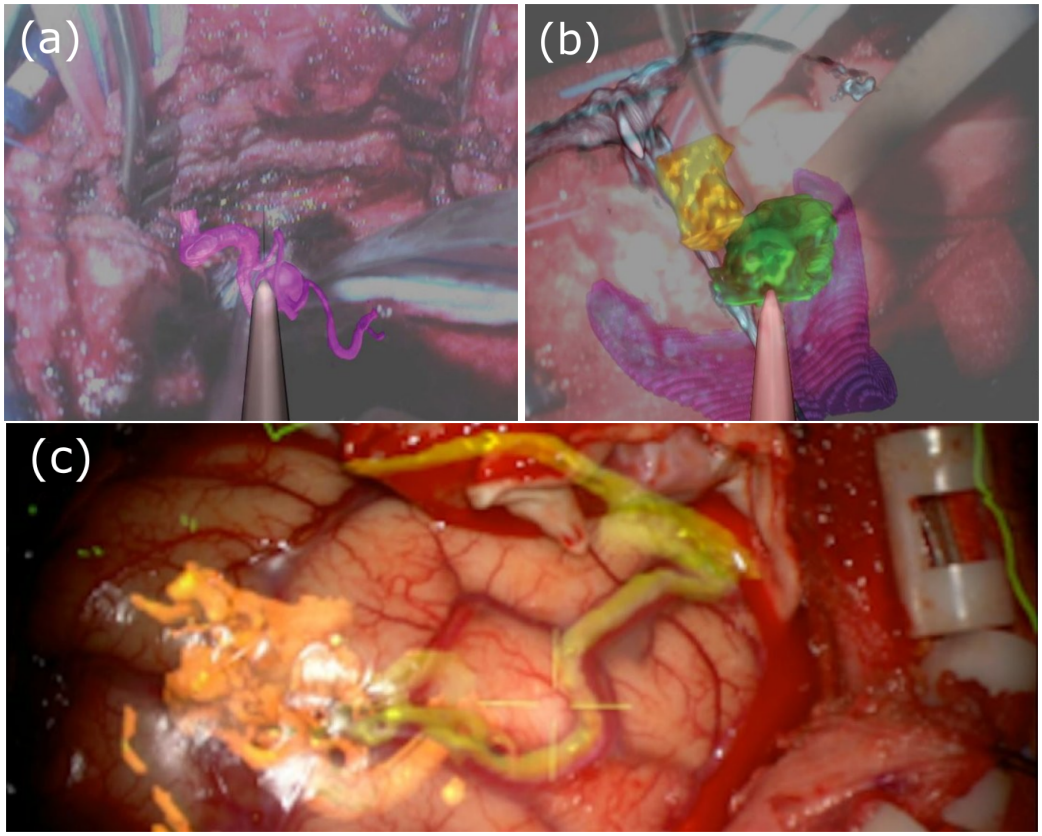


Figure 2.24: Augmented reality overlays using only transparency to merge with the scene. (a-b) DEX-Ray system, reproduced with permission from Neurosurgery [129]. (c) Augmented reality system for the treatment of cerebral arteriovenous malformations, reproduced with permission from Springer [118].

and (b) do not seem to belong to this surgical scene. One could argue that this is due to the fact that the virtual elements use unrealistic colours, or are too opaque, or even that the corresponding anatomy is not yet visible. However the fundamental issue with this visualisation is the lack of any visual cues allowing the user to perceive the relative depth of elements in the scene. Augmented reality is supposed to generate the illusion of see-through vision, and in this case the virtual elements simply look like they were placed on top of the surgical image.

The augmented reality shown in Figure 2.24(c) appears much more convincing around the main vein, the yellow texture reminiscent of fluoroscopy. However the same visualisation problem occurs in the left side of the image. Interestingly, while the authors report in [118] a well-functioning system, they conclude that

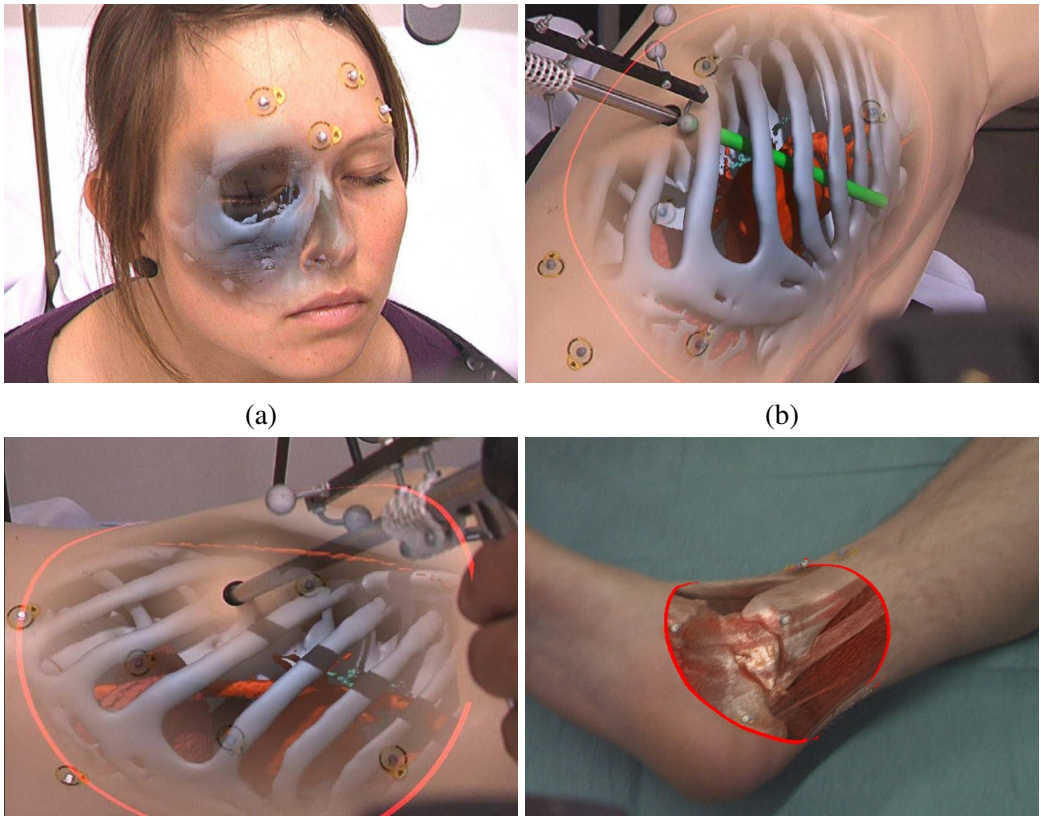
the guidance from the augmented reality displayed for the feeder vessels was not able to provide significant improvements.

In both of these cases the lack of visual cues regarding depth undermine the guidance provided by the virtual overlays. The human brain is highly sensitive to visual cues, capable of feats such as making windows with substantial glare appear transparent. When faced with contradicting cues, however, the resulting view is rejected, giving the impression of a "fake" image. Studies have shown that poor depth perception in augmented reality results in poor performance of the image guidance [130, 131], and as such finding appropriate methods for displaying depth is paramount to successful augmented reality [132, 133]. In particular, some of the most useful depth cues include occlusion, motion parallax, and shadows [134].

Figure 2.25 shows examples of augmented reality overlays making use of depth cues to enhance the realism of the scene. Shadows are used in Figures 2.25(b) and (c) to give the illusion of the laparoscopic instrument interacting with the virtual scene. Note how the trocar point in Figure 2.25(c) is rendered opaque to obstruct the virtual scene. Another important element to assist with the realism of the scene is the erasing of the real image to give way to the virtual scene. All images from Figure 2.25 contain areas where the real image is almost completely faded, showing instead the virtual scene. In Figures 2.25(a,b,c), smooth transition zones are used, outlined with a red boundary in Figures 2.25(b) and (c). Likewise, the faintly visible markers are made more opaque around the edges of the virtual scene to highlight the transition around the see-through zone. In Figure 2.25(d), the red boundary gives a sharp transition, emphasizing a "cutting" see-through visualisation.

The fading of the real image in favour of the virtual scene highlights an important trade-off in augmented reality for surgery. Fading the real image is useful to prevent a clash of depth cues between the real image and the virtual scene, and allowing the user to better perceive the virtual elements. However, fading the real image effectively hides part of the surgical scene from the surgeon, potentially masking critical information such as uncontrolled bleeding.

A solution to lessen the strong trade-off made in favour of the virtual scene in Figure 2.25 is to retain some of the features of the real image. A technique proposed by [137] exploiting this concept is shown in Figure 2.26. In this *inverse realism*, salient features from the real image are retained and displayed over the



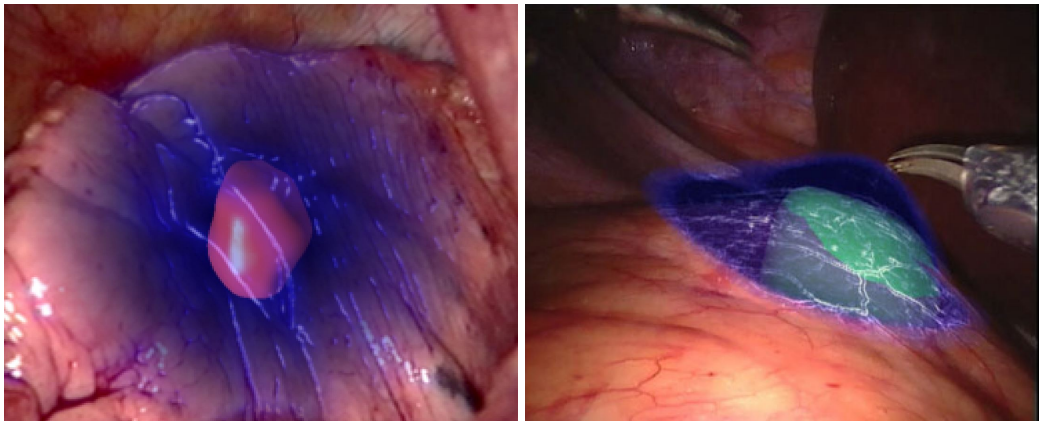
(a) (b)

(c) (d)

Figure 2.25: Augmented reality using depth clues such as occlusion, shadows, and boundary highlighting. (a,b,c) Anatomic mimesis, ©IEEE 2007 [135]. (d) Augmented reality system using volume rendering, reproduced with permission from Annual Reviews [136].

virtual overlay. To fade the real image while highlighting the retained features, the affected area is coloured. This approach allows the surgeon to keep track of any changes in the tissue surface, while still benefiting from the image guidance. Note that this colouring itself constitutes a new trade-off, preventing the surgeon from differentiating colours underneath the augmented reality overlay.

Finally, the last key design consideration in augmented reality is the choice of when to display it. The vast majority of existing systems in the literature do not discuss this point, and the augmented reality is simply controlled with an on/off switch. However, as highlighted previously, all augmented reality overlays necessarily present some trade-off between image guidance and an unobstructed view



(a) (b)

Figure 2.26: Augmented reality using distinct filtering and occlusion effects to enhance depth perception. (a) The *Inverse Realism* non-photorealistic rendering, reproduced with permission from Springer [137]. (b) Application in robotic partial nephrectomy, reproduced with permission from Springer [138].

of the surgical scene. Studies conducted in [139] show that augmented reality can have a blinding effect on users, highlighting the importance of not relying entirely on manual toggling of the augmented reality. However, while context-aware augmented reality has generated interest [140], relatively little work has been conducted in that direction. A notable example is [141], where context-aware augmented reality is achieved by detecting the current surgical phase.

The work related to visualisation interfaces presented in this thesis explores context-based augmented reality behaviour, using gaze-contingent methods and other context cues to infer user intention.

2.5 Conclusion

Surgical robots have improved drastically over the last decades, their growing capabilities facilitating more precise and less invasive procedures. To relay the capabilities of these systems to the user, appropriate human-robot interfaces are required. Control interfaces such as master haptic devices boasting high accuracy have been developed, allowing surgeons to benefit from robotic advantages while retaining complete control of the surgery. Similarly, improvements in augmented

reality have resulted in the development of realistic rendering methods and strong perceptual insights.

However, for the most part advances in the development of human-robot interfaces have been slower than those made on the surgical robots themselves. While new robotic systems push the boundaries of minimally invasive surgery, such as the recently unveiled Da Vinci SP system, haptic master interfaces suffering from similar limitations as those used two decades ago are still used. These include workspace and ergonomic limitations resulting in the need for clutching, and the lack of integrated solutions for the simultaneous control of more than two instruments. Likewise, while augmented reality approaches are now better able to relay depth, the same manual on/off display approaches are used, with demonstrated risks of inattentional blindness.

Gaze-contingent approaches provide a set of potential solutions to alleviate these issues. By tracking the user's gaze, methods have been developed to provide hands-free control of the laparoscope or generate haptic guidance of the surgical instruments. However, most existing methods provide either incomplete solutions for practical use, or rely on active gaze gestures in a way that does not constitute natural gaze behaviour.

The work presented in this thesis takes advantage of eye tracking and other scene context to develop implicit robot control and visualisation interfaces, capable of adapting their behaviour to match the situation. Implicit robot interfaces are developed to provide seamless control of the surgical camera and additional instruments. Master device workspace restrictions are also reduced by detecting user intention during reaching motions and adapting system behaviour accordingly, limiting the need for clutching. Lastly, new augmented reality methods are developed for surgical navigation, implicitly adapting the augmented reality display based on the scene context to provide improved image guidance, without obstructing the surgical view.

3

Gaze-Contingent Control for Focussed Energy Delivery

3.1 Introduction

The advent of technology in the operating theatre over the last decades has enabled surgeons to perform increasingly dexterous and delicate operations. A clear example of this is in laparoscopic surgery, where minimally invasive approaches have replaced traditional open approaches as the gold standard for a number of procedures, and have been shown to result in shorter recovery times and reduced risks of infection [142]. However, these approaches are more technically challenging than traditional open surgery, due to the complexity involved in manipulating tissue inside the body through trocar points. A key difficulty is the lack of direct line of sight to the anatomy, requiring the surgeon to use a laparoscope. As numerous laparoscopic tasks are bi-manual, i.e. they require the surgeon to use two surgical instruments at the same time, the laparoscope must be handled by an assistant. As a consequence of this, the quality of the laparoscopic vision provided by the assistant is dependent on their communication with the surgeon. Since only the surgeon knows precisely what they want to see at any given time, it is up to

Content from this chapter was published and is reproduced with permission from,

Intention recognition for gaze controlled robotic minimally invasive laser ablation. Gras G., Yang, G-Z. In IEEE/RJS International Conference on Intelligent Robots and Systems 2016.
©2016 IEEE.

them to relay those instructions to the assistant. The assistant may also use their knowledge and understanding of the scene to predict what the surgeon wants to visualise, but such predictions can be highly disruptive when they are incorrect [143, 73].

Eye tracking technology has shown promising results for allowing hands-free control of robotically-mounted cameras and tools. However, as discussed in section 2.3, existing systems present only limited capabilities in allowing the full range of camera motions in a safe, intuitive manner. These solutions typically either have limited associated robotic control capabilities [76, 77], or lack an interface capable of identifying desired interaction [85]. Importantly, the large majority of existing interfaces do not support zooming of the camera along its viewing axis, an essential requirement during surgery. An approach using gaze gestures presented previously in section 2.3.2 ([94]) allows zooming capabilities. However, while the control modes presented in section 2.3.2 exploit natural gaze behaviour, the activation of the gaze gestures themselves requires the user to perform specific motion patterns with their gaze. As discussed in section 2.3.2, this constitutes unnatural gaze behaviour and can feel both unintuitive and tiring. Additionally, existing approaches for gaze-assisted control of surgical tools are principally limited to haptic guidance using virtual constraints [87, 59].

This chapter presents a framework for the gaze-contingent control of both a surgical camera and a focussed energy delivery system, in this case a laser. An approach for implicit interaction between the surgeon and the gaze-controlled system is developed, allowing activation, panning, and zooming of the camera, as well as automatic targeting of the focussed energy delivery system. Only natural gaze behaviour is utilized, without the need for artificial gestures. The proposed framework is resistant to noise such as blinking, and safely allows the surgeon to look away to talk to an assistant before resuming control of the system at any point. Smooth velocity profiles and acceleration constraints are used to generate an intuitive motion in Cartesian-space. Zooming of the camera and automatic targeting of the surgical laser is performed implicitly by recovering scene depth around areas of interest fixated by the surgeon. This is achieved by exploiting the synergy between eye tracking and stereo reconstruction to allow accurate depth estimation in a targeted manner around the areas of an interest. This is in contrast to [58] and [86] in which 3D coordinates are obtained by assuming that the left

and right eye pixel coordinates from binocular eye tracking represent valid feature pairs. The proposed system uses this synergy to control the zoom of the camera so that optimal camera-to-tissue viewing distance is maintained. Additionally, this enables automatic re-targeting of the surgical laser towards the relevant areas, speeding up the operative workflow. The calibration methodology developed to achieve this with millimetre precision, without visual servoing, is also presented. Finally a user study involving the entire framework is presented, validating the usability of the system compared to a manually controlled robotic system.

3.2 Gaze-Contingent Laparoscope and Laser Control

3.2.1 System Overview

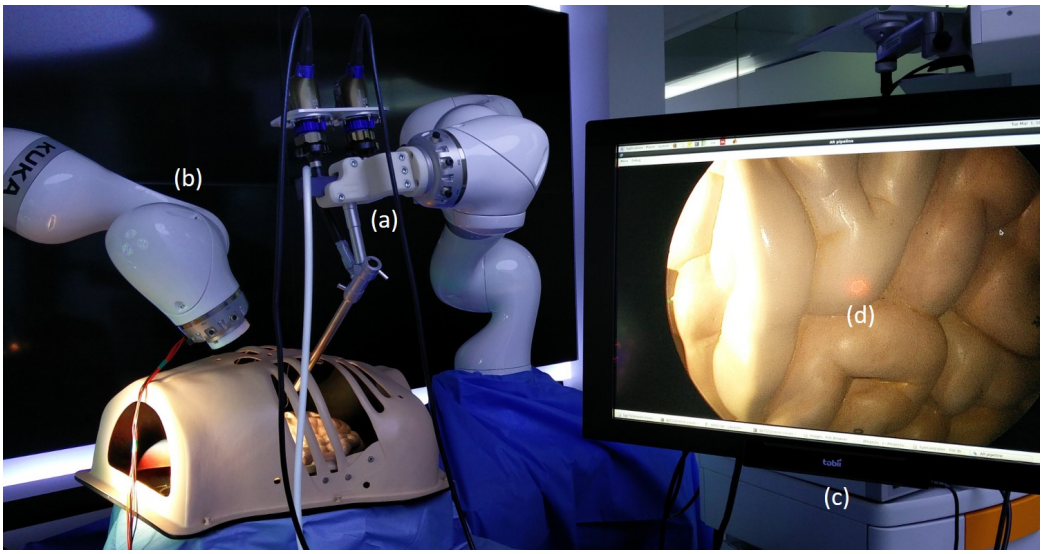


Figure 3.1: System overview. (a) Robot 1 holding the stereoscope, (b) Robot 2 holding the laser diode, (c) Tobii X2-60, (d) View of the laser dot on the screen.

The presented system comprises a Tobii Pro X2-60 (Tobii Technology AB) mounted on a LMD-3251MT medical-grade stereo monitor. This monitor relays the video feed acquired by a stereoscope. Additionally, two Kuka iiwa (KUKA Roboter GmbH) 7 degrees of freedom redundant robotic arms are used. The stere-

oscope is rigidly mounted on the end effector of one of the robotic arms. In order to simulate a surgical laser, a laser diode was placed inside a custom holder, mounted onto a second Kuka iiwa, and controlled via an Arduino Teensy 3.2. A picture of the complete system is shown in Figure 3.1. Note that the laser pointer is kept off by default, as would be the case in a real system. The laser pointer can be activated to simulate two different scenarios with a real system: 1) verifying where the laser is going to fire using a low-power mode (or a real laser pointer), and 2) firing the laser for tissue ablation. This section divides the eye tracking framework into three parts: the interaction detection interface is presented first, followed by the velocity planner used to direct the panning motion of the surgical camera, and finally the use of the localised stereo reconstruction to perform the automatic zooming motion and laser targeting.

3.2.2 Interaction Detection

The aim of the interaction detection is to understand whether or not the user is trying to interface with the system, and to provide the safest and least frustrating experience possible while interfacing with the system. Ideally the system should feel completely transparent to the user: it should activate as soon as the user wants to interact with it, disable safely if the user switches their attention away from the system, and be robust enough to disregard minor natural nuisances such as blinks or brief glances. While controlling the camera, the system should provide enough speed to prevent frustration when the surgeon wants to navigate, without sacrificing stability while performing a static task.

The interaction detection algorithm attempts to evaluate whether the user is trying to interact with the system, and whether the eye tracking data is reliable enough for robotic control. This question is processed using a state engine that updates with every cycle of the eye tracker. A Tobii Pro X2-60 with an update rate of 60 Hz is used to track the user's eyes. Every update cycle, the algorithm analyses multiple past sequences of eye data and settles in one of three possible states: ON, OFF, and PAUSED. The camera controller is active in the ON state and disabled in the OFF state. The PAUSED state is triggered when incoming data is deemed unreliable enough for the system to determine whether the user is still trying to interact with it or not, or whether the data quality is no longer

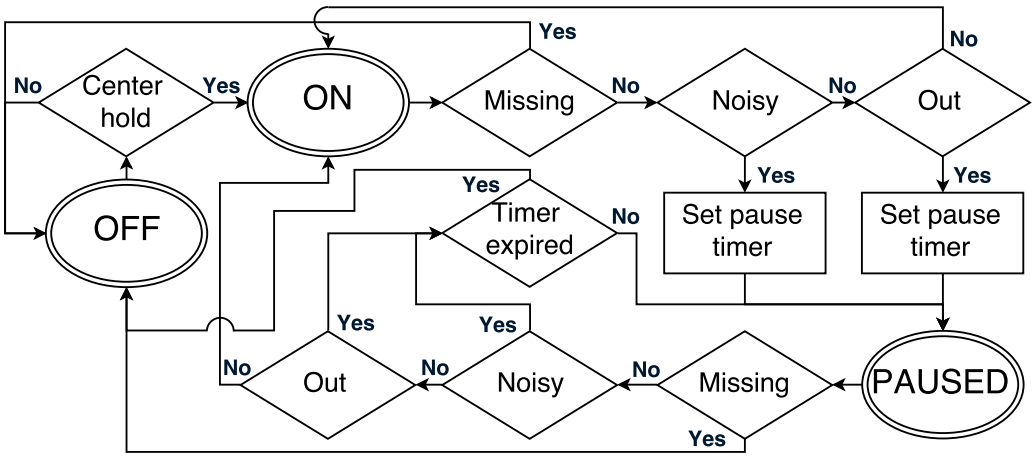


Figure 3.2: Interaction detection state diagram.

sufficient for robotic control. In this state the camera controller is temporarily suspended, without bringing the camera to a full stop, while the system waits for more data cycles to choose the subsequent state. The state diagram of the algorithm is displayed in Figure 3.2.

Circular buffers store all the raw data coming from the eye tracker for the last three seconds, more than required by the processing. Every transition in the state diagram requires the analysis of sequences of the stored data, to identify whether or not specific criteria are met. At the beginning of every update cycle a series of these checks is run, as shown in Figure 3.2. The results of these checks dictate which state the system achieves at the end of the update cycle. The different conditions and numerical values for the criteria are shown in Table 3.1. These values were obtained and fine-tuned using results from over 20 participants, over multiple iterations of the system. Each criterion possesses a tolerance threshold and sequence length. In order for a criterion to be met, the samples within the corresponding time-sequence have to satisfy the criterion's condition within the tolerance limit. For example, for the *center hold* criterion to met, no more than 15% of the samples from the previous 700 milliseconds should be outside a sphere centred in the middle of the screen with a diameter of 0.2 in normalised coordinates (normalised values range from 0 to 1 from one edge of the screen to the other).

The system starts in the OFF state and is activated by looking at the centre of

	Center Hold	Missing	Noisy	Out
Success Condition	Gaze point inside centred sphere of normalised radius 0.2	Received samples match expected number of samples	Sample has one or both eyes visible	Gaze point is within screen margins
Threshold Tolerance	15%	15%	22%	90%, margins: 5%
Sequence Length	700 ms	300 ms	300 ms	300 ms

Table 3.1: Criteria for the eye data analysis.

the screen. If the gaze is maintained in the center part of the screen long enough to validate the *center hold* criterion, and the data connection is stable, the system is activated. Once activated, the system verifies the stability of the eye data every cycle. The first check, named *missing*, is to ensure that the connection with the eye tracker is stable, and the received number of data points matches what is expected based on the update frequency. In practice this check will only come into effect when the tracking is experiencing critical difficulties. It should not occur even when the user disengages from the system, and is only meant as a backup safety to quickly deactivate the system without passing through the PAUSED state in case something obstructs the eye tracker or the connection is lost. The next check verifies the level of noise on the eye tracking data. This refers to data obtained through a stable connection, but where the eyes are not found by the eye tracker. In this work, the eye data was considered valid if even only one eye was found. If the signal is considered noisy by failing the *noise* check, the system proceeds to the PAUSED state. Finally, the last check analyses whether the user is looking within the screen boundaries. A 5% hysteresis is allowed to accommodate for tracking imprecision around the edges of the screen, as well as a large tolerance limit. If either the *noise* or *out* check fails, the system moves to the PAUSED state. In that state, all three checks *missing*, *noise*, and *out* are performed every data cycle. If the system passes all checks before a certain time has elapsed, the system switches back to the ON state. If the *missing* test is failed, or if the timer expires, the system switches to the OFF state. The duration of the timer depends

on the nature of the check that triggered the PAUSED state. If triggered by a failed *noise* check, the pause timer is 500 milliseconds. In the case of a failed *out* check, the timer is 300 milliseconds. The shorter value of the timer for the failed *out* check is due to the high velocities that can be generated from eye data at the edges of the screen, as explained in the following section.

The values presented here were manually optimised to reflect the intended capabilities of the system. With a large portion of the screen available to start the

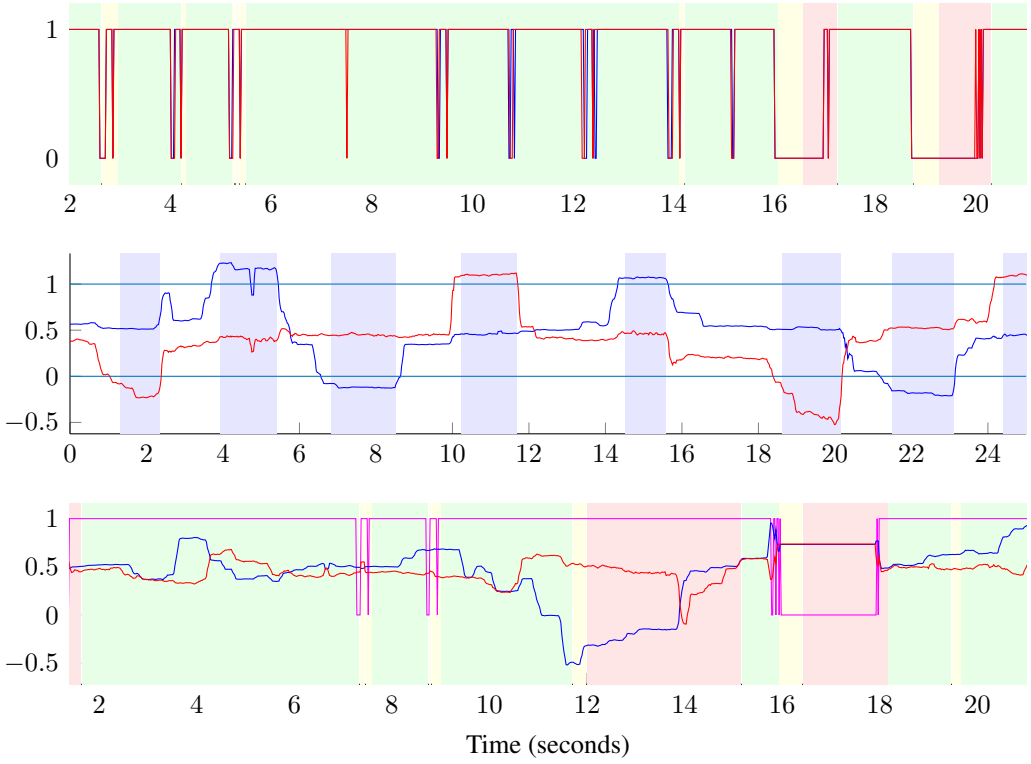


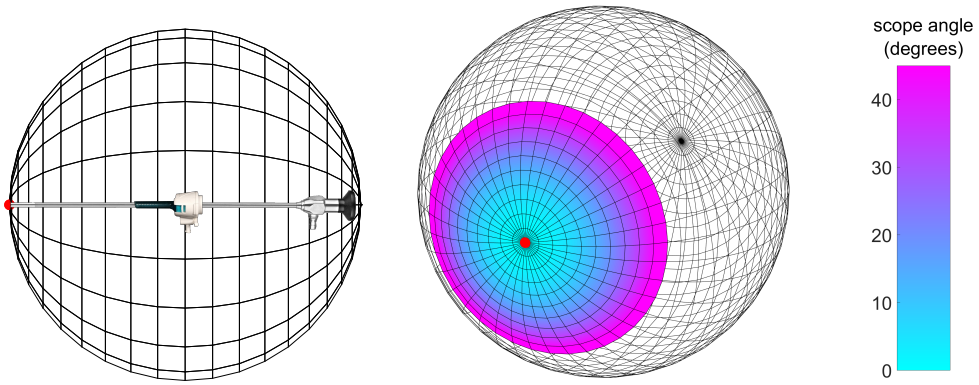
Figure 3.3: Top: blue and red lines respectively represent whether the left and right eye are visible (value of 1) or not (value of 0). Background colors represent the state of the sytem: red = OFF, yellow = PAUSED, green = ON. Center: blue and red lines represent the normalized X and Y screen coordinates of the average gaze point (averaged between the left and right eyes, on-screen values are between 0 and 1, with 0.5 being the center of the screen). Background color represent whether the user is looking inside the screen limits, white: within limits / *Out* check succeeded; blue: outside limits / *Out* check failed. Bottom: blue and red lines same as center figure. Magenta line shows if either one of the two eyes is detected (values same as top figure). Background colors same as top figure.

control, the system is easily activated by looking approximately at the center of the screen for just over half a second. In the event that the system is unintentionally activated, the noise and boundary checks will trigger a pause of the system within a few hundred milliseconds. Examples of the system behaviour in response to eye tracking signals are shown in Figure 3.3.

3.2.3 Panning Velocity Control

The interaction detection presented above allows the system to evaluate how the user wants to interact with the system. However, in order for the system to be functional a strategy needs to be developed to control the motion of the laparoscope. A remote centre of motion (RCM) was implemented to force the stereoscope to respect the trocar position. The subsequent panning motion around the RCM was implemented using an angular velocity controller, as illustrated in Figure 3.4. Practical considerations regarding the implementation of the RCM related to the panning and zooming functionality are further discussed in section 3.3.2.

When the system is enabled (ON state), the eye data is used to generate two-dimensional vectors. The centre portion of the image is a zero velocity zone, any points within this zone will only slow down the camera. Points outside of



(a) (b)

Figure 3.4: Illustration of the RCM constraint. (a) Representation of the RCM constraint around the trocar (centre). The red dot represents the initial scope position. (b) Representation of the range of motion of the scope around a fixed RCM. The colour gradient indicates the magnitude of the angle of rotation away from the initial position (illustrated by the red dot).

this zone will generate a vector as shown in Figure 3.5. The resulting vector is normalized according to screen dimensions, and scaled proportionally to the maximum allowable angular speed as shown in equation 3.1:

$$\boldsymbol{V_{2D}} = \begin{cases} V_{a_{max}} \cdot \left[\frac{1}{\sqrt{2}(0.5-R)} \cdot \left(\boldsymbol{V_t} - R \cdot \frac{\boldsymbol{V_t}}{\|\boldsymbol{V_t}\|} \right) \right]^{\alpha} & , \text{ if } \|\boldsymbol{V_t}\| \geq R \\ 0 & , \text{ else} \end{cases} , \tag{3.1}$$

where $\boldsymbol{V_t}$ is the gaze vector expressed in centred normalized screen coordinates (values ranging from -0.5 to 0.5, where 0 is the centre of the screen), R is the radius of the zero velocity zone in the same coordinates, and $V_{a_{max}}$ is the maximum reachable angular velocity. An additional scaling parameter $\alpha = 0.5$ is introduced to more rapidly increase the angular velocity away from the zero velocity zone. The resulting angular velocity fields are illustrated in Figure 3.6. The end result is transformed to a rotation vector with non-zero components along the camera X(pitch) and Y(yaw) axis. This rotation vector is then set as the desired velocity for an acceleration-constrained 3 degree of freedom velocity planner. Every control cycle of the robotic arm, the current rotation vector is obtained from the

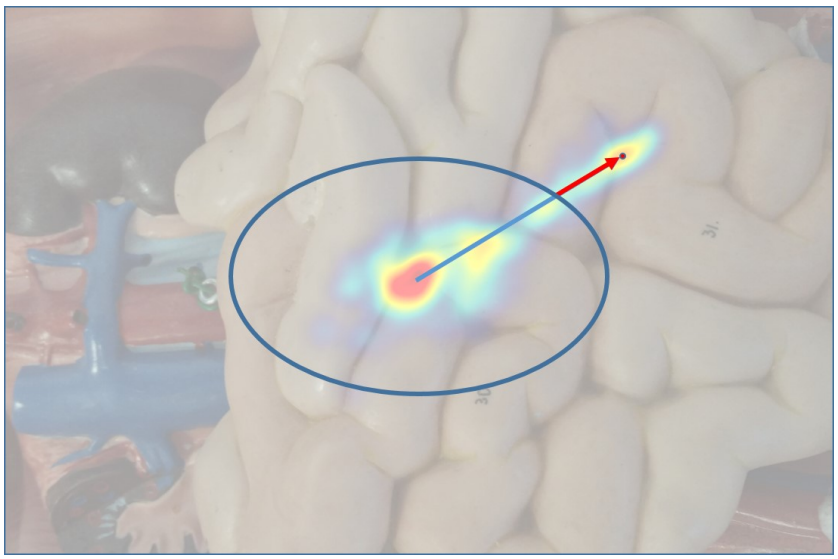
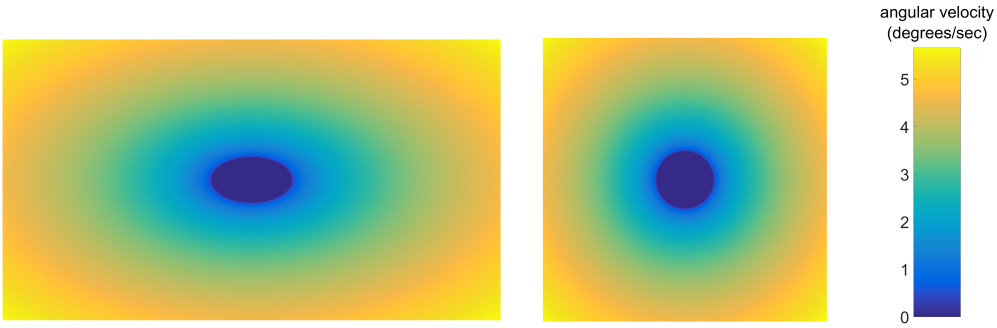


Figure 3.5: Generation of velocity vectors based on eye data. The blue ellipse illustrates the boundaries of an example zero velocity zone. The red arrow represents the two-dimensional vector used to generate desired angular velocities. The overlaid heatmap shows typical eye motions in this direction.



(a) (b)

Figure 3.6: Visualisation of the velocity field in (a) standard screen coordinates, and (b) in normalized screen coordinates.

velocity planner and used to update the robot position and orientation.

This approach presents several advantages compared to [76] and [89]. The use of task space velocity control offers more intuitive movements than those of a joint by joint position control based on image quadrants. However, unlike in [89] no discrete velocity zones are used, allowing for continuous transitions between desired velocities. The capacity to enforce desired angular accelerations also further increases the controllability of the system, such as by allowing faster braking when the system enters the OFF state.

Coupled with the interaction detection presented in section 3.2.2, the velocity planner offers an intuitive and robust platform for the control of the stereoscope. Due to the generation of angular velocity and acceleration profiles, unintentional system activation will typically not result in any perceptible movement before the system pauses the control.

3.2.4 Gaze-Centric Stereo Reconstruction

In this paper, eye tracking is combined with stereo reconstruction to transform two-dimensional screen coordinates into corresponding 3 dimensional robot base coordinates. The proposed system uses the stereo reconstruction described in [144]. While the running time of this algorithm is typically over five seconds for high definition images, the input image size can be considerably reduced by considering only the area surrounding the gaze fixation point. Combined with image scaling, this allows the system to run the targeted stereo reconstruction algorithm

at 500 ms per frame. This update rate is largely sufficient for the requirements of the proposed system, as the robot planning allows for smooth, non-jerky movements.

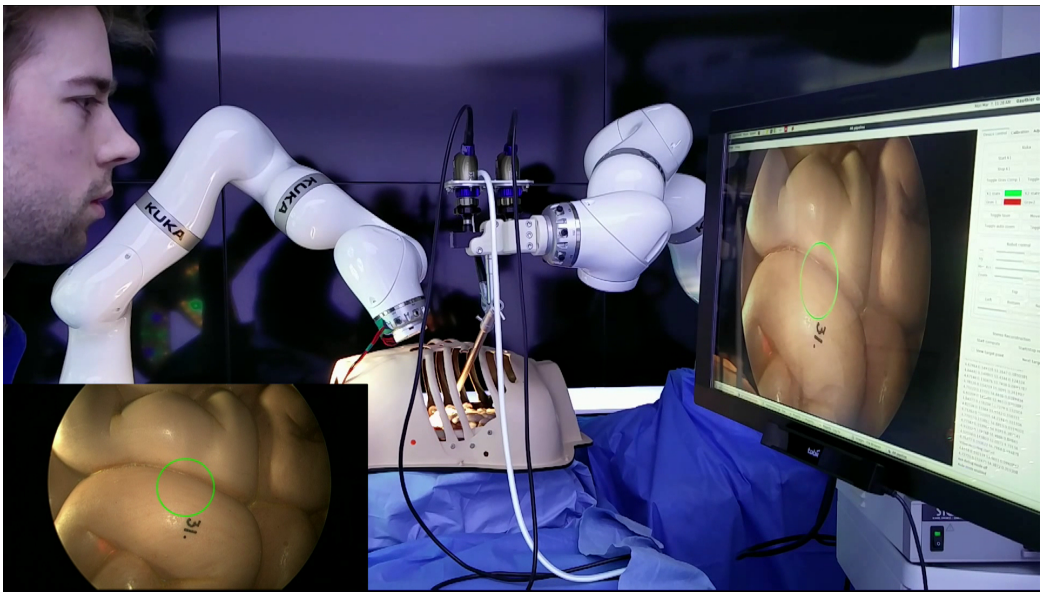
The information obtained from the stereo reconstruction is used to update a proportional controller adapting the translation of the scope along its optical axis (zoom). In the proposed system the controller tried to maintain a tissue-to-camera viewing distance of 60 millimetres. This distance was chosen based on the focal length set on the calibrated stereoscope during the studies.

3.2.5 Interface Design

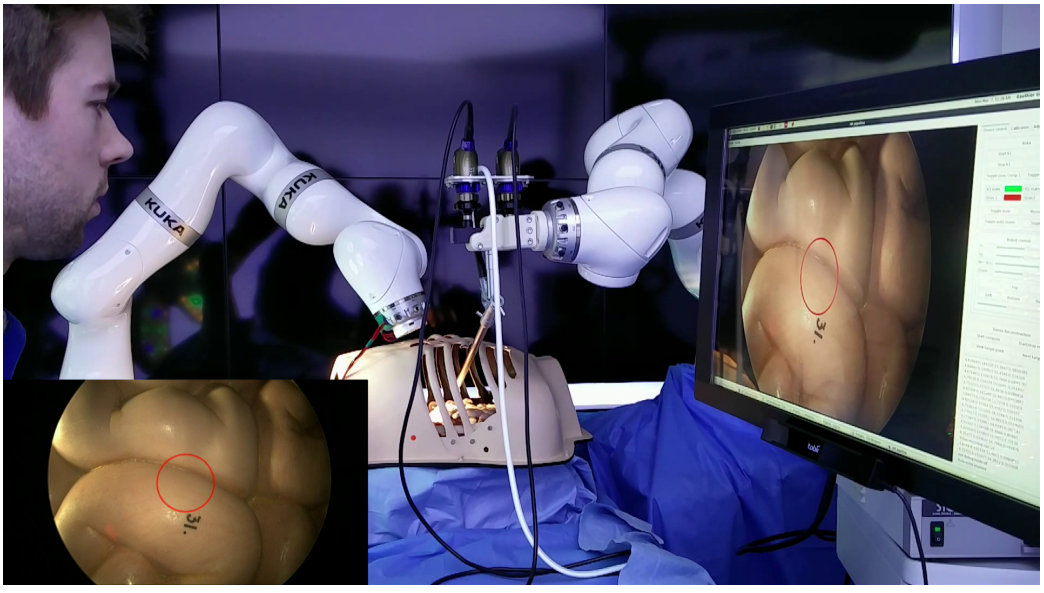
Using the interaction detection and velocity planning presented previously, a user is able to operate the system completely hands free. As mentioned in section 3.2.2, the system is activated by briefly fixating the zero velocity zone in the centre of the screen. The user receives visual feedback of the activation via the display of a green circle in the centre of the screen, shown in Figure 3.7(a). This circle fades away over a duration of 800 ms. Likewise, deactivation of the system is signalled with a red fading circle, shown in Figure 3.7(b).

3.3 Control and Calibration of System Components

This section covers key practical considerations necessary to obtain a high-accuracy, well calibrated robotic system capable of operating in a realistic scenario. While some of these considerations are well known to the robotics community, such as camera calibration and RCM implementation, others are often much less present in the literature or have required noteworthy modifications for the applications considered. Additionally, some of the techniques presented below are used repeatedly throughout the work presented in this thesis, and introducing them here allows the introduction of key notation. As such, the sections below cover calibration and control considerations of particular interest, and redirect to the corresponding literature otherwise.



(a)



(b)

Figure 3.7: Example activation (a), and deactivation (b) of the system. The circle fades over 800 ms, indicating the system status.

3.3.1 Calibration of Robotic Arms

The two robotic arms are mounted on optical tables screwed to each other, and so their bases are considered statically linked. The setup is illustrated in Figure 3.8. Let T_{A1} be the homogeneous transformation from the robot base to the end effector for the first robot at a sampling time t_1, T_{B1} for the second robot, and T_{A2} and T_{B2} their respective transformations at a later sampling time t_2. X_1 and X_2 are the homogeneous transformations from the end effector of the first robot to the end effector of the second robot at sampling time t_1 and t_2 respectively. Finally, L is the unknown homogeneous transformation from the first robot base to the second robot base. Note that the standard notation for homogeneous transformations used in this thesis is of the form ${}^{F_1}H_{F_2}$, such that $\boldsymbol{P_{F_1}} = {}^{F_1}H_{F_2}.\boldsymbol{P_{F_2}}$, where $\boldsymbol{P_{F_1}}$ and $\boldsymbol{P_{F_2}}$ represent the coordinates of a 3D point $\boldsymbol{P}$ within the reference frame F_1 and F_2 respectively. ${}^{F_1}H_{F_2}$ is represented graphically with an arrow pointing from F_1 to F_2. However, when the notation can be shortened without loss of clarity, e.g. when those transformations are listed explicitly on a figure such as in Figure 3.8, then the shortened notations will be used for clarity. Vectors are written in bold.

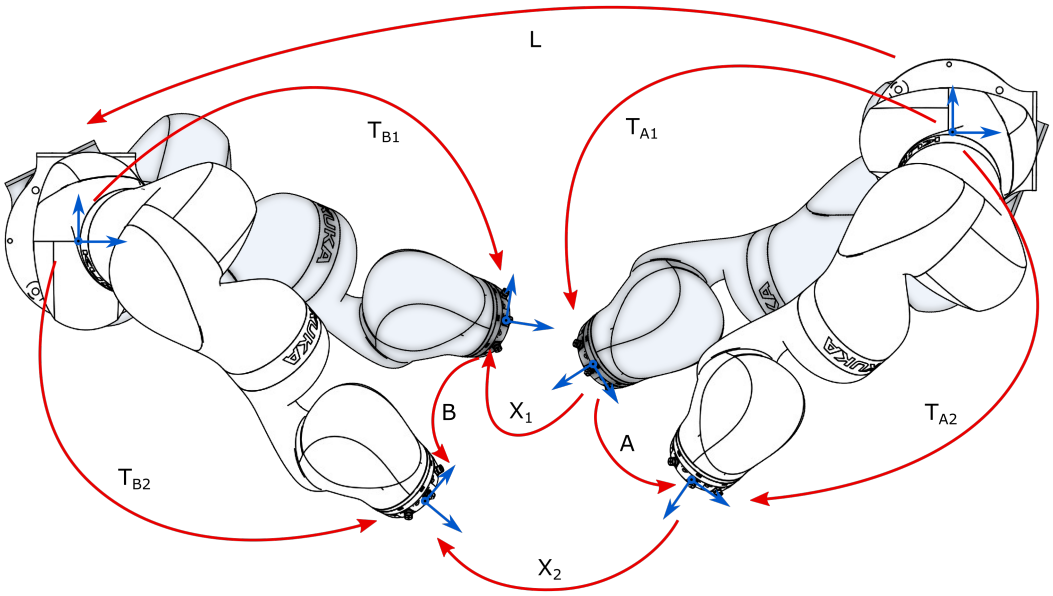


Figure 3.8: Setup for the calibration of the robotic arms.

In order to determine L, an additional source of information must be provided linking the two robot bases. One approach consists in rigidly linking the robot end-effectors together such that $X_1 = X_2 = X$. In this work, this was done by screwing both of the end effectors rigidly to a metal board. The robots are then placed into a compliant mode and moved to cover as much of the workspace as possible, recording multiple pairs of A and B for each relative motion. For a given pair of positions, such as illustrated in Figure 3.8, it is then possible to write:

$$A.X = X.B, \tag{3.2}$$

which is the standard formulation of the hand-eye calibration problem. Many methods have been developed to solve this equation in a robotic context. Some approaches such as [145] and [146] opt to solve the rotation and translation parts separately, while others such as [147] solve both simultaneously. In this work, a dual-quaternion hand-eye calibration method combining advantages from both of these approaches was implemented based on [148], ideal for this application. By solving for X using the hand eye calibration method mentioned, we can then solve for L in each of the previous measurements:

$$L = T_{A1}.X.{T_{B1}}^{-1}. \tag{3.3}$$

Multiple different measurements of L can then be combined to minimize the overall error. The proposed system was calibrated with over 4000 measurements, and resulted in an RMS error of under 0.2mm. The approach of rigidly linking the robot end-effectors is effective and highly accurate, however it does impose physical restrictions on the position of the robots relative to each other. Additionally, care must be taken to keep the robots well outside of any potential singularities, as abrupt motions from one of the robots could potentially damage both. An alternative approach would be to rigidly affix a remotely tracked marker on both robot end effectors. Each robot arm can then be calibrated individually using the same procedure as mentioned previously, except this time the resulting transformation is the one between each of the robot bases and the tracker's frame of reference. Once these transformations are known, it is trivial to find the transformation between the two robot bases. This approach, however, is more lengthy, requires additional hardware, and introduces additional sources of errors.

3.3.2 Camera Calibration and Remote Centre of Motion Considerations

In order to transform the desired panning motions from the camera frame to the robot frame, the static transformation from the robot end effector to the camera frame must be determined. This is illustrated in Figure 3.9. Additionally, the camera intrinsic properties must obtained to perform the stereo reconstruction necessary for the zooming functionality.

The stereoscope was calibrated using standard camera calibration techniques through the OpenCV library [149], yielding a root mean square (RMS) reprojection error of under 0.2 pixels. The same dual-quaternion method implemented in section 3.3.1 was used to find the transformation between the camera frame of reference and the robot base frame. The RMS error for the hand-eye calibration was sub-millimetre. To keep the transformation ${}^{K}T_{C}$ constant throughout the use of the system, the scope was rigidly mounted on the robot end effector, and any degrees of freedom of the scope were constrained (some scopes have a flexible end).

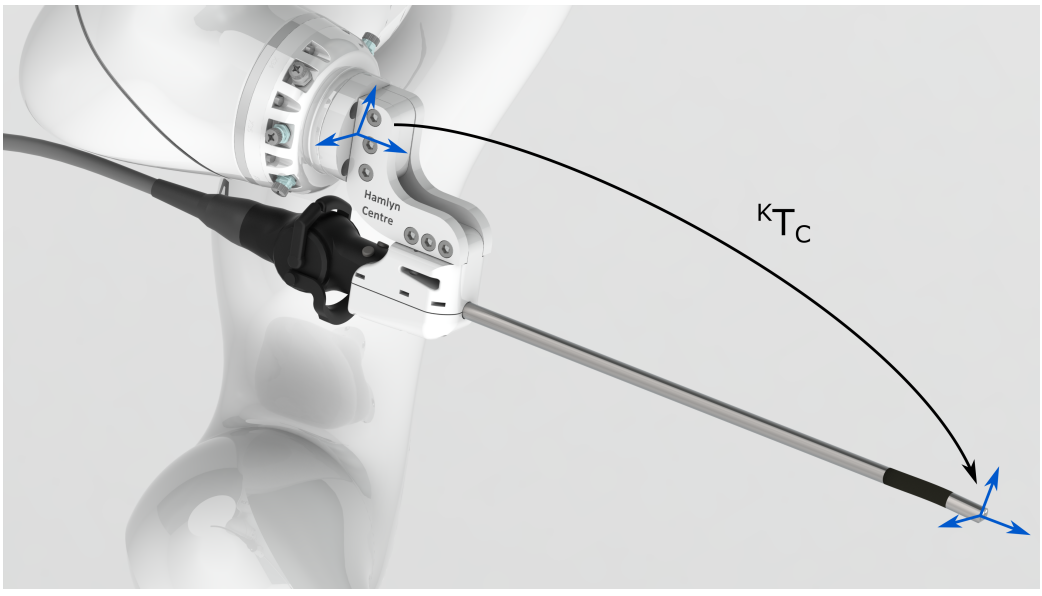


Figure 3.9: Stereoscope rigidly affixed on a robot end effector. The hand-eye transformation ${}^{K}T_{C}$ connects the robot end effector frame K with the camera frame C.

A few practical considerations related to the RCM can be noted here. Let ${}^{B}T_{K}$ be the transformation from the robot base B to the robot end effector frame K. Let also ${}^{K}\boldsymbol{V}_{trocar}$ be the trocar position of the robot end effector frame at initialization. Let the rotation and translation of a homogeneous transformation follow the same writing convention as the homogeneous transformation. The transformation from the robot base to its end effector at initialization can then be expressed as

$$ {}^{B}T_{K_i} = \left[{}^{B}R_{K_i}, {}^{B}\boldsymbol{V}_{K_i} \right]. $$

The position of the RCM in the base frame is ${}^{B}\boldsymbol{V}_{RCM} = {}^{B}T_{K_i} \times {}^{K}\boldsymbol{V}_{trocar}$. For a given change in the desired camera orientation generated by the gaze controller ${}^{C}R_{desired}$, described in section 3.2.3, the new robot end effector orientation ${}^{B}R_{K_{new}}$ is expressed below:

$$ {}^{B}R_{K_{new}}.{}^{K}\boldsymbol{R}_{C} = {}^{B}R_{K_i}.{}^{K}\boldsymbol{R}_{C}.{}^{C}R_{desired}, \tag{3.4} $$

$$ {}^{B}R_{K_{new}} = {}^{B}R_{K_i}.{}^{K}\boldsymbol{R}_{C}.{}^{C}R_{desired}.{}^{K}\boldsymbol{R}_{C}{}^{-1}. \tag{3.5} $$

To keep the same point of the laparoscope precisely at the RCM, the new robot end effector position ${}^{B}\boldsymbol{V}_{K_{new}}$ is expressed as:

$$ {}^{B}R_{K_{new}}.{}^{K}\boldsymbol{V}_{trocar} + {}^{B}\boldsymbol{V}_{K_{new}} = {}^{B}\boldsymbol{V}_{RCM}, \tag{3.6} $$

$$ {}^{B}\boldsymbol{V}_{K_{new}} = {}^{B}\boldsymbol{V}_{RCM} - {}^{B}R_{K_{new}}.{}^{K}\boldsymbol{V}_{trocar}. \tag{3.7} $$

The translation along the shaft the scope, respecting the RCM, can be expressed by replacing ${}^{K}\boldsymbol{V}_{trocar}$ by ${}^{K}\boldsymbol{V}_{trocar}+\alpha.{}^{K}\boldsymbol{V}_{ShaftDir}$ in equation 3.7, where ${}^{K}\boldsymbol{V}_{ShaftDir}$ is the constant normalised vector indicating the direction of the shaft in the end effector frame of reference, $\alpha \in \mathbb{R}$. However, the direction of translation required to perform the zooming motion, also expressed in the end effector frame of reference, is ${}^{K}\boldsymbol{R}_{C}.{}^{C}\boldsymbol{V}_{SightDir}$, where ${}^{C}\boldsymbol{V}_{SightDir}$ is the normalised vector orthogonal to the camera plane in the camera frame of reference. This important consideration means that, while it is always possible to translate along the shaft of the scope, this translation will only result in a pure translation along the viewing axis if and only if ${}^{K}\boldsymbol{V}_{ShaftDir} = {}^{K}\boldsymbol{R}_{C}.{}^{C}\boldsymbol{V}_{SightDir}$.

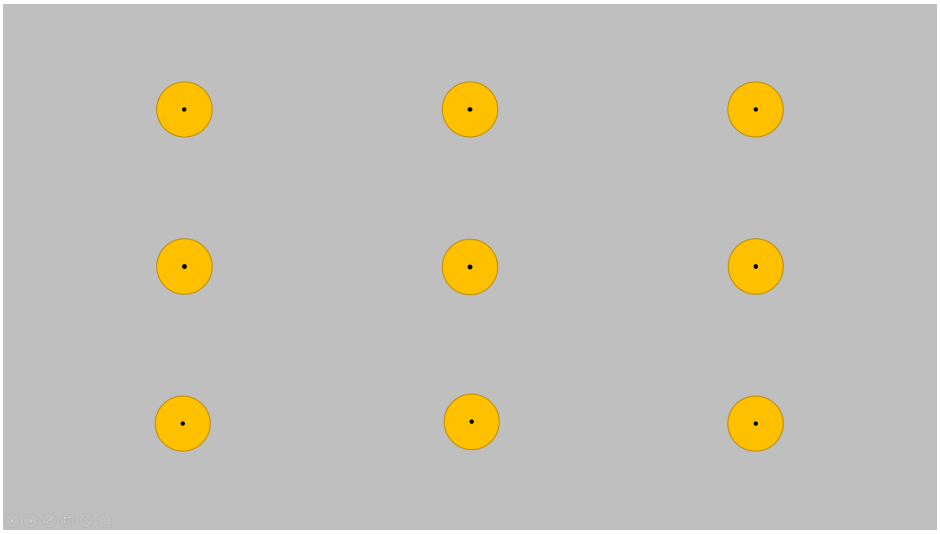


Figure 3.10: Example eye tracking calibration pattern.

3.3.3 Eye Tracker Calibration

Commercial eye trackers such as the one used in this work are prone to user-specific errors in gaze tracking. As such these products typically require an additional calibration procedure to be performed by users prior to working with the eye tracker. In this thesis, a thin plate spline (TPS) approach is used to compensate these errors. The TPS method presented in this work effectively replaces the commercially provided additional calibration procedure with one of our own. Further details on how the TPS is implemented as a mapping function can be found in appendix A. Additionally, a study quantifying calibration drift over time is detailed in appendix B.

The calibration procedure consists in the user looking at a series of points shown sequentially on the screen. These points are chosen so as to cover the areas of interest of the screen, such as shown in Figure 3.10. As mentioned previously, normal gaze behaviour consists in using the eyes to observe, not to fixate arbitrary points. This is apparent even in this simple task, as users will find the calibration procedure tiring if done several times in a row. To alleviate this issue, the yellow part of the points was made to pulsate. Although not quantified, that small amount of motion seemed to help users focus better on the task, and generated positive feedback.

3.3.4 Laser Calibration

The complete system was first tested by firing the laser held by the second robot at a 3D reconstructed point obtained from the stereoscope held by the first robot. It was initially thought that the laser could be assumed to fire straight along the end effector direction of the robotic arm. However, this assumption quickly proved to lead to a significant loss of accuracy when the laser was fired at a distance further than a few centimetres. This was due to imperfections in the laser manufacture, as well as in the mount holding the laser diode. As such, a method was designed to compensate for the fact that the laser did not fire precisely along the Z axis of the robot end effector. Figure 3.11 shows the setup used to calibrate the laser direction.

The calibration of the laser direction is a two step process. The initial step consists of calibrating a pointed tool attached to the second robot's end effector (Figure 3.11b) to find the coordinates of the pivot point in the base frame of the robot (Figure 3.11a). The second step consists of mounting the laser on the second robot and manually aligning it with the pivot point in multiple different poses. Let ${}^{B}T_{K}$ be the transformation from the robot base to the robot end effector at sampling time t, ${}^{K}T_{L}$ the transformation from the robot end effector to the tip of the laser diode, ${}^{B}\boldsymbol{P}$ the position of the pivot point in the robot base frame, and $\boldsymbol{V}_t$ the translation from the laser tip to ${}^{B}\boldsymbol{P}$. It then follows that:

$$\boldsymbol{V}_t = {}^{K}T_{L}{}^{-1}.{}^{B}T_{K}{}^{-1}.{}^{B}\boldsymbol{P}. \tag{3.8}$$

The firing direction of the laser is then obtained by normalizing $\boldsymbol{V}_t$ and averaging it over multiple measurements. While taking into account the laser direction substantially improved the accuracy of the system, a remaining error of up to 5 millimetres in worst cases persisted. It was hypothesized that this error was due to the accumulation of calibration errors spread throughout the system. Indeed, while all individual calibration errors were kept low, the full kinematic chain of the system involved a total of six possible sources of error, not counting the robot encoders:

- The calibration of the stereo camera.
- The stereo reconstruction.

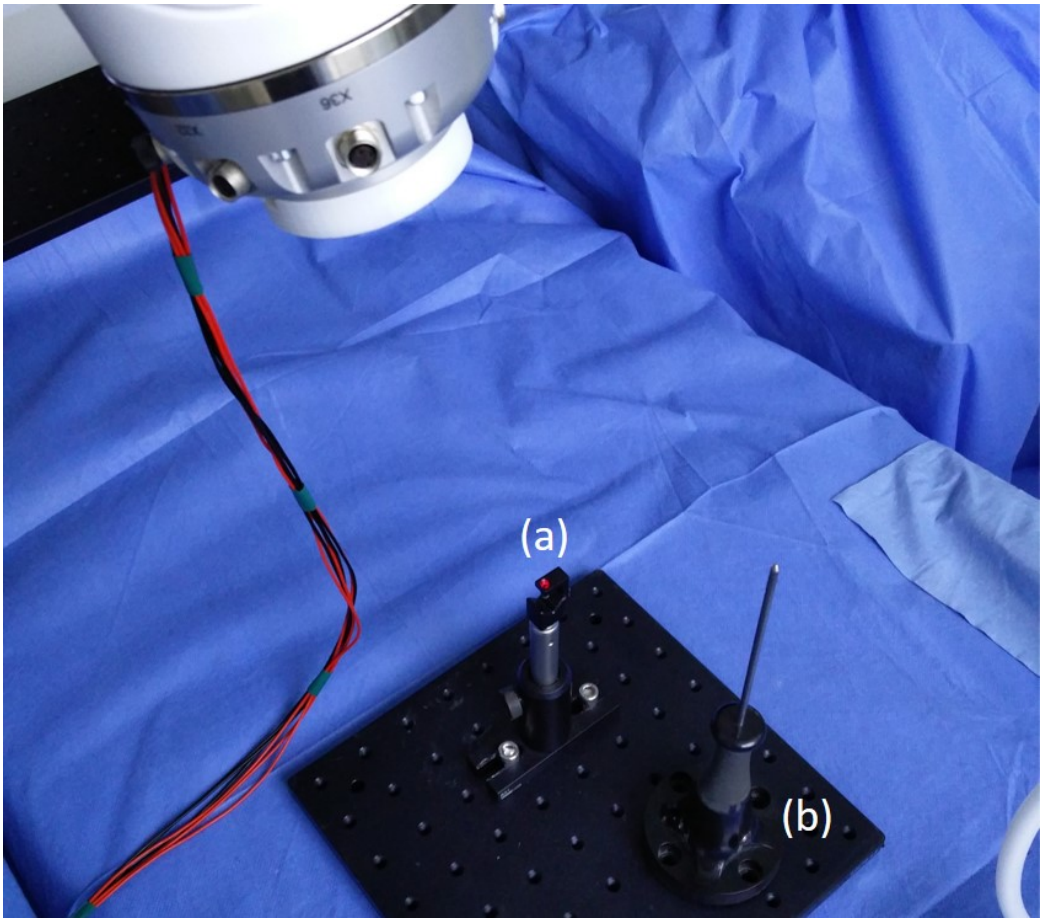


Figure 3.11: Laser calibration setup. (a) Pivot point for the tool and the laser, (b) Pivoting tool used to determine the base frame coordinates of the pivot point.

- The camera to robot 1 end effector hand-eye calibration.
- The robot 1 base frame to robot 2 base frame calibration.
- The laser mount hand-eye calibration.
- The laser beam direction.

3.3.5 Error Minimization

One final calibration step was performed to attempt to model the residual error on the entire system. The same asymmetric circles calibration grid used for the camera calibration was placed in a fixed position. The circles were identified,

triangulated, and their 3D coordinates brought into the base frame of the robot 2 using the stereoscope. Following this, the 3D coordinates of the points in the calibration grid were determined separately using the second robot holding the laser laser. This was performed by first aiming the laser at each of the four corners of the grid. Multiple measurements were made for every point, each resulting in the equation of a 3D line containing that point. The coordinates of each corner point on the calibration grid were then obtained by solving for the point of minimum distance between all of the corresponding lines. Figure 3.12 shows a picture of this compensation step.

Due to the manual adjustment of the laser on the points as well as the size of the laser dot itself, the laser pivot calibration can present submillimetre inaccuracies. The quality of these measurements were further improved by spreading the measurement error evenly across the four points. This was achieved by first finding the centre point $\boldsymbol{P}_c$ of the four points and then computing the vectors $\boldsymbol{V}_1$, $\boldsymbol{V}_2$, $\boldsymbol{V}_3$ and $\boldsymbol{V}_4$ from the $\boldsymbol{P}_c$ to each of the four corners. Figure 3.13 illustrates this process. The vectors $\boldsymbol{V}_x$ and $\boldsymbol{V}_y$ along the directions of the grid are then found by summing up the corner vectors as follows:

$$\boldsymbol{V}_x = \frac{\boldsymbol{V}_3 + \boldsymbol{V}_4 - \boldsymbol{V}_1 - \boldsymbol{V}_2}{2}, \boldsymbol{V}_y = \frac{\boldsymbol{V}_3 + \boldsymbol{V}_2 - \boldsymbol{V}_1 - \boldsymbol{V}_4}{2}, \tag{3.9}$$

From there, 3D coordinates of all the points in the calibration grid in the robot 2 base frame can be rebuilt using $\boldsymbol{P}_c$, $\boldsymbol{V}_x$ and $\boldsymbol{V}_y$. The outcome of the previous sequence is two sets of points supposed to represent the same values. Using singular value decomposition, the transformation T_{error} can be computed that minimizes the difference between the two sets of points. Applying this transformation matrix to the system resulted in bringing the error of the entire system down to below two millimetres. This was a critical threshold to reach as the limiting accuracy factor was no longer the system, but the eye tracking values.

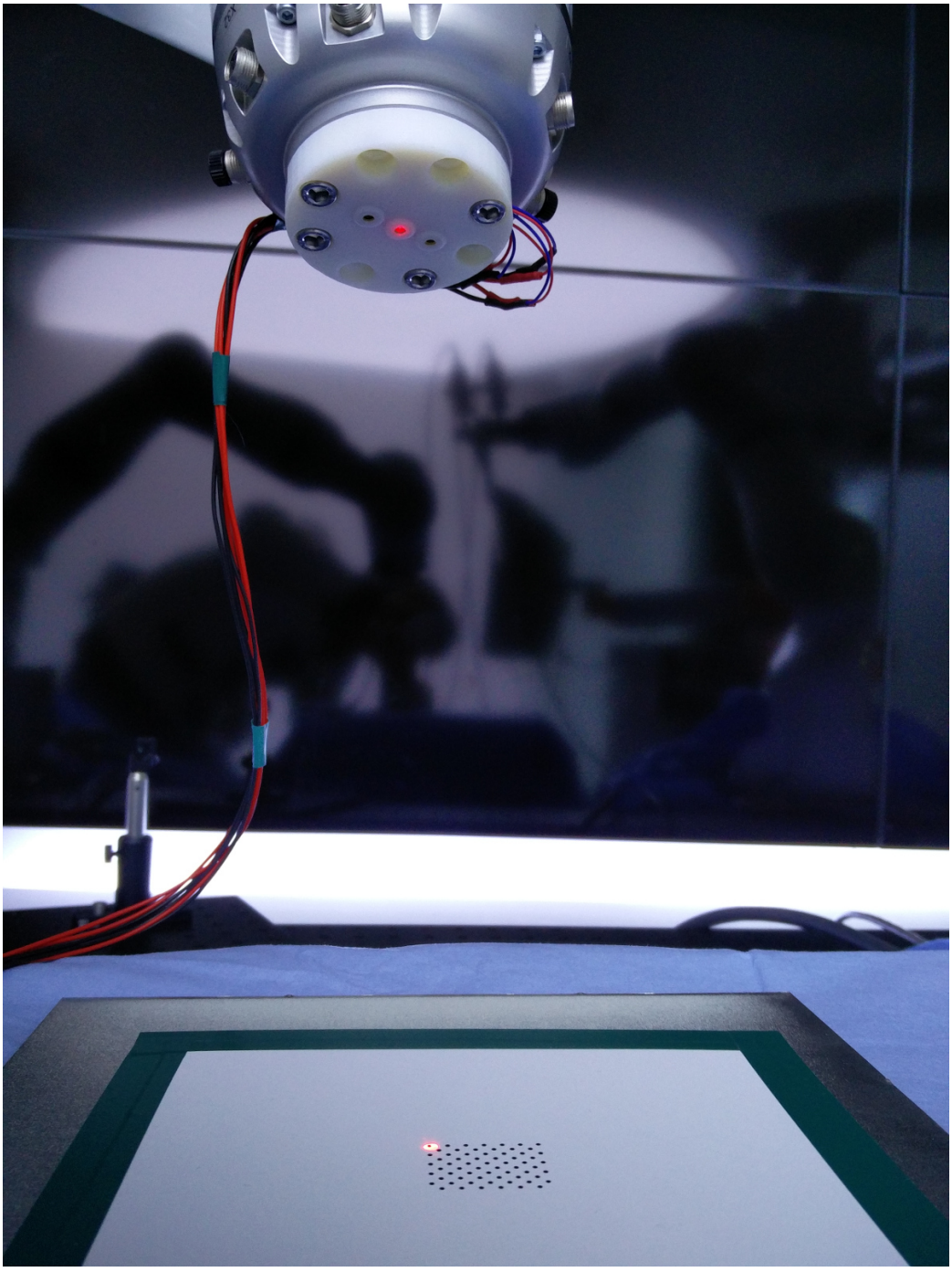


Figure 3.12: Error compensation by projection of the laser on the calibration grid.

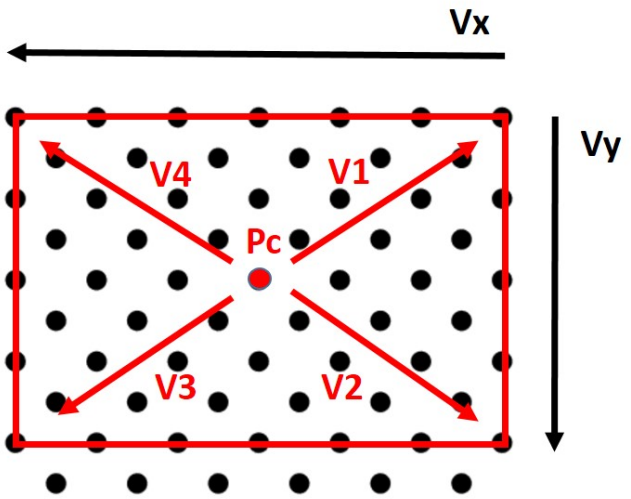


Figure 3.13: Minimisation of the measurement error through reconstruction of the ideally spaced grid points.

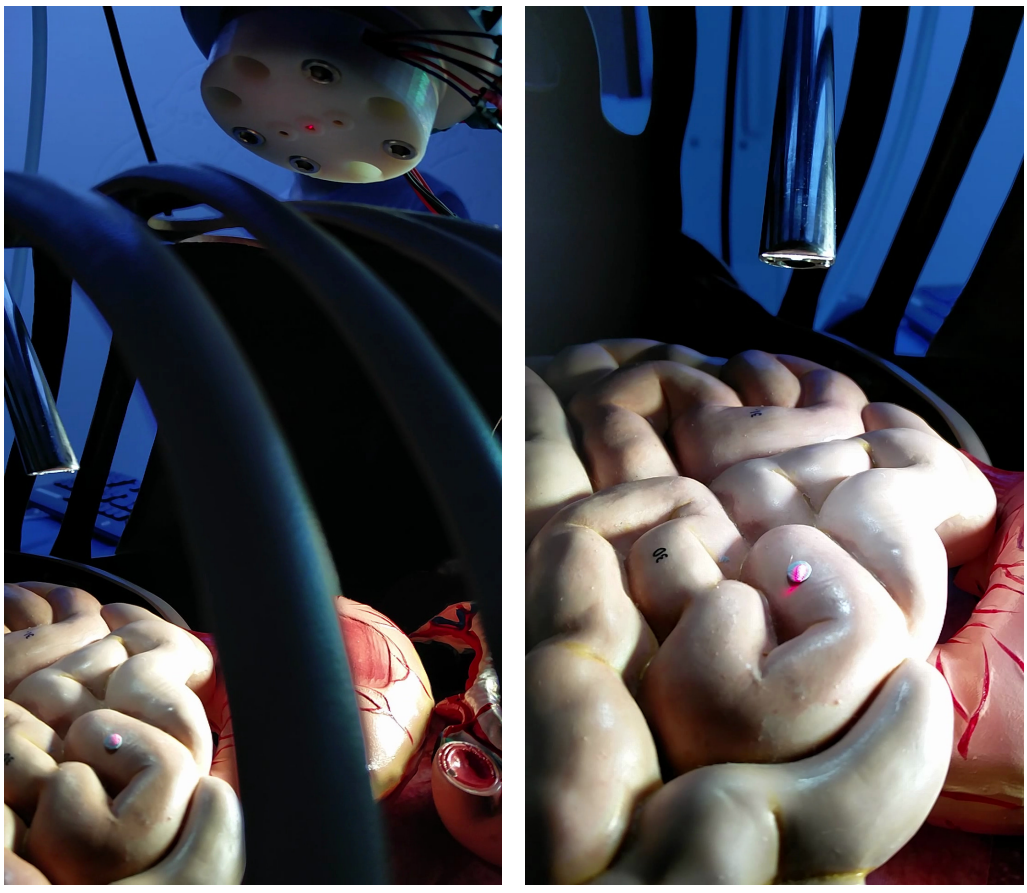
3.4 System Evaluation

3.4.1 User Study Protocol

In order to evaluate the system, an experiment was designed in which participants had to control both the stereoscope and the laser to mark four target points spread out in a phantom model at varying heights. The purpose of this study was to assess the usability of the system by determining whether it presented any improvements in terms of speed and ease of use over a manually controlled system. Participants had to bring each target point within the centre of the field of view of the stereoscope, and position the laser on top of the point. The task was performed twice for each participant, once with the eye-controlled system and once with manual control over the robotic system. The order of the tasks was randomized between participants. Manual control of the system was performed via keyboard input.

In the manual mode, arrow keys and the number pad of the keyboard were used to control the laser position and the stereoscope position respectively. The movement of the stereoscope controlled by the arrow keys was the panning motion around the RCM. Participants could ask to zoom in and out. In the event that

participants achieved a tissue-to-camera distance of over 90 millimetres the system was zoomed in for them to prevent unrealistically high rotation angles around the RCM. For the eye-controlled mode participants would perform an eye tracking calibration prior to the task. During the task they would fixate each target point until they felt they had reached a stable position, at which point they would activate the laser. If the laser was on target, the participants were free to continue to the next target. Otherwise the eye control would be paused and the user would adjust the laser on the target manually. Figure 3.14 displays images obtained during the experiments, with the laser activation on target after the automatic targeting.



(a) (b)

Figure 3.14: Pictures of the system in action: (a) automatic laser targeting and (b) automatic zooming of the laparoscope, based on the gaze-targeted stereo reconstruction.

Participants were shown all the target points and trained once with each control mode before the experiment. Lastly, participants were made aware that they should attempt to minimize the amount of time the laser was on if possible. In a real case a laser pointer could be used to indicate where the laser is going to fire, but it would still be distracting for the surgeon to have to constantly keep the laser pointer on. A total of 8 non-clinical participants, 5 male and 3 female, took part in the study. The different metrics considered were:

- The relative time taken to reach each target point.
- The task completion time.
- The total time with the laser activated.
- The tissue-to-camera distance.
- In eye control mode, the amount of time the system had to be disabled to manually adjust the laser.

3.4.2 Results

The results of the experiment for the core metrics are displayed in Figures 3.15 and 3.16. These boxplots compare the performance of the manually controlled system against the gaze-controlled one. Additionally, numerical values for the mean, median, and statistical significance of these metrics are listed in Table 3.2.

The time required to reach each point from the previous one is displayed in Figure 3.15. As can be seen, these times show a significant improvement for every point other than the last one. This can be explained by the fact that the two last points were close to each other such that it required only minimal movement from the stereoscope and the laser to reach it. Although the median times for the last point show a large improvement in favour of the gaze-controlled system, the mean and quartiles shown in figure 3.15 highlight that participants experienced varying levels of improvement over the manually controlled system for such close points. Averaged over all four points the median time improvement is 53.9%.

Figure 3.16 shows boxplot results for the three other core metrics: task completion time, total laser activation time, and average tissue-camera distance. The overall task time shows a clear improvement in usability with a 52.92% reduction

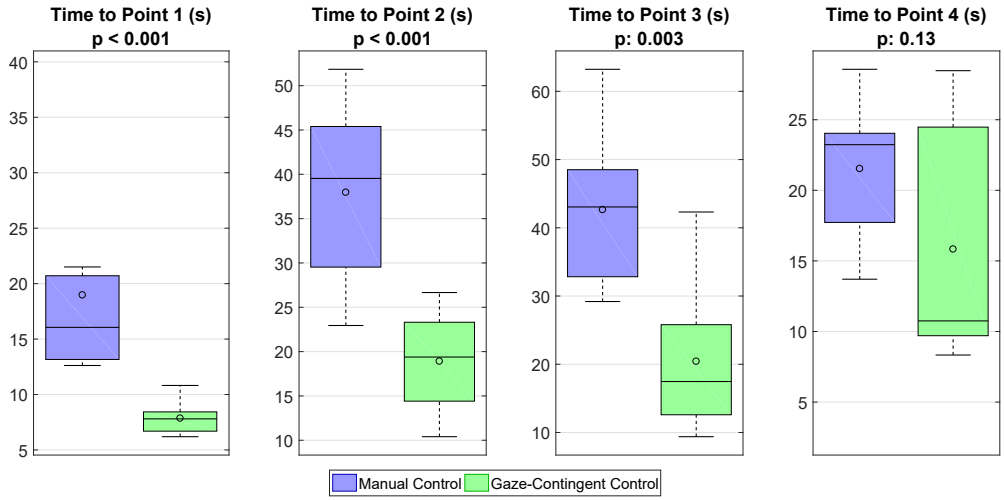


Figure 3.15: Experimental results: Time to reach each target point from the previous one.

of median task completion times for the gaze-controlled system relative to the manually controlled system. Unsurprisingly, the total laser activation time shows a significant 90.6% median reduction for the gaze-controlled system relative to the manually controlled system, as it was difficult for participants to assess where the laser target position without visual feedback.

The tissue-to-camera distances show a 20.15% median reduction for the gaze-

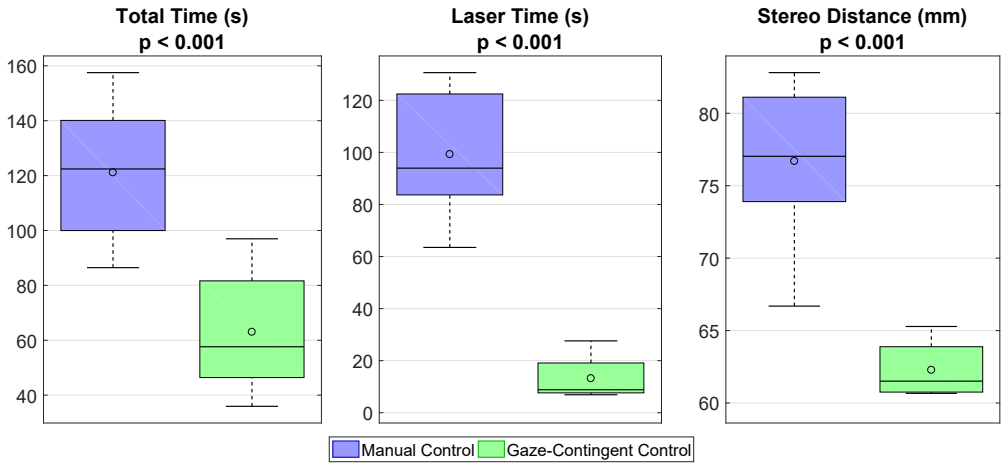


Figure 3.16: Experimental results: Total task time, total laser activation time, and tissue-to-camera distance.

	Mean		Median		p-value
	Manual	Gaze	Manual	Gaze	
Time to point 1 (s)	18.97	7.86	16.05	7.80	1.55×10^{-4}
Time to point 2 (s)	37.96	18.91	39.54	19.39	6.21×10^{-4}
Time to point 3 (s)	42.65	20.43	43.05	17.47	2.95×10^{-3}
Time to point 4 (s)	21.53	15.83	23.23	10.75	0.131
Total task time (s)	121.1	63.03	122.4	57.63	6.21×10^{-6}
Total laser use (s)	99.23	13.20	93.95	8.82	1.55×10^{-4}
Camera-tissue distance (mm)	76.70	62.29	77.03	61.51	1.55×10^{-4}

Table 3.2: Mean, median, and statistical significance results from the laser ablation study.

controlled system relative to the manually controlled system. Importantly, this highlights that the stereo-guided zoom controller is capable of stabilizing the system around the desired 60 millimetres camera-tissue distance, with a median camera-tissue distance of 61.51mm using the gaze-controlled system. In the manual mode, however, the median tissue-camera distance of 77.03mm represented a 28.38% increase over the desired 60mm distance, causing a slight loss of image focus and increasing the difficulty to visualise the target points properly. Note that, while participants were advised to retain the same camera-tissue distance for their own sake when using the manual system, this was not a strict requirement. Lastly, the mean time across all participants that the eye controller had to be turned off to manually position the laser dot on the target was 2.87 seconds. This time, associated with a positional error of the laser, represents less than 5% of the median task time for the gaze-controlled approach. As most users did not resort to using the manual mode when using the gaze-controlled system, this error can be mostly attributed to the quality of the eye tracking calibration prior to the experiment.

3.5 Discussion and Conclusion

This chapter proposes a novel gaze-contingent control framework for focussed energy delivery. Using only natural gaze behaviour, the proposed system can be activated and de-activated implicitly without relying on artificial gestures. Furthermore, the system is robust to noise, can recover from brief interruptions, and possesses redundant layers of built-in safety to prevent it from unintentionally moving in an uncontrolled manner. The proposed system exploits the synergy between eye tracking and stereo reconstruction to perform accurate 3D gaze tracking in real-time. This enables the use of an automated zoom functionality for the surgical camera, and automatic targeting of the focussed energy delivery system using detailed calibration techniques. Finally, a user study is conducted to evaluate the usefulness of the presented approach.

Results from the study show a clear improvement in task execution times. By taking advantage of the gaze-contingent control, users were able to effortlessly drive the robotic system to the desired locations and reposition the laser accordingly. The automatic repositioning of the laser further helped reduce the time necessary to complete the task, as the high calibration accuracy of the system meant that users rarely had to re-adjust the laser pointer manually. Although not quantified, user feedback was also highly positive as the task is simply much quicker to perform using the gaze-controlled system. A criticism can be made regarding the choice of the manual interface. For example, instead of a keyboard, a gamepad could have been used for example. However, the use of a gamepad would have introduced an additional variability factor in the study as some participants might not have been familiar with how to use one, unlike a keyboard. Additionally, as noted in 2.2.2, gamepads are not suitable for the control of dual-instrument systems, which this framework is ultimately designed for. While yet other choices could have been made for the manual interface, the key limitation consisted in the fact that it was difficult for participants to control both the camera and the laser pointer at the same time. This limitation is simply due to the difficulty in controlling the laser pointer while simultaneously moving the camera, and is not something which conventional interfaces present a solution to. Instead, master-slave systems such as the daVinci move the camera separately from the instruments, validating the conditions of this experiment.

This study also brings notable insights regarding for which situations the gaze-contingent control is most beneficial. In particular, the gaze control seems most beneficial when the operative workspace is sufficiently large to justify camera movement. As mentioned in section 3.4.2, the results show only marginal improvement for successive target points located close to each other. This is because participants controlling the system in manual mode can opt to only move the camera by a small amount, or indeed not at all, before adjusting the laser pointer. While the gaze-controlled system does not present a disadvantage in those situations, its usefulness is understandably minimized.

Finally, some limitations regarding the evaluation study can be noted. While the users were more performant in terms of time to reach the target points, the performance of the system during bimanual surgery was not assessed. The unquantified bimanual performance means that it is possible that some system parameters would have to be adjusted to provide ideal performance during surgery, e.g. reduce or increase the zero velocity zone or acceleration parameters. However, the strong performance of similar techniques for bimanual surgery, such as in [150], suggest that this type of approach should not suffer from fundamental limitations for these tasks. Lastly, cognitive load was not assessed during the task, and users were free to concentrate on the proposed task with no distractions.

This chapter provides an initial framework for the gaze-contingent robotic work in this thesis. It establishes key concepts regarding implicit interfaces, and presents tools and methods used in the following chapters. However, the work presented here is mainly limited to improving the control of the laparoscope and using the 3D gaze to automatically target the laser. Current robotic systems possess many additional challenges regarding the control of complex surgical instruments, such as workspace limitations and un-intuitive behaviour near joint limits. As shown, eye tracking data lends itself well to use with other sources of information such as scene depth. It is particularly helpful to determine the user's intention during a task, especially when other sources of data are available to enhance the system's understanding of the scene. The following chapter takes full advantage of this to tackle some of the longest-dating issues in master-slave systems, namely limitations due to workspace mismatch between slave and master devices, and the resulting need for clutching.

4

Intention Recognition for Motion Assistance

4.1 Introduction

The previous chapter explored the limitations of existing camera control in robotic systems, and implicit gaze-contingent solutions were presented to improve the control of the camera. Additionally, the previous chapter highlighted how eye tracking can be combined with other sources of information, in this case 3D scene reconstruction, to provide useful functionality implicitly such as automatic camera zooming and laser targeting. This chapter expands on this notion, using eye-tracking to infer scene context and implicitly improve the control of the surgical instruments.

Recent advances in surgical robots have provided surgeons with enhanced dexterity, consistency and accuracy for a range of minimally invasive procedures [151]. The development of complex wristed instruments for minimally invasive surgery (MIS) has greatly contributed to the success of these systems. However, traditional robotic systems such as the da Vinci® (Intuitive Surgical Inc., CA, United States) typically rely on supporting robotic arms for the triangulation of

Content from this chapter was published and is reproduced with permission from,

Implicit gaze-assisted adaptive motion scaling for highly articulated instrument manipulation. Gras G., Leibrandt K., Wisanuvej P., Giataganas P., Seneci C.A., Ye M., Shang J., Yang G-Z. In IEEE International Conference on Robotics and Automation 2017. ©2017 IEEE.

these instruments. This is effective for certain categories of procedures such as in general abdominal surgery, where a large workspace is available and the instrument shafts can come through distant entry ports. However, this approach can present significant drawbacks when operating in tight spaces or along narrow anatomical pathways [152, 153]. In cases such as Transanal Endoscopic Micro-Surgery (TEMS), Ear, Nose and Throat (ENT) surgery, and keyhole neurosurgical approaches the instrument shafts are constrained to be nearly parallel, and as such the supporting robotic arms cannot offer the triangulation necessary [154].

In order to increase the dexterity of the robot, highly articulated instruments with an inherent capability to bend can be developed. This is achieved by increasing the number of degrees-of-freedom (DoF) available to the instruments, effectively increasing their dexterous workspace. However, to keep transmitted forces high, the joint segments should remain short, leading to joints often operating close to or at their joint limits. The control of these instruments thus needs to take particular care of the kinematic challenges when operating near the joint limits [155, 156]. Previous work on highly dexterous instruments has studied the control challenges of such devices in single access surgery [157, 158, 159]. This work attempts to merge the advantages of these approaches with those associated with the use of external robotic arms. A control scheme for novel 6-DoF instruments possessing a flexible elbow mounted on robotic arms is presented in the first part of this chapter. The inverse kinematic approach used to control a combined total of 9-DoF per robot instrument is detailed, placing particular emphasis on maintaining dexterity close to the joint limits.

In addition to the mechanical and kinematic limitations outlined above, existing ergonomic issues also limit several of the main advantages of robotic systems. In systems such as the da Vinci the surgeon is able to control the camera directly but is required to clutch the instruments before being able to do so, thus preventing simultaneous control of the instruments and the camera. Similarly, while motion scaling allows the surgeon to perform fine and precise movements it also prevents the surgeon from making larger movements without having to clutch and reposition the master device end-effectors. This problem is particularly noticeable when using master devices with a smaller workspace. Combined, these issues can make navigating around the operating site time consuming and frustrating by requiring the user to clutch the instruments and the camera multiple times.

This chapter builds on the previous chapter's work in intention recognition for camera control to introduce the novel concept of gaze-assisted adaptive motion scaling. Eye-tracking data is combined with hand motion information obtained from the master device, and tool tracking information, to infer where the user is trying to move the instruments. The scaling factor of the master device is modulated accordingly, allowing the user to quickly reach distant locations while retaining precise control for delicate manipulations. Unlike in [59] where an eye-tracking approach is used to provide haptic guidance towards a visual point of interest, no external forces or motions are generated using this approach. Finally, a user study is conducted to evaluate the effectiveness of this approach, showing a clear reduction in task execution times and number of times clutched.

4.2 A Highly Dexterous Robotic System for Surgery

The robotic system presented in this chapter was developed to combine the advantages of highly articulated instruments with external robotic arms. This allows the system to take full advantage of the additional dexterity provided by external arms in large workspaces, while still being completely capable of operating in narrow anatomical corridors where both instrument shafts need to be parallel. The surgical instruments were developed as part of a system for transanal micro-surgery in collaborative work. This system, named "micro-IGES", is described in detail in [160] and [159]. However, the control of these instruments was developed in close collaboration with the author of this thesis, and is key to the work that follows and its motivation. As such, this section first presents the design of these instruments, followed by a full overview of the design, kinematics, and workspace of the complete system.

4.2.1 Articulated Instruments for Single Port Surgery

The micro-IGES system [160] is a compact single port system for transanal surgery. It is designed to be able to work in long and narrow anatomical pathways, where the shafts of both surgical instruments are parallel. Unlike in other robotic systems such as the da Vinci, this requires that the surgical instruments be able to

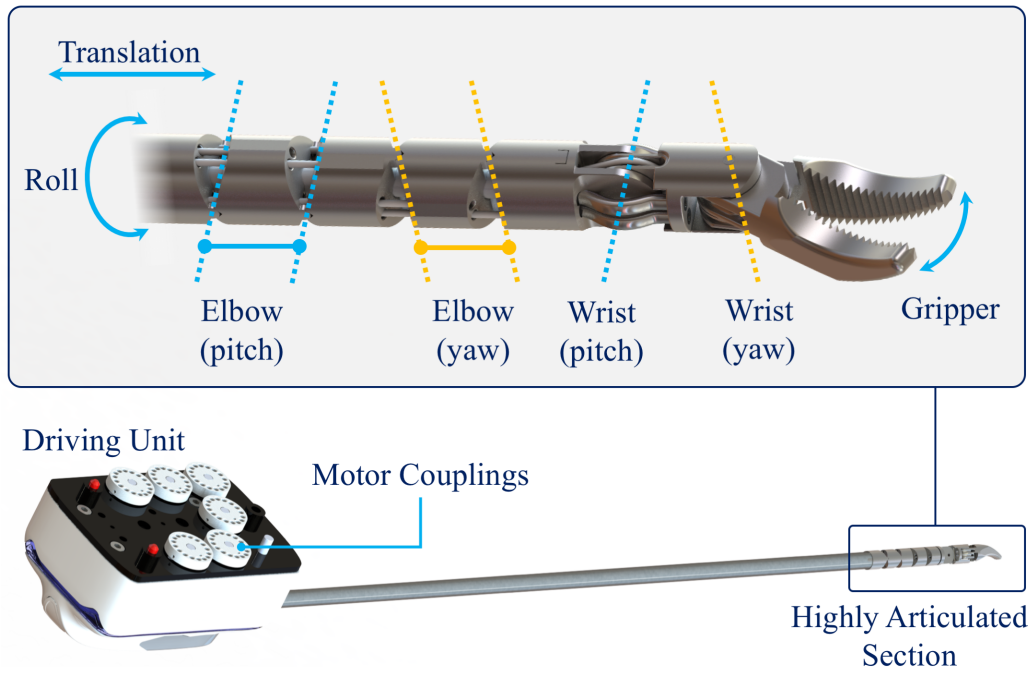


Figure 4.1: Rendering of the 6-DoF highly articulated surgical instrument. Reproduced from co-authored publication with permission from Springer [160].

offer sufficient dexterity to the user to comfortably operate, without relying on external robotic arms for triangulation. To achieve this, highly articulated instruments were designed providing a total of 6 DoFs. The joints of the instruments are designed to provide an "elbow" behaviour. This allows both instruments to self-triangulate, i.e. to come in at different angles from each side of the target anatomy, even though the instrument shafts are parallel. Figure 4.1 shows the design of the surgical instruments.

The 6-DoF instruments are fabricated with stainless steel using an additive manufacturing process (Selective Laser Melting - SLM). For a detailed analysis on SLM manufacturing of surgical instruments, please refer to [161]. Each instrument has an overall diameter of 5.5 mm, complete shaft length of 250 mm and flexible tip length of 50 mm, enabling it to explore tight operating sites effectively. The transmission of the actuation is based on tendons that are affixed to capstan mechanisms through a pulley system. The capstan mechanisms are coupled with the motors in the proximal actuation system. The latter is enclosed in a custom designed casing that is rigidly attached to the robotic arm. The casing is fabricated

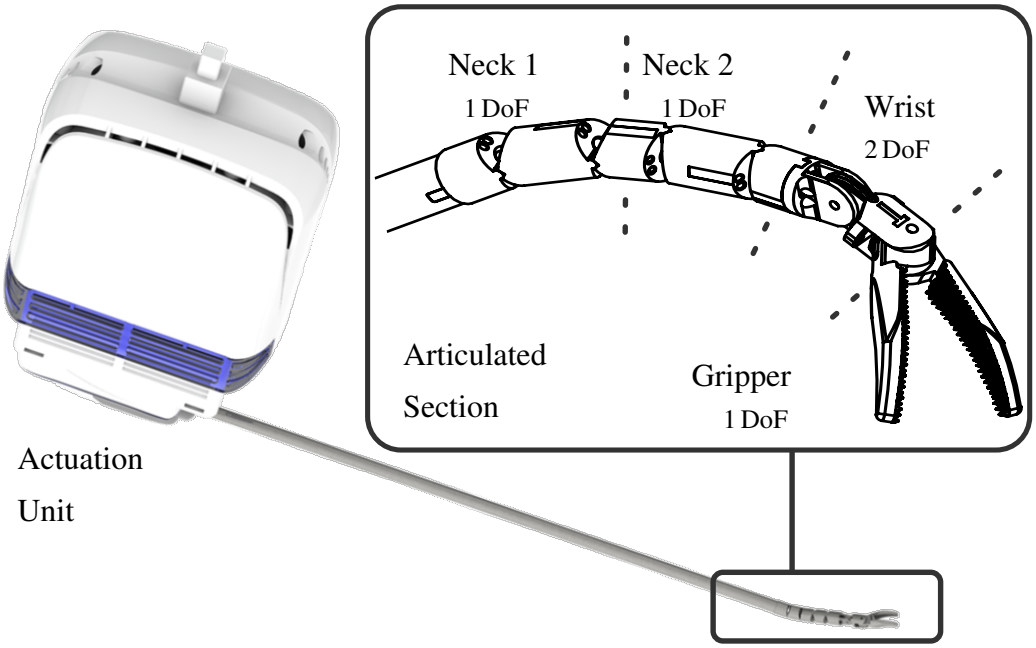


Figure 4.2: Actuation unit and articulated instrument section.

with sterilisable, thermoplastic ABS material using a rapid prototyping process (Fused Deposition Modelling - FDM). Figure 4.2 shows the surgical instrument and actuation units. These actuation units house the hardware components required for the motion of the tools. This comprises Maxon EC brushless DC motors (Maxon Motor, Switzerland), custom designed motor controller modules and an interface motherboard.

4.2.2 Highly Redundant Master-Slave System

The proposed system uses three KUKA LBR iiwa 7-DoF redundant robotic arms (KUKA Roboter GmbH, Germany) for the global positioning of two flexible instruments and a stereoscope. The user interacts with the system via two Omega.7 haptic devices (Force Dimension, Switzerland), and a Tobii Pro X2-60 eye tracker (Tobii AB, Sweden). The system setup is shown in Figure 4.3.

The 6-DoF flexible instruments are attached onto the robotic arms via custom-designed actuation units. The power line and communication bus for each of the flexible instruments are routed internally through the corresponding robotic arm

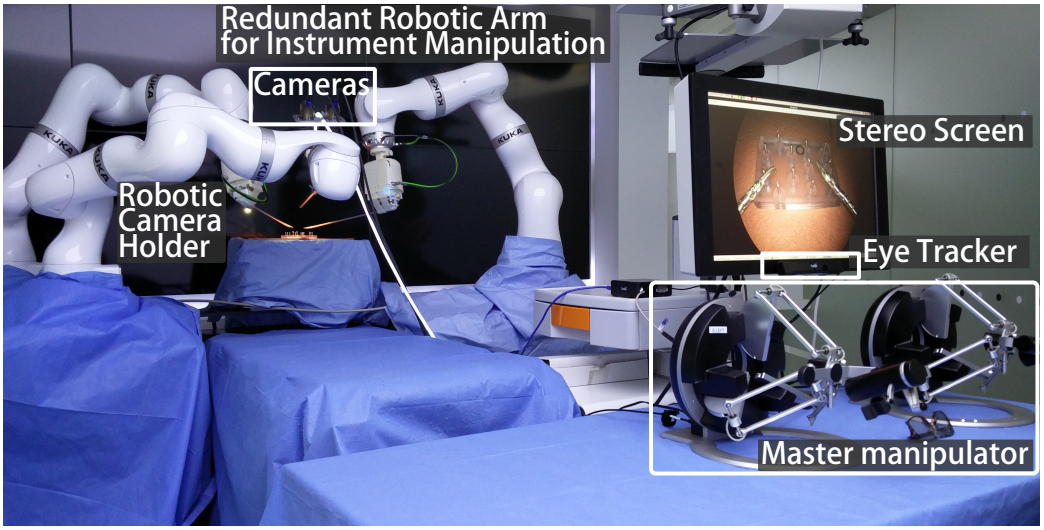


Figure 4.3: Overview of the proposed adaptive motion scaling system.

and connected via a short cable from the arm's flange, to prevent cable tangling during large movements. This is visible in a close-up of the system shown in Figure 4.4.

The three KUKA iiwa robotic arms communicate with a workstation via User Datagram Protocol (UDP) messages, at a frequency of 250 Hz each. The workstation communicates with the surgical instruments via a separate channel routed through the robotic arms for each instrument. For a given instrument, the workstation issues messages through USB2 to a transceiver. The transceiver communicates to the interface motherboard via RS-485, which interfaces directly with each of the Maxon EC brushless DC motors. A safety watchdog housed within the actuation unit monitors the communication channels and motor status to prevent undefined behaviour in case of a fault. Each actuation unit RS-495 communication is limited to 1000 packets (i.e. message sent and answer received) per second. As each unit houses a total of seven motors, the effective communication rate per motor is 143 Hz. A schematic summarizing the connection between the different components of the system is shown in Figure 4.5.

The robotic arms are calibrated relative to each other using the method presented in section 3.3.1. The arms are calibrated two at a time, for a total of two calibration procedures. The stereoscope is mounted similarly as in chapter 3, and the same hand-eye and camera calibration methods are used as those presented

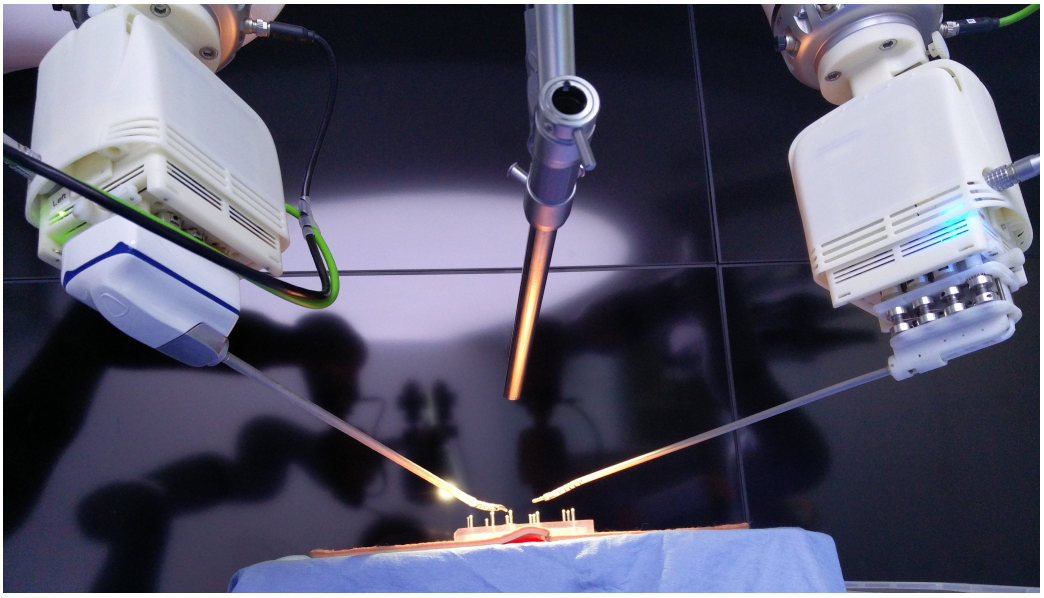


Figure 4.4: Micro-IGES instruments mounted on the robotic arms.

in section 3.3.2. Due to the number of robotic components involved, the manner in which the frames of reference of the system are set up have been modified. Figure 4.6 illustrates the transformation chains of the different components. All robot base frames B_1, B_2, and B_3, are expressed relative to the world frame. For simplicity, the world frame is often chosen as one of the robot base frames. The virtual tool frame T_v and the corresponding transformations to the instrument and camera end-effector X_{v1}, X_{v2}, and X_{v3} are useful to define geometrical constraints between the different system components.

4.2.3 Instrument Kinematics

Both 9-DoF tools are controlled using a combined kinematics scheme between the redundant robotic arms and the flexible instruments. This is achieved by considering a virtual 9-DoF manipulator, see Figure 4.7. One DoF ($n_g = 1$) represents the opening angle of the gripper, and the remaining 8 DoF ($n_k = 8$) are used for positioning and orienting the tool end-effector in the surgical task space. The base of the virtual tool is placed at the surgical port, where the tool enters the workspace, and where a virtual ball joint provides pivoting motion. The axial rotation of the virtual tool as well as the five distal DoFs are actuated by the 6-DoF instrument

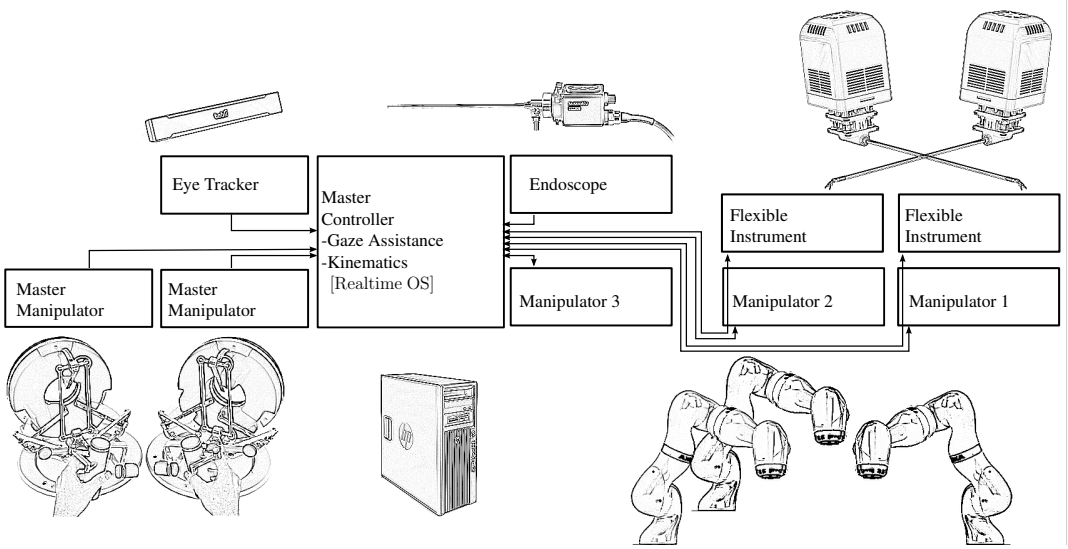


Figure 4.5: Functional overview of the complete system. Communication and sensor data paths are marked with arrows.

and are directly mapped to the joint-space of the flexible instrument. The remaining 3 DoF of the virtual tool are actuated by the redundant robotic arms, where the pitch and yaw rotation angles of the ball joint and the translation of the prismatic link are used to construct the corresponding end-effector poses of the robotic arms.

The manufacturer-provided inverse kinematic solver is used to control the robotic arms. The inverse kinematic problem of the virtual tool is solved using the Jacobian ($\boldsymbol{J} \in \mathbb{R}^{6\times n_k}$, $n_k = 8$) based Newton-Raphson method. The developed inverse kinematic scheme is designed to perform well in conventional laparoscopic procedures, and to also be suitable for surgical interventions permitting only very limited triangulation at the port. This last configuration results in close-to-parallel shafts, significantly limiting the operation space of the ball joint in its pitch and yaw motion, as illustrated in Figure 4.8. In such configurations the positioning of the end-effector has to be provided mainly by the flexible instruments. Due to the relatively short length of the articulated instrument tips, manipulation close to the joint limits is likely in such situations.

Based on these considerations as well as surgeon feedback, the task space ($\boldsymbol{x} = [t_x, t_y, t_z, r_x, r_y, r_z]^T$) is separated into primary ($\boldsymbol{x}_p = [t_x, t_y, t_z, r_z]^T$) and secondary ($\boldsymbol{x}_s = [r_x, r_y]^T$) task-spaces. The primary task-space represents the position and instrument roll whereas the secondary task-space represents the re-

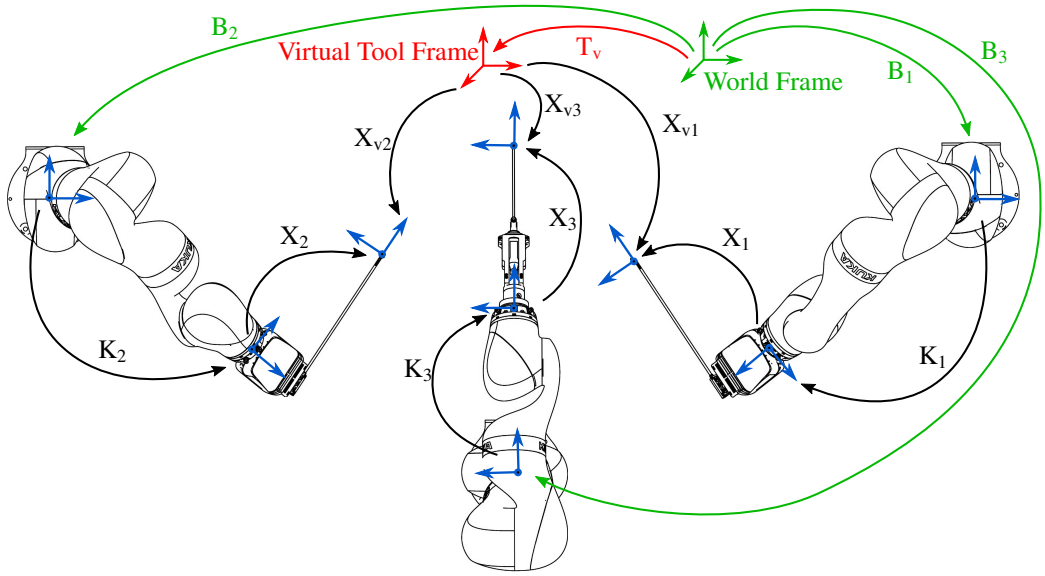


Figure 4.6: Schematic representing the relationships between the relevant reference frames of the system.

maining orientational DoFs (pitch, yaw). Consequently, the primary Jacobian $\boldsymbol{J}_p \in \mathbb{R}^{4\times n_k}$ consists of the first three rows and the last row of $\boldsymbol{J}$; and $\boldsymbol{J}_s \in \mathbb{R}^{2\times n_k}$ consist of the fourth and fifth row of $\boldsymbol{J}$. The secondary task goals are optimized using the *null*-space of the primary goals, for which the pseudo-inverse Jacobian $\boldsymbol{J}^\star$ is calculated based on the damped-least-square (DLS) method by

$$\boldsymbol{J}^\star = \boldsymbol{V}\,\mathrm{diag}\left(\frac{1}{\tilde{\sigma}_1},\ldots,\frac{1}{\tilde{\sigma}_n}\right)\boldsymbol{U}^T, \tag{4.1}$$

where $\boldsymbol{U}$, $\boldsymbol{V}$ are right and left singular vectors provided by a singular value decomposition (SVD) of $\boldsymbol{J}$. $\tilde{\sigma}_i^2$ is the modified i^{th} eigenvalue potentially damped by,

$$\tilde{\sigma}_i = \begin{cases} \sigma_i & ,\text{if } \min_j\left(|\sigma_j|\right) \geq \sigma_{\min} \\ \left({\sigma_i}^2 + {\lambda_t}^2\right){\sigma_i}^{-1} & ,\text{else} \end{cases}, \tag{4.2}$$

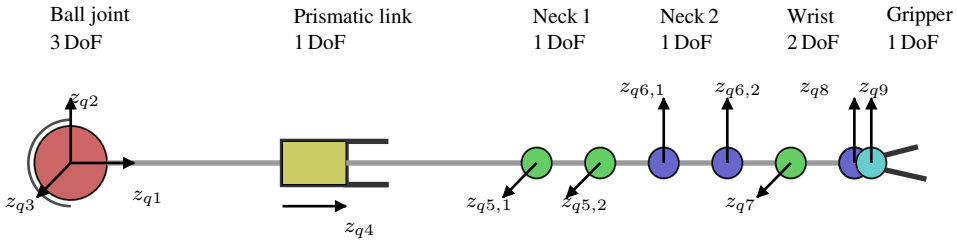


Figure 4.7: Illustration of the kinematic chain of the virtual 9 DoF robotic manipulator. Featuring dependent joints in neck 1 and neck 2, and a wrist with 2 DoF, where the pitch is actuated by q_7, the yaw is the mean of q_8 and q_9, and the gripper DoF is represented by the angle difference of q_8 and q_9.

and with

$$\lambda_t{}^2 = \lambda_{\max}{}^2 \left[1 - \left(\min_j(|\sigma_j|)\sigma_{\min}{}^{-1}\right)^2\right], \tag{4.3}$$

where $i, j \in \{1, \dots n\}$, and $\lambda_{\max}$ represents the maximum damping allowed. This provides stable manipulation close to singular configurations, by avoiding excessive joint-velocities. However, this comes at the cost of slower convergence rates in the optimization process, and a suboptimal *null*-space mapping, therefore the damping in equation 4.2 is only active in case the smallest singular value is below a critical singular value threshold. Furthermore, in equation 4.3 the damping is increased the smaller the minimal singular value is. Let $\boldsymbol{q}_t$ be the joint values at time step t, $\Delta\boldsymbol{q}_{t+1}$ be the joint increment vector from time t to $t+1$, and $\boldsymbol{q}_{null}$ the null-space joint values. The damped inverse Jacobian for the primary ($\boldsymbol{J}_p \in \mathbb{R}^{n_k \times 4}$) and secondary ($\boldsymbol{J}_s \in \mathbb{R}^{n_k \times 2}$) task-space are calculated using equation 4.1, and to reach the desired end-effector pose $\boldsymbol{T}_{e,d}$ the joint-values of the virtual tool are iteratively updated with

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \Delta\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \dot{\boldsymbol{q}}_t \Delta t, \tag{4.4}$$

$$\dot{\boldsymbol{q}}_t = \boldsymbol{J}_p^\star \dot{\boldsymbol{x}}_p + \left(\boldsymbol{I}_{n_k} - \boldsymbol{J}_p^\star \boldsymbol{J}_p\right) \dot{\boldsymbol{q}}_{null}, \tag{4.5}$$

$$\dot{\boldsymbol{q}}_{null} = \boldsymbol{f}_{\text{ori}}\left(\dot{\boldsymbol{x}}_p, \dot{\boldsymbol{x}}_s, \boldsymbol{J}_s^\star\right) + \boldsymbol{f}_{\text{qlim}}\left(\boldsymbol{q}_t\right), \tag{4.6}$$

where the *null*-space of the primary optimization goal is used in equation 4.5 to optimize for secondary goals in equation 4.6. The secondary goals are i) the

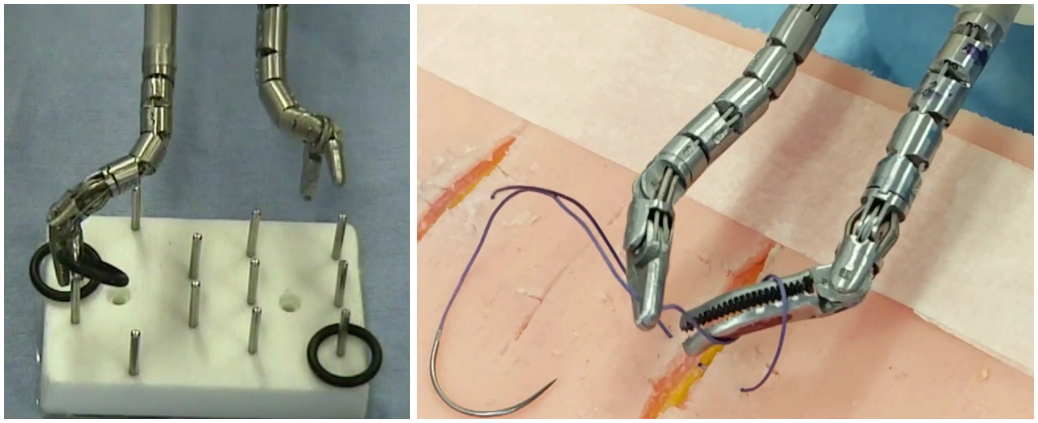


Figure 4.8: Robotic instruments in parallel configuration as used in single port access surgery.

yaw, pitch-orientation of the end-effector pose ($\boldsymbol{f}_{\text{ori}}$), and ii) joint-limit avoidance ($\boldsymbol{f}_{\text{qlim}}$). They are defined as,

$$\boldsymbol{f}_{\text{ori}} = k_{\text{ori}}\Big(1 - \frac{\min\left(\|\dot{\boldsymbol{x}}_p\|,\ \dot{x}_{\text{p,max}}\right)}{\dot{x}_{\text{p,max}}}\Big)\boldsymbol{J}_{\text{s}}^{\star}\dot{\boldsymbol{x}}_{\text{s}}, \tag{4.7}$$

$$f_{\text{qlim},i} = k_{\text{qlim}}\left|\frac{q_{i,\text{max}} + q_{i,\text{min}} - 2\,q_i}{q_{i,\text{max}} - q_{i,\text{min}}}\right|^{k_e} \cdot \operatorname{sgn}(q_i - \frac{q_{i,\text{max}} + q_{i,\text{min}}}{2}). \tag{4.8}$$

The weight in equation 4.7 is dependent on the primary task-space velocity $\dot{\boldsymbol{x}}_p$ to ensure that primary goals are met first even when the *null*-space mapping is suboptimal. Furthermore, the exponent $k_e > 1$ (here $k_e : 3$) in equation 4.8 emphasises join-limit avoidance only for configurations close to the limits which allows to optimize mainly for the orientation goals. However, joint limit avoidance cannot guarantee that joint limits are completely avoided, in particular if the desired pose is in locations of low dexterity or outside of the reachable workspace.

4.2.4 Dexterity Analysis for Task Space Restriction

While the previous section addressed the challenge of reaching joint limits in joint-space, a complementary method based on workspace analysis [162] is used to restrict the end-effector set point $\boldsymbol{T}_{e,d}$ in task space, as illustrated in Figure 4.9.

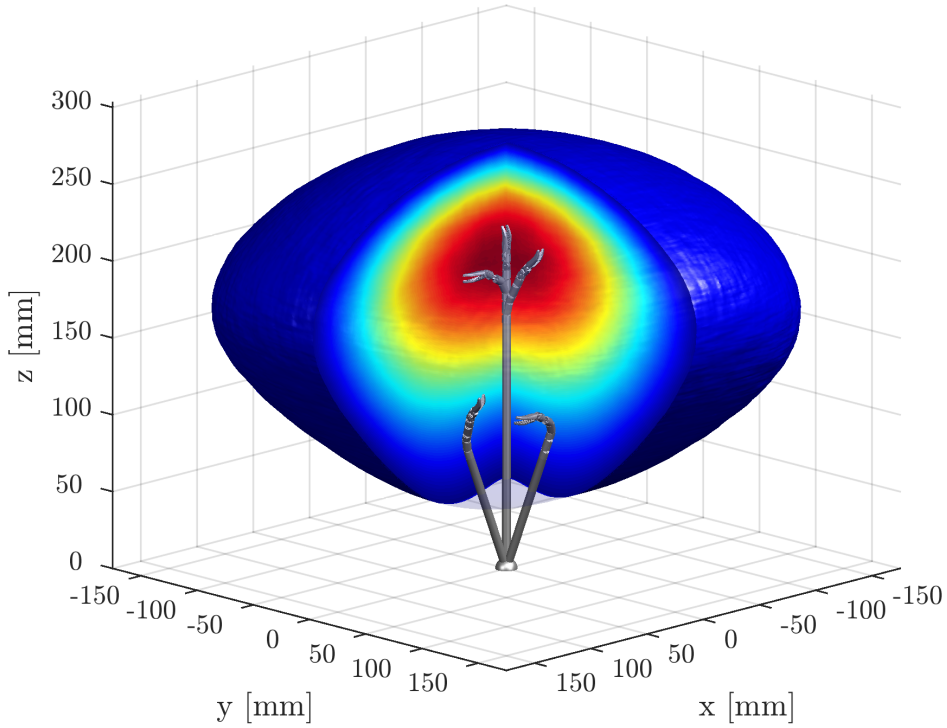


Figure 4.9: Dexterity $\mathcal{D}$ analysis of the virtual instrument, using 1×10^9 random robot configurations. Red: high dexterity, blue: low dexterity.

The dexterity measure $\mathcal{D}$ is calculated as,

$$\mathcal{D} = \mathcal{L}_q \, \mathcal{M} = \mathcal{L}_q \sqrt{|\boldsymbol{J} \, \boldsymbol{J}^T|}, \tag{4.9}$$

where the joint limit measure $\mathcal{L}_q$ are augmented onto the manipulability measure $\mathcal{M}$, and where $\mathcal{L}_q$ is defined as

$$\mathcal{L}_q = 1 - \exp\left\{ -\kappa \prod_{i=1}^{n_k} \frac{(q_i - q_{i,\min})(q_{i,\max} - q_i)}{(q_{i,\max} - q_{i,\min})^2} \right\}. \tag{4.10}$$

Evaluating the dexterity of random tool configurations based on the end-effector position provides a map which depicts regions of low (blue) and high (red) dexterity, as shown in Figure 4.9. Voxel entries in this map are generated by recording the maximum dexterity value obtained across multiple tool configurations for

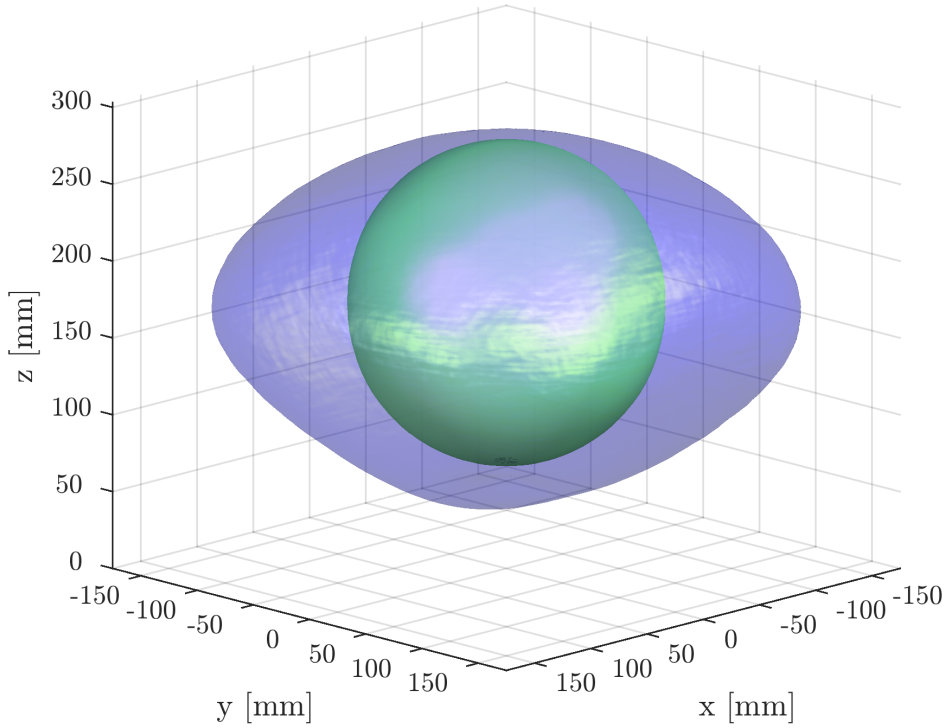


Figure 4.10: Manipulation restriction based on dexterity measure, green: permitted workspace, blue: reachable workspace.

a given end-effector position. This further allows the definition of a dexterity-dependent threshold boundary, limiting the permitted manipulation workspace to areas of sufficient dexterity, see Figure 4.10. Using simple geometric primitives such as spheres, capsules or convex hulls, the restricted workspace is enforced by projecting the user defined poses onto the permitted workspace to obtain the desired end-effector pose $\boldsymbol{T}_{e,d}$. To avoid slack in the manipulation, $\boldsymbol{T}_{e,d}$ follows the set-point if it is moved back towards the inside of the restricted workspace.

4.3 Gaze-Assisted Adaptive Motion Scaling

4.3.1 Limitations of Master Devices for Highly Dexterous Slave Systems

Surgical robots, such as the one presented in this chapter, typically operate under a master-slave configuration. The surgeon's motions on the master device are used to directly control the slave robot, allowing the surgeon to stay in complete control of the system while benefiting from robotic advantages. As noted in section 2.2, a consequence of this paradigm is that the performance of the whole system is highly dependant on the master device, to provide a well-suited interface to the slave robot. A highly dexterous slave system will ultimately not perform optimally if the user is not capable of harnessing this dexterity using the interface at their disposal. A well-known limitation arising from this is tied to motion scaling. Motion scaling has always been considered a clear advantage of robotic systems, allowing the surgeon to perform much finer movements than would be possible by hand. However, the same scaling factor can make moving the instruments around the surgical scene time consuming and tedious by forcing the user to clutch and reposition the master device end-effectors. This behaviour is particularly noticeable on master devices with small workspaces, as users will often reach the edge of the master device's workspace while performing larger movements.

In effect, this problem arises because the optimal motion scaling factor changes depending on what the user is trying to achieve. For delicate tasks requiring precise movements, a small scaling factor is desired. To cross the surgical workspace during reaching motions, a larger scaling factor would be desired. For the sake of clarity, in this thesis an increase in the scaling factor generates faster robot motions, and a decrease in the scaling factor generates slower robot motions. While it is possible to change the value of the scaling factor manually on some systems, this does not constitute a practical solution as the surgeon is effectively substituting clutching for manually toggling between different scaling factors. Ideally, the system should modify the system behaviour depending on what the user is trying to accomplish. This work proposes to use eye-tracking data in conjunction with multiple other sources of information available from the robotic system to infer where the surgeon is trying to reach. This information is used to modulate the

motion scaling factor such that distant targets are reached more quickly.

4.3.2 Definition of Aims and Behaviour Analysis

The proposed approach seeks to fulfil the following aims:

1. Safety: to be safe, such that surgeons never encounter undesirable motions or unpredictable system behaviour,

2. Efficiency: to noticeably improve the speed at which distant goals can be reached by reducing the need for clutching,

3. Fidelity: to fulfil the above goals without sacrificing the capability to perform fine movements by taking full advantage of motion scaling,

4. Intuitiveness: to fulfil the above goals using an implicit approach, i.e. without introducing additional inputs for the user to handle.

The choice of motion scaling modulation displays some clear advantages to achieve these goals. In particular, adaptive motion scaling presents an inherent safety as any motion performed by the robot has to be directly generated by the user. However, in order to modulate the motion scaling in an effective manner the user's intention must be inferred by the system. The core premise behind the proposed approach stems from the observation that, when reaching for an object, humans fixate the object they are trying to reach and not their hands. Similar behaviour can be observed in a variety of disciplines, from sports to instruments: once an individual is proficient enough in a task, they tend to fixate their destination and not their tool. The same principle is exploited here to determine where the user is trying to move the surgical instruments. In order to elaborate on this concept, several different types of information are needed. Firstly, the system must be able identify where the user is looking with regards to the instruments. This necessitates the use of an eye-tracker, but also to track the instruments in the camera image. To further assess the user's intention, the hand motions performed using the master devices are also recorded. By combining gaze and tool tracking information with the direction in which the user is moving the master device end-effectors as well as its speed, the system is capable of further improving its evaluation of the user's

intention. Lastly, the homogeneous transformations of the robotic arms are necessary to bring these different sources of information into the same reference frame. In summary the following input data is required:

- Eye-tracking data,
- Tool tracking information,
- Hand motions (from the master device),
- 3D pose of the robotic instruments.

4.3.3 Instrument Tracking

A detection-and-tracking approach was implemented to find the 2D locations of the robotic tools. This approach consists of a colour-based detector that localises the end-effector of the tool (defined here as the wrist joint), which was coloured with a green dot. As the instrument shape can change significantly depending on its kinematic configuration and cause the detector to fail, a tracker was included to consider the temporal information of the tool motion. The salient image gradient features [163] were extracted inside a bounding box that encloses the end-effector. As can be seen from Figure 4.11, these gradient features identify well the tool contours. We then perform forward-backward tracking based on optical flow [164] to estimate the locations of these gradient features in subsequent images. To determine the location of the tool in the current image, we consider the 2D offsets of

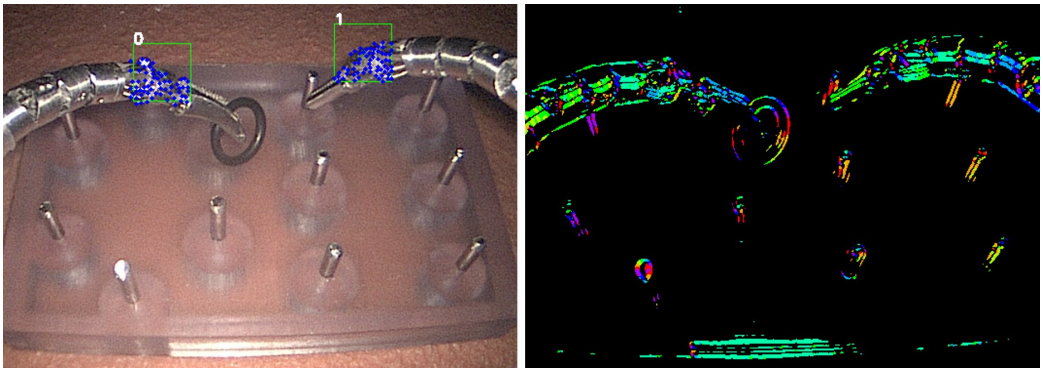


Figure 4.11: Tool tracking. Left: temporal feature tracking and bounding boxes. Right: salient image gradients.

the features and calculate the location and size of the bounding box. The location of the tool from the tracker is then defined as the centre of the box in the current image. Together the detector and temporal tracker provided a robust tool tracking method running at over 30 frames per second.

4.3.4 Adaptive Scaling Generation

The aim of the adaptive motion scaling is to generate a dynamic value $k_{as} \in [0, 1]$ such that the final scaling factor of the system k_{st} is defined as:

$$k_{st} = s_b \times (1 + s_c \times k_{as}), \tag{4.11}$$

where s_b is base scaling factor of the system (a typical value for robotic systems being 0.25 for a ratio of 1 to 4) and s_c is the scaling coefficient multiplier attributed to adaptive scaling. Using the input data detailed in section 4.3.2, three main contributing factors to k_{as} can be listed:

1. The distance between the gaze fixation point and the instrument detected in the image. Higher distances should result in a higher scaling value, while the scaling should not be increased when the user is looking close to the tool as this does not indicate a desire to reach a distant location.

2. The angle between the vector generated from the instrument tip to the fixation point and the hand motion. Lower angle values should result in a higher scaling value as the user would be moving the robot in the direction where he is looking. High angle values should not increase the scaling value.

3. The speed at which the user is moving the master device end-effectors. Low speeds could indicate that the user is not trying to reach a distant objective, and as such the scaling factor should not be increased.

These three factors are respectively written as k_d, k_a, and k_s, and can all range from 0 to 1. The contribution of the adaptive scaling can then be defined as:

$$k_{as} = k_d \times k_a \times k_s. \tag{4.12}$$

Figure 4.12 displays the flow of information used to generate the each scaling

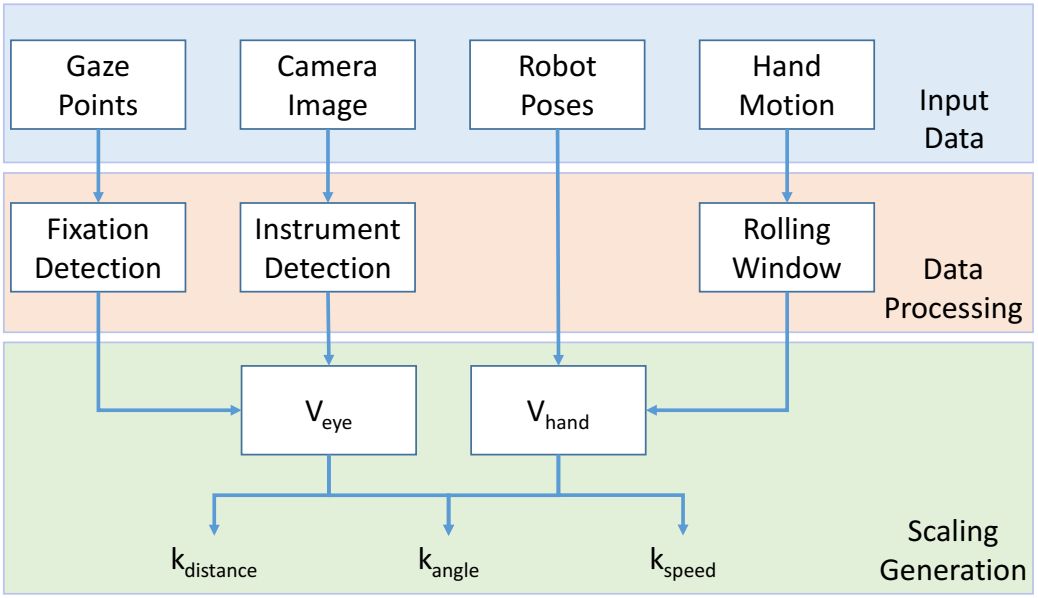


Figure 4.12: Input data flow for the generation of the scaling factor.

factor. In order to define these three factors, the two following vectors must be defined:

$$V_{\text{eye}} = P_{\text{fixation}} - P_{\text{tool}}, \tag{4.13}$$

$$V_{\text{hand}} = P_{\text{hand}_t} - P_{\text{hand}_{t-t_w}}, \tag{4.14}$$

where P_f is the user's 2D fixation point on the screen, P_{tool} is the detected 2D position of the instrument on the screen, P_{hand_t} is the position of the hand at the present time t as obtained from the master device, and $P_{\text{hand}_{t-t_w}}$ is the position of the hand at a previous time based on a specific time window duration t_w. To obtain the fixation points from the eye tracking data, a Dispersion-Threshold Identification (ID-T) approach was used [80, 81]. Note that V_{hand} is a 2-dimensional vector obtained from the projection of the hand motion at specific time intervals t_w onto the camera plane. Using these definitions, each of the three factors comprising k_{as}

s_c	d_{pmax} (pixels)	a_{max} (deg)	sp_e (mm/s)	t_w (s)	a_{min} (deg)
1.3	221	40	20	0.1	20

Table 4.1: Adaptive scaling parameters. The pixel values for d_{pmax} are for a 1920x1080 monitor resolution. The parameters in red are used to tune the system.

can be defined:

$$k_d = \frac{\min(||V_{eye}||, d_{\mathrm{pmax}})}{d_{\mathrm{pmax}}}, \tag{4.15}$$

$$k_a = 1 - \frac{\min(\max(0, a_c - a_{\mathrm{min}}), a_{\mathrm{max}} - a_{\mathrm{min}})}{a_{\mathrm{max}} - a_{\mathrm{min}}}, \tag{4.16}$$

$$k_s = \frac{\min(||V_{\mathrm{hand}}||/t_w, sp_e)}{sp_e}, \tag{4.17}$$

where d_{pmax} is the distance in pixels under which k_d starts to decrease linearly, a_c is the angle between V_{eye} and V_{hand}, a_{min} is the angle below which k_a is maximum (value of 1), a_{max} the angle above which k_a is minimum (value of 0), and sp_e is the speed under which k_s starts to decrease linearly. The different parameters presented were fine-tuned over multiple iterations of the system to provide the responsiveness desired. Table 4.1 displays the values used for the user study. Note that t_w and a_{min} were fixed during the tuning of the parameters, allowing the linear model for each coefficient of the adaptive motion scaling to be adjusted using only one parameter. These were chosen because their values were more easily set intuitively: the accuracy of the eye tracker and normal eye behaviour makes a value of a_{min} less than $20°$ meaningless, and t_w is chosen as small as possible to generate a reasonable velocity measure.

To avoid sharp changes in the scaling factor due to single steps in the algorithm, a moving window spanning 100 milliseconds (same value as t_w and the fixation detection window from the ID-T algorithm) was introduced to smooth out the variations of k_{as}. As eye tracking and tool detection can both fail for various reasons (e.g. tools missing from the image, user looking away), an additional layer of safety was introduced in the implementation of the adaptive scaling algorithm. In case either the instrument or the user's gaze fixation point is not found, both k_d and k_a are considered invalid and set to zero.

4.3.5 Algorithm Behaviour Analysis

Figure 4.13 shows the behaviour of the adaptive motion scaling during a typical reaching motion, for the left instrument in this case. In the first part of the figure (top left image, graphs highlighted in light red) the user has just started looking at the new destination for the left instrument and the motion has not yet started. As

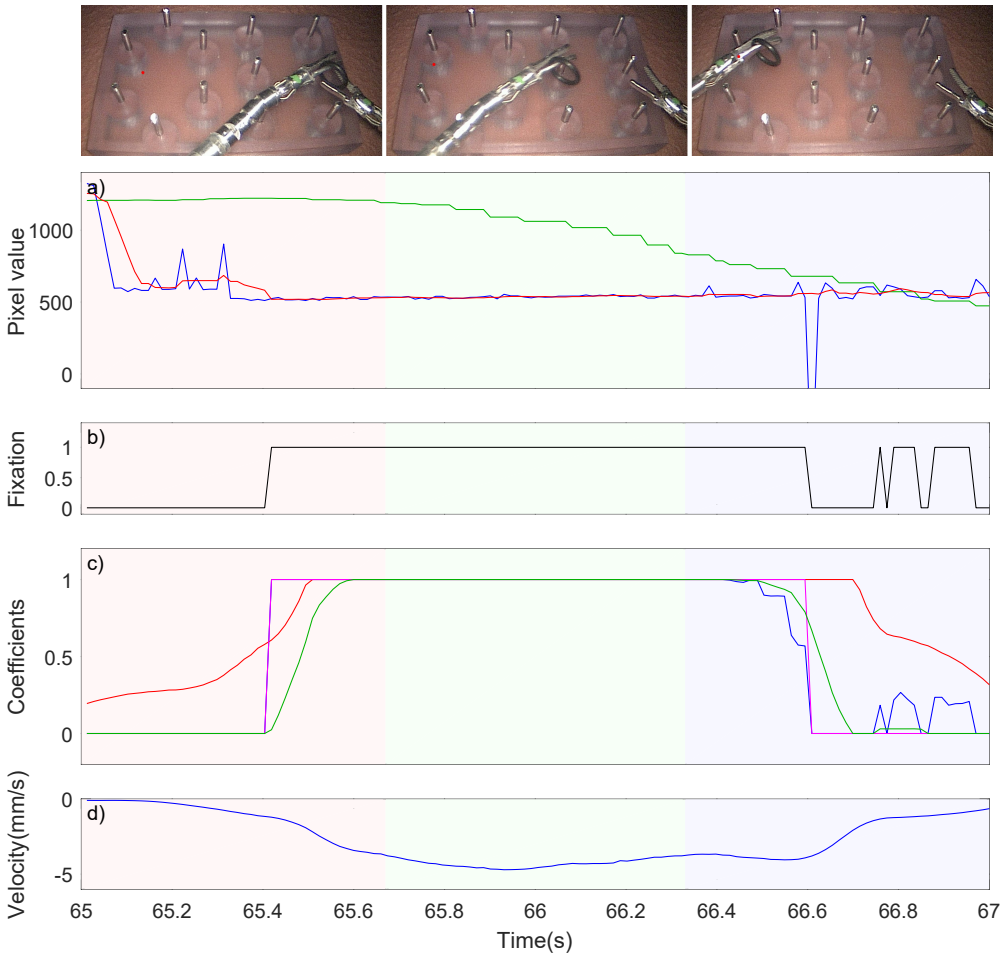


Figure 4.13: Evolution of scaling coefficients from input data. Top: from left to right, images of a ring transfer from the right instrument to the left instrument in chronological order. The red dot represents the gaze point. a) Raw X-axis gaze pixel values (blue), filtered X-axis pixel values (red), and left instrument end-effector X-axis pixel values (green). b) validity of fixation data (fixation identified: value of 1, otherwise 0), c) k_d (blue), k_a (magenta), k_s (red), and k_{as} (green). d) V_{hand} X-axis value in mm/s (blue).

such the ID-T fixation detection algorithm has not had time to register a fixation and so both k_d and k_a are set to a value of 0. By the time the fixation is registered the motion of the master device in the correct direction has already begun, and so the scaling factor k_{as} increases as all conditions are met. The smooth increase of k_{as} is due to the moving filter used to prevent sharp variations of the scaling factor. In the second phase the scaling factor reaches its maximum value as the motion carries on in the right direction. Finally, upon reaching the desired location k_d drops significantly, followed by the other coefficients as the user is no longer fixating on the same point. This results in a smooth decrease of k_{as} starting before the target has been reached, and finishing under 100ms after the target has been reached.

4.4 User Studies

4.4.1 Experimental Setup

A user study was conducted to evaluate the usability and effectiveness of the adaptive motion scaling. As presented in section 4.2, the robot is capable of a large dexterous workspace due to its numerous degrees of freedom. To replicate the clutching issues encountered by this type of robotic system, a task was designed to require both a large workspace and a requirement for precise motions. A ring-passing task was chosen, requiring large motions to move the ring across the workspace, while also requiring precise motions two pass the ring from one instrument to the other. Users were asked to pass a rubber ring back and forth between the two instruments at specific locations marked using a large peg board. Figure 4.14 illustrates the ring-passing task. The instruments would start the task at standardized locations at the bottom of the screen, marked (1) for the right instrument, and (2) for the left instrument, as noted in Figure 4.14. The rubber ring was held by the right instrument at the beginning of the task. The participant would begin by bringing the left instrument to grab the ring from the right instrument at position (1), and then bring the left instrument back to position (2). The right instrument was then brought to position (2) to grab the ring, and then to position (3). This process would carry on, passing the ring between both instruments while moving upwards in rows until the right instrument arrived at position (5).

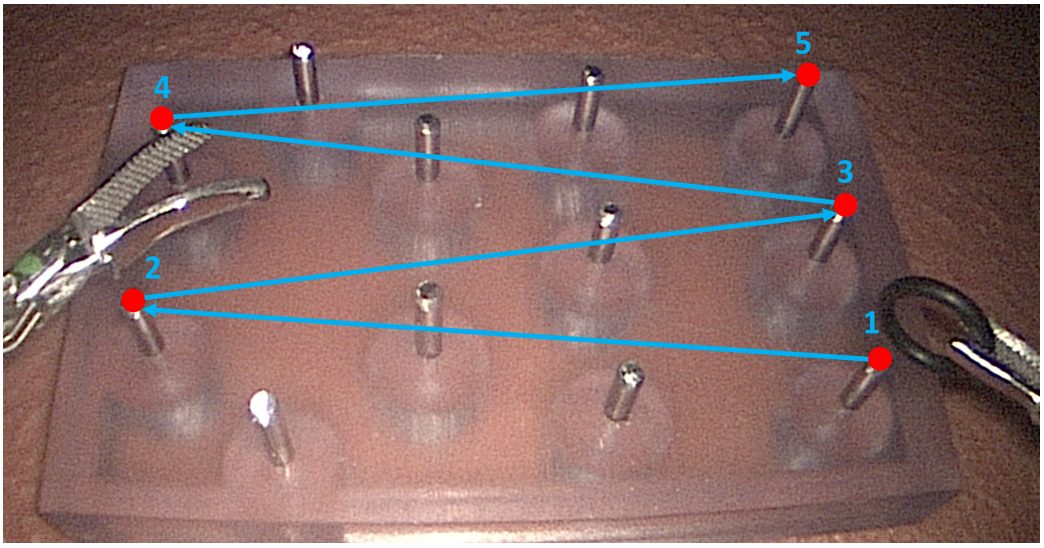


Figure 4.14: Illustration of the different transfer points for the adaptive motion scaling user study.

The left instrument was then brought to grasp the ring one last time at position (5), and the task ended. As such, a total of nine reaching motions had to be performed.

Nine non-clinical participants were presented to the system and given time to familiarize themselves with the controls, six male, 3 female. Participants were first trained to use the robotic system, with no time limit. This training consisted in performing basic motions with the system to familiarise them with the master-slave controls, and stopped once participants were sufficiently comfortable controlling the instruments. Participants were then trained on the task twice, once with adaptive motion scaling and once without. After training, each participant would conduct the experiment once with adaptive motion scaling and once without, in a randomized manner. The key metrics for the study were the task completion time and the number of times the system had to be clutched. The static motion scaling factor chosen was 0.25, a standard value used for master-slave systems in robotic laparoscopic surgery.

4.4.2 Results

The results of the study are displayed in Figure 4.15. Numerical values for the mean, median, and statistical significance of the key metrics are listed in Table 4.2.

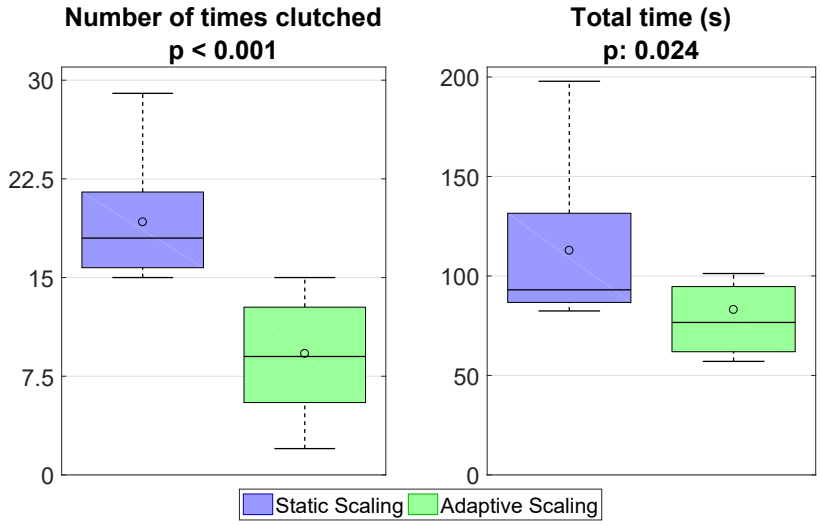


Figure 4.15: User study comparing task execution times and number of times the system was clutched across nine participants.

As can be seen, the use of the adaptive scaling provides a highly significant improvement in both task execution times and in the number of times the system had to be clutched. A median reduction of 50% in number of times the system was clutched was observed using the adaptive scaling relative to the static scaling. Additionally, a median reduction of 17.6% in the task completion time was

	Mean		**Median**		**p-value**
	Static	Adaptive	Static	Adaptive	
Number of times the system was clutched	19.22	9.22	18.00	9.00	2.47×10^{-4}
Total task time (s)	112.8	83.05	93.03	76.64	0.0244

Table 4.2: Mean, median, and statistical significance results from the ring transfer study.

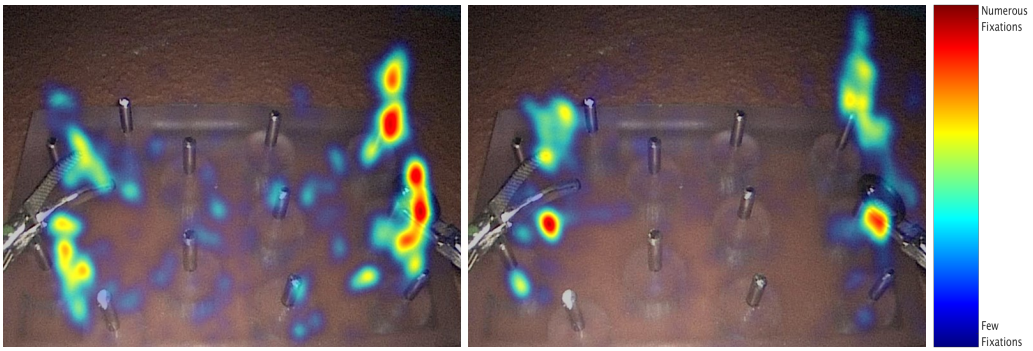


Figure 4.16: Heatmaps of gaze fixations during a ring transfer task. Left: without adaptive scaling. Right: with adaptive scaling.

observed using the adaptive scaling relative to the static scaling. User feedback was strongly in favour of the system, which can be explained by the reduction in frustration at having to constantly clutch and reposition the master manipulator.

Figure 4.16 displays heatmaps of the eye-tracking data obtained from the ring transfer task, with and without the adaptive motion scaling.

4.5 Discussion and Conclusion

This chapter presents implicit, gaze-assisted adaptive motion scaling for highly articulated instrument manipulation. It formalizes the differential inverse kinematic approach used to control a redundant 9-DoF virtual tool, suitable to operate in tight spaces with little or no triangulation and close to joint-limits, while also providing highly dexterous manipulation in conventional laparoscopic procedures. Furthermore, a novel gaze-assisted control scheme was presented using implicit operator input to seamlessly scale-up master-device motions and allow more effective manipulation, without requiring any active input from the user. The developed system infers the user's intention and facilitates time-efficient motions, as well as reduces the need for clutching in an inherently safe manner. The fact that the presented approach only affects the motion scaling factor means that its implicit nature does not compromise the core safety of the system, as the surgeon always remains in complete control of the surgical instruments. A user study is conducted to evaluate the performance of the adaptive motion scaling.

The results from the user study show that the adaptive motion scaling is able

to reduce for the need for clutch in half compared to using a static scaling factor. This improvement not only serves to reduce user frustration due to interruptions from clutching, but also translates in a significant 17% reduction in median task completion time. Both of these results highlight the clear usefulness of the presented approach. The heatmaps generated from the eye-tracking data shown in Figure 4.16 further illustrate this point. The lack of adaptive scaling is clearly noticeable by the user having to constantly look back at the instruments to clutch the master devices, resulting in noticeable scattering of the heatmap across the centre of the screen.

The strong performance of the presented approach compared to the static motion scaling is likely emphasized by the fact that the Omega.7 master devices used have a small workspace. As noted previously in section 2.2, other haptic master interfaces exist capable of providing larger workspace, with different trade-offs. However, any use of a non-velocity based haptic master device will inevitably run into clutching issues at some point if the procedure performed requires both a large workspace and precise motions, preventing the comfortable use of a larger static scaling factor. This is compounded by the fact that, while a larger master device workspace may reduce the need for clutching, users will typically clutch to re-centre the master manipulators rather than work with their arms far apart. This behaviour can be observed even in existing commercial systems, making the approach presented in this chapter particularly valuable.

A key limitation of the work presented in this chapter is the lack of evaluation of the natural gaze behaviour on which the adaptive motion scaling relies. While the base assumption that the user will fixate their destination during reaching motions will typically be valid to some extent in the majority of cases, the exact extent to which that assumption will stay valid could vary substantially depending on the context. It could be that some users fixate only briefly on their destination while others linger on it during the whole motion. In this study, some participants took full advantage of the adaptive motion scaling by making sure to linger their gaze on the destination. However, this behaviour may not be representative of typical surgical conditions. In this case the system parameters would need to be updated accordingly, ideally adapting to the user's preferences.

In this chapter, the key parameters of the adaptive motion scaling were optimized manually, over several iterations of the system. While this approach is

commonly used in practice to tune robotic systems and human-robot interfaces, it does not provide any guarantees towards the performance of those systems. When those parameters influence directly the behaviour of the robotic system, such as in this case, it may be desirable to have a more principled approach to tune them. Additionally, different users may have different preferences for the system behaviour, making it even more enticing to develop a user-specific approach for the optimization of these parameters. The next chapter tackles this problem, exploring probabilistic approaches to optimize human-robot interfaces, in particular the gaze-assisted adaptive motion scaling.

5

User-Specific Adaptation of Motion Behaviour

5.1 Introduction

In the previous chapter, a novel gaze-assisted adaptive motion scaling was presented, allowing users to perform reaching motions significantly more quickly and with less need for clutching. The adaptive motion scaling presented in chapter 4 consists in assessing the user's intention to reach a distant objective with an instrument, and dynamically adjusting the scaling factor to facilitate that aim. To that effect, four main components are computed:

1. The speed at which the user's hands are moving k_s.
2. The angle between the hand motion direction and the instrument-to-gaze fixation point direction k_a.
3. The distance between the instrument and the gaze fixation point k_d.
4. The maximum scaling allowed s_c.

Content from this chapter was published and is reproduced with permission from,
Gaze-Assisted Adaptive Motion Scaling optimisation using Graded and Preference Based Bayesian Approaches. Gras G., Seneci C.A., Giataganas P., Yang G-Z. In IEEE International Conference on Robotics and Automation 2018. ©2018 IEEE.

The final scaling factor is computed as:

$$s_b \times (1 + s_c \times k_s \times k_a \times k_d),$$

where s_b is the base scaling factor of the system. Each of the first three components is parametrized by one main coefficient regulating that component's linear drop off rate, i.e the (small) speed at which k_s becomes 0, the (large) angle at which k_a becomes 0, and the (small) distance at which k_d becomes 0. These have all been detailed previously in Table 4.1.

In the previous chapter, these parameters were all manually optimised over several iterations of the system to provide the desired behaviour. Additionally, the inherent safety of motion scaling means that users can always stop their motion to stop the robot. However, one aspect not considered in the previous work is whether the behaviour displayed by the system is one that the user expects. In particular, problems can occur during rapid motions when the user is essentially controlling the robot without visual feedback, i.e. in open loop. Master-slave robots work as a closed-loop control system where the user is closing the loop visually, constantly adjusting their motions according to what they see. However, during rapid motions, users will instead rely on their expectation of the system behaviour to decide how far to move their hand. If the user's expected robot behaviour does not match the real system due to the change in behaviour caused by the adaptive motion scaling, the instrument may not arrive at the desired position. If the system is overtuned for a given user (i.e. too sensitive relative to the user's expectation), the instrument might move further than the user expects it to when performing rapid motions.

The component k_d was initially introduced as a safety precaution against overshoots, as it will fall to zero when the instrument gets close to its destination. However, this again depends on how the corresponding coefficient is tuned and the user's expectation of the system behaviour. Conversely, an undertuned system will not move the instrument far enough, and the user may have to clutch the master device to finish the motion. Thus, as with a number of other heuristically tuned user interfaces, there is a strong argument to be made for wanting a principled optimisation method to derive these parameters. This chapter is concerned with the optimisation of these four parameters, and how to design a realistic pro-

tocol for integrating user-specific parameter optimisation in the design of human-interfaces. To this end, Bayesian optimisation approaches with Gaussian process priors are explored for their highly flexible and data-efficient properties. Finally, a user study is conducted with two different Bayesian optimisation approaches to find the optimal robot behaviour in a user-specific manner. A graded approach, in which users are asked to numerically rate the performance of the system, and a preference approach, in which users are asked to specify pairwise preference relationships between different sets of robot behaviour. The results show the strong performance of Bayesian optimisation, yielding better task times and robot behaviour than with the manually optimised parameters.

5.2 Human-Robot Interface Optimisation

5.2.1 Overview of optimisation Techniques and Requirements

The objective of this work is to provide an approach for finding the user-specific parameters maximizing the intuitiveness of the adaptive motion scaling control scheme. Let $f(\boldsymbol{x})$ be the unknown, black-box, real-valued function representing how intuitive the system is to use for a given set of parameters $\boldsymbol{x}$, with $\boldsymbol{x} \in D$ where $D \subset \mathbb{R}^n$ is the domain of valid parameter values and n the number of parameters to optimize. Finding the set of parameters

$$\boldsymbol{x_{max}} = \underset{\boldsymbol{x} \in \mathcal{D}}{\operatorname{argmax}} f(\boldsymbol{x}) \tag{5.1}$$

is a global optimisation problem within D. Additionally, the human-robot interface dimension of the presented problem justifies that the optimisation method meet the following criteria:

- The optimisation method must work with as few samples as possible, since sampling f requires the user to perform a task and give some feedback.
- The optimisation method cannot make any assumptions on f, other than certain assumptions of smoothness.
- The optimisation method should be robust to noise.

While a large body of works exists on the topic of nonlinear optimisation, Bayesian optimisation is one of the most attractive strategies under these criteria [165]. In addition to being highly data-efficient under few samples, Bayesian inference methods provide a natural way of handling uncertainty and trading-off exploitation and exploration [166, 167]. Bayesian optimisation has been successfully used in a number of mobile robotic applications such as gait optimisation for bipedal locomotion [168], gait adaptation after incurring damage in a 6-legged robot [169], gaze planning [170], and path planning [171]. In [172] a Bayesian optimisation approach was used to tune the manner in which a human-transfer assistant robot lifts patients. Typical Bayesian optimisation methods rely on a grading scheme to evaluate the unknown function f. However, in applications such as the one presented in this paper where f can take arbitrary values, it can be possible to evaluate f using only preference relations between pairs of samples [173]. This approach can be useful when expressing preference between two samples is easier than giving numerical values to grade them, such as when searching for an optimal material design in computer graphics [167].

While Bayesian optimisation approaches have been highly successful in the fields mentioned above, to the authors' knowledge they have not been used before to optimize the behaviour of master-slave surgical systems. This work takes advantage of the strong properties of Bayesian optimisation to improve the behaviour of the adaptive motion scaling algorithm, in a user-specific manner. Furthermore, this work utilizes and evaluates both a grade-based optimisation approach and a preference-based approach. The usefulness of these methods is validated in a user-study, showing a clear preference for the results of the Bayesian optimisation over the manual optimisation. Additionally, the performance of the graded approach shows a statistically significant improvement over the manual optimisation.

5.2.2 Gaussian Processes for Bayesian Learning

Bayesian inference is concerned with updating a belief about a specific proposition based on the data available. At its core, Bayesian inference presents itself as a subjective predictor, based solely on a prior belief and the likelihood of the observed data. Let $\boldsymbol{w}$ represent the parameters of a generic model, and D represent

the available data. Given the prior belief for the model $p(\boldsymbol{w})$, and the likelihood of the data given the model $p(D \mid \boldsymbol{w})$, the posterior distribution $p(\boldsymbol{w} \mid D)$ is given by Bayes' rules as

$$p(\boldsymbol{w} \mid D) = \frac{p(D \mid \boldsymbol{w})p(\boldsymbol{w})}{p(D)}, \tag{5.2}$$

where $p(D)$ is the marginal likelihood. While the idea of having to provide prior beliefs may be off-putting at first, formulating learning problems under Bayesian approaches can have practical advantages. Models where the posterior distribution is tractable are capable of propagating uncertainty and can yield very flexible models. As will be seen below, the inherently evidence-based and probabilistic nature of Bayesian learning allows for the creation of highly data-efficient models.

As a Bayesian method, Bayesian optimisation requires a prior distribution. While many different models could be used to fit the requirements of Bayesian optimisation priors, Gaussian Processes (GP) [166] are commonly used in practice due to the useful properties they exhibit. The main workhorse of Bayesian optimisation used in this thesis is the Gaussian Process. Appendix C introduces Gaussian Processes in the context of Bayesian linear regression, showing that they can appear from an extended Bayesian linear regression model with an infinite number of basis functions. The unfamiliar reader is invited to refer to appendix C, as it establishes key notation and concepts useful for this chapter.

A Gaussian Process is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution[166]. A Gaussian Process is fully specified by defining its mean and covariance functions. A real-valued unknown function $f(\boldsymbol{x})$ defined for input vectors $\boldsymbol{x} \in \mathbb{R}^d$ is modelled by a Gaussian Process when:

$$\begin{aligned} f(\boldsymbol{x}) &\sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x'})) \\ m(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})] \\ k(\boldsymbol{x}, \boldsymbol{x'}) &= \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x'}) - m(\boldsymbol{x'}))]. \end{aligned} \tag{5.3}$$

To every input $\boldsymbol{x}_i$ corresponds a Gaussian distributed random variable $f_i \triangleq f(\boldsymbol{x}_i)$, with mean $m(\boldsymbol{x}_i)$ and covariance $k(\boldsymbol{x}_i, \boldsymbol{x}_i)$. Additionally, any collection of such random variables, e.g. as generated by input vectors $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, is jointly Gaussian distributed. Let $\boldsymbol{f} \triangleq f_{1:n}$ be the collection of these random variables,

then

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{m}, K), \tag{5.4}$$

where $\boldsymbol{m}$ is the joint distribution mean vector with entries $m_i = m(\boldsymbol{x}_i)$, and K is the joint distribution's covariance matrix whose entries are defined as $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. An important observation that can be made from the above formulation is that Gaussian Processes do not require any explicit modelling for the process f other than the covariance function and mean prior. It is thus easy to see the link between Gaussian Processes with Bayesian linear regression, when the parameters have been marginalized out and the covariance is specified using the kernel trick.

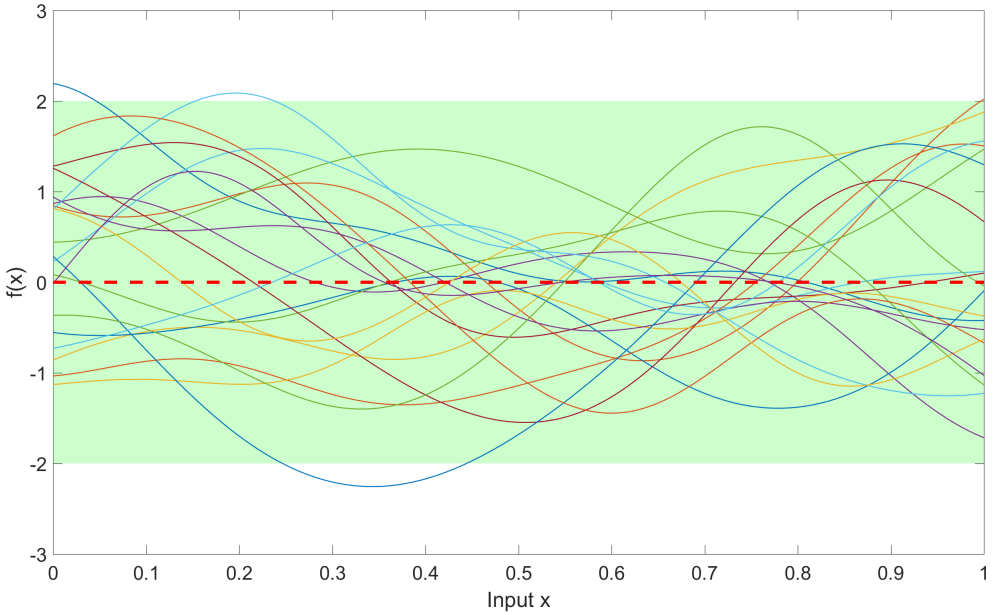


Figure 5.1: Functions drawn from a zero-mean Gaussian process. The dashed red line represents the mean. The green area represents 2 standard deviations from the mean prior.

As the set of possible inputs is infinite ($\boldsymbol{x} \in \mathbb{R}^d$), a Gaussian Process can be interpreted as a distribution over functions. This distribution can be used to model our belief about the possible forms of the unknown unknown function f. As such, Gaussian Processes can be used as priors for Bayesian inference, where the mean and covariance functions express our beliefs about the core properties of f in the absence of data. Figure 5.1 shows functions drawn from a zero-mean Gaussian process. Note that, as expected, the functions are mostly centered around

the zero-mean prior and contained within 2 standard deviations from the mean. The following section shows how to derive the posterior and predictive posterior distributions in the presence of data.

Gaussian Process Prediction

Let X be a set of training samples, and X_* be a set of test points. Let $K(X_1, X_2)$ refer to the covariance matrix evaluated at all pairs of inputs for two arbitrary sets of samples X_1 and X_2. The following matrices can be computed, their shorthand introduced for simplicity: the covariance matrix of the training set $K = K(X, X)$, the covariance matrix $K_* = K(X, X_*)$ of the training set with the test set, and the covariance matrix $K_{**} = K(X_*, X_*)$ of the test set. Let $\boldsymbol{f}$ be the values of the objective function at the training samples, and $\boldsymbol{f_*}$ the values of the objective function corresponding to the test samples. Finally, let $\boldsymbol{m}$ be the training means and $\boldsymbol{m}_*$ be the test means. Then:

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f_*} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{m_*} \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} \right). \tag{5.5}$$

Using the properties for conditioning a jointly distributed Gaussians [166], the conditional distribution of the predictive posterior for the test samples given the training samples is

$$\boldsymbol{f_*}|\boldsymbol{f} \sim \mathcal{N}\left(\boldsymbol{m}_* + K_*^T K^{-1}(\boldsymbol{f} - \boldsymbol{m}), K_{**} - K_*^T K^{-1} K_*\right). \tag{5.6}$$

The posterior process is also a Gaussian process with mean function $m_p(\boldsymbol{x})$ and covariance function $k_p(\boldsymbol{x}, \boldsymbol{x}')$ specified as

$$\begin{aligned} m_p(\boldsymbol{x}) &= m(\boldsymbol{x}) + K(X, \boldsymbol{x})^T K^{-1}(\boldsymbol{f} - \boldsymbol{m}) \\ k_p(\boldsymbol{x}, \boldsymbol{x}') &= k(\boldsymbol{x}, \boldsymbol{x}') - K(X, \boldsymbol{x})^T K^{-1} K(X, \boldsymbol{x}'). \end{aligned} \tag{5.7}$$

Figure 5.2 illustrates the posterior Gaussian process obtained from the Gaussian process prior illustrated in Figure 5.1 combined with the observed data. Note how the uncertainty vanishes at the sampled points, and the posterior mean tries to go back to the zero-mean prior when far from the observed data.

Equations 5.6 and 5.7 consider that the user has direct access to noise-free samples of the unknown function f. In many scenarios however, noise-free as-

sumptions are not realistic. Fortunately the Gaussian Process formulation presented above makes it easy to add a *iid* Gaussian noise term ε of variance σ_n^2. Let $y = f(\boldsymbol{x}) + \varepsilon$ be the noisy observations under this new formulation. Then

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f_*} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m} \\ \boldsymbol{m_*} \end{bmatrix}, \begin{bmatrix} K + \sigma_n^2 I & K_* \\ K_*^T & K_{**} \end{bmatrix} \right), \tag{5.8}$$

and so finally the predictive posterior distribution under Gaussian noise is

$$\boldsymbol{f_*}|\boldsymbol{y} \sim \mathcal{N}\left(\bar{\boldsymbol{f}}_*, \Sigma_*\right) \tag{5.9}$$

$$\bar{\boldsymbol{f}}_* = \boldsymbol{m}_* + K_*^T (K + \sigma_n^2 I)^{-1} (\boldsymbol{f} - \boldsymbol{m}) \tag{5.10}$$

$$\Sigma_* = K_{**} - K_*^T (K + \sigma_n^2 I)^{-1} K_*. \tag{5.11}$$

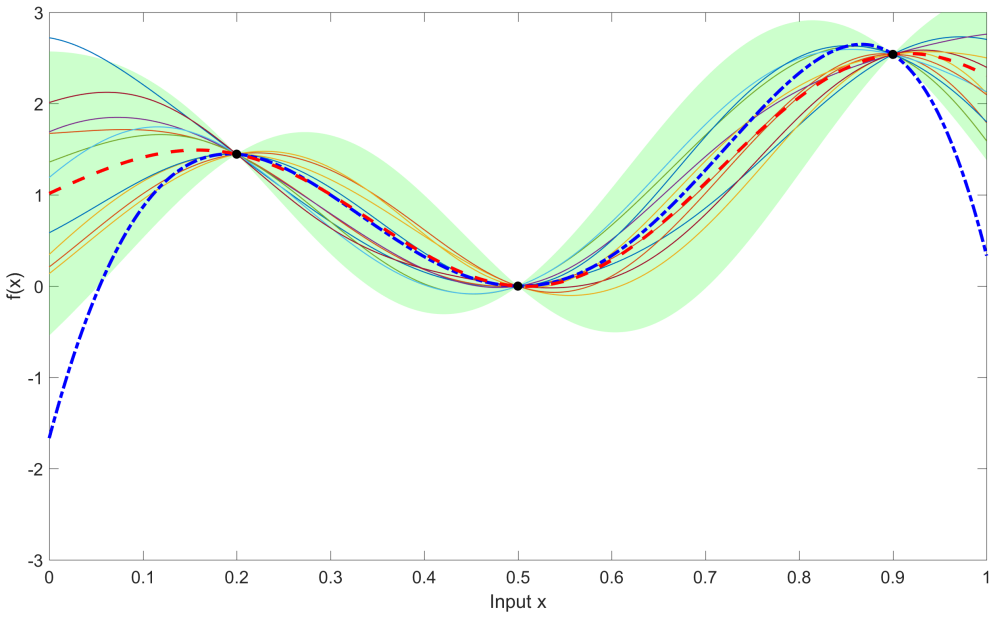


Figure 5.2: Functions drawn from a zero-mean Gaussian process posterior. The dashed red line represents the posterior mean, the dashed blue line represents the unknown objective function from which the observations (indicated by black circles) are drawn. The green area represents 2 standard deviations from the posterior mean.

Covariance Functions

The choice of the covariance function, or kernel, is arguably the main component of choosing a Gaussian Process model. While informative mean priors can be leveraged to great effect, as shown in chapter 6, often times un-informative mean priors such as a constant zero mean are selected when no information about the general shape of f is available. The choice of the covariance function, however, dictates a more subtle yet defining property of f. By defining how groups of input vectors are correlated, the covariance function dictates the behaviour of the surrogate model for f. Intuitively, this can be understood as answering questions such as "Does f vary quickly or slowly?", "Is f periodic?", or even "Does f display very different behaviours in different areas of the input space?".

Many different popular kernel types exist, such as linear kernels, rational quadratic kernels, and periodic kernels. As the aim of this section is not to review different kernels, but to present and justify the kernels used, other kernels will not be discussed. For more information on different kernels the reader is referred to [166]. In this chapter the objective function is a measure of the intuitiveness of a human-robot interface. As all the parameters to optimize have a linear impact on their respective component, it is expected that the objective function will vary smoothly. Under these assumptions, the choice of a smoothly varying kernel is justified and important. As such, the main kernel used in this thesis is the squared exponential (SE) kernel, defined as

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}')^T(\boldsymbol{x}-\boldsymbol{x}')}{2l^2}\right). \tag{5.12}$$

The SE kernel is infinitely differentiable, and in the Bayesian linear model view can be considered as an infinite collection of basis functions. It provides a smoothly-varying prior over functions, with two hyperparameters l and σ. The characteristic lengthscale l defines how rapidly the functions drawn from the corresponding Gaussian Process are likely to vary. Small lengthscales lead to quickly varying functions, while high lengthscales lead to slowly varying functions. This is illustrated in Figure 5.3. The hyperparameter σ is a scaling factor representing the signal variance.

In some cases, the different dimensions of the input vectors $\boldsymbol{x}$ can vary significantly differently. This can be the case when multiple independent sources of

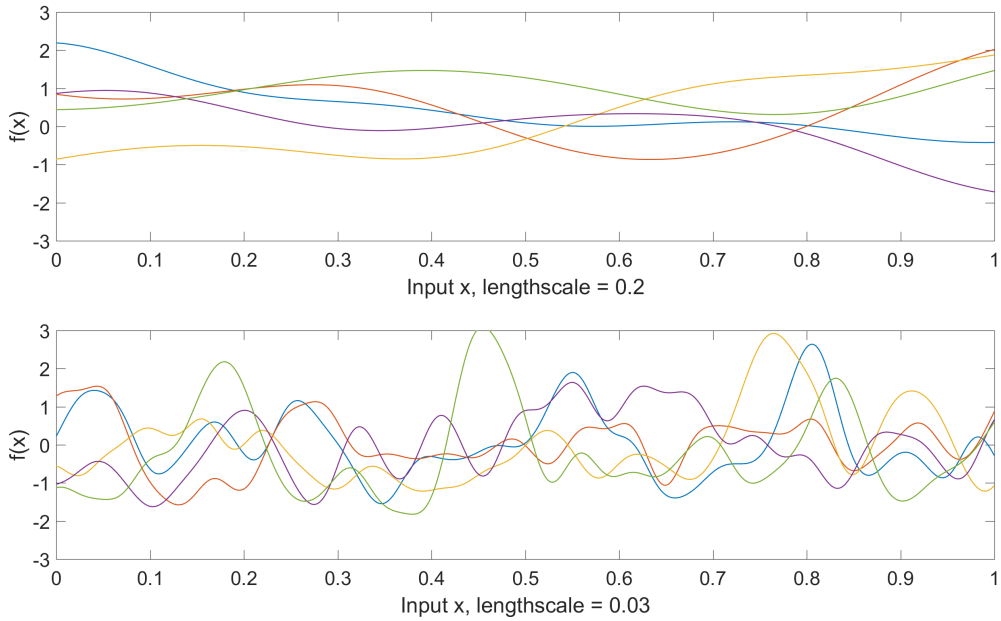


Figure 5.3: Functions drawn from a zero-mean Gaussian Process with an SE kernel of lengthscale $l = 0.2$ (top), and $l = 0.03$ (bottom).

information are compounded into a single input, such as a vector $\boldsymbol{x}_t = [v_t, \alpha_t]^T$ combining some velocity v and angular α information at a time t. A formulation of the SE kernel allowing different lengthscales for each dimension of the input vector $\boldsymbol{x}$ can be written as

$$k_{SE\ ARD}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-(\boldsymbol{x} - \boldsymbol{x}')^T L(\boldsymbol{x} - \boldsymbol{x}')\right), \tag{5.13}$$

where L is a diagonal matrix containing the different lengthscales:

$$L = \frac{1}{2}\begin{bmatrix} l_1^{-2} & & \\ & \ddots & \\ & & l_n^{-2} \end{bmatrix}. \tag{5.14}$$

This kernel is referred to as an Automatic Relevance Determination (ARD) kernel. This is because the lengthscale of the different dimensions can also be interpreted as the relevance of those dimensions towards the output. A high lengthscale for a given dimension indicates low change in the output when the input changes along that dimension. This effectively allows the model to "weigh" the importance of

the different dimensions, since high lengthscale dimensions are less relevant to the manner in which the output varies than dimensions with smaller lengthscales.

The lengthscales, signal variance σ^2, and noise variance σ_n^2, constitute the hyperparameters of the Gaussian Process model under the previously presented likelihood and kernels. These parameters can be manually set when sufficient prior information is available on the system. It is worth noting that it can be fairly easy to make reasonable estimations of these coefficients in some cases. For example, by normalizing the input values to the $[0, 1]$ range along each dimension, the lengthscale in an SE kernel can be chosen to represent a rough estimation of how quickly the function is expected to vary as a whole. Due to the inherent flexibility of the Gaussian Process, this can yield effective results with minimal design considerations.

Another approach is to search for the optimal hyperparameters. A standard way of doing this is by maximizing the marginal likelihood, or *evidence*, of the model. This approach can be very useful when combined with ARD kernels, allowing the model to automatically determine which dimensions have a strong impact on the output and which do not. This approach will be detailed in chapter 6. It is important to point out that this manner of finding the optimal hyperparameters requires sufficient data, lest the optimisation result in overfitting. For applications with very few samples, such as in this chapter, it is best to leverage as much prior knowledge as possible to obtain reasonable values for the hyperparameters.

5.2.3 Gaussian Processes for Preference Relationships

In some optimisation cases the objective function does not take physically meaningful values, such as in this chapter where the objective function is a representation of how much users like certain robot behaviors. In these cases it can be interesting to directly compare samples against one another, and try to optimize the objective function through preference ratings. In [173], an approach is presented showing how this can be achieved using a likelihood function comparing two samples.

Under this proposition, the dataset D consists of M ranked pairs establishing preference between samples:

$$D = \{\boldsymbol{r}_i \succ \boldsymbol{c}_i, i = 1 \dots M\}, \tag{5.15}$$

where $\succ$ indicates the preference relationship between $\boldsymbol{r}$ and $\boldsymbol{c}$. $\boldsymbol{r}_i$ and $\boldsymbol{c}_i$ are samples from the training dataset $\{\boldsymbol{x}_j, j = 1 \dots n\}$. The key concept behind this approach for preference is the assumption that there exists a latent function value $\boldsymbol{f} \triangleq f(\boldsymbol{x}_i)$ associated with every training sample $\boldsymbol{x}_i$ such that the function values $\{f(\boldsymbol{x}_i)\}$ preserve the preference relationships dictated by equation 5.15. Note that the latent values that f takes do not have any physical meaning, as only preference relationships are specified. The Bayesian treatment of this approach consists in imposing a Gaussian process prior on the latent function values $\boldsymbol{f}$ and choosing a suitable likelihood. The chosen Gaussian Process prior is zero mean and of covariance matrix K:

$$p(\boldsymbol{f}) = \frac{1}{\sqrt{|2\pi K|}} \exp\left(-\frac{1}{2}\boldsymbol{f}^T K^{-1} \boldsymbol{f}\right). \tag{5.16}$$

The likelihood function defined in absence of noise is expressed as

$$p(\boldsymbol{r}_i \succ \boldsymbol{c}_i \mid f(\boldsymbol{r}_i), f(\boldsymbol{c}_i)) = \begin{cases} 1, & \text{if } f(\boldsymbol{r}_i) \geq f(\boldsymbol{c}_i) \\ 0, & \text{otherwise.} \end{cases} \tag{5.17}$$

Similarly to section 5.2.2, the latent function f can be augmented with *iid* Gaussian noise of zero mean and variance σ_n^2: $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. The stochastic value functions $v(.)$ for items $\boldsymbol{r}$ and $\boldsymbol{c}$ is expressed as

$$v(\boldsymbol{r}_i) = f(\boldsymbol{r}_i) + \varepsilon \tag{5.18}$$

$$v(\boldsymbol{c}_i) = f(\boldsymbol{c}_i) + \varepsilon. \tag{5.19}$$

The likelihood in equation 5.17 then becomes

$$p(\boldsymbol{r}_i \succ \boldsymbol{c}_i \mid f(\boldsymbol{r}_i), f(\boldsymbol{c}_i)) = p(v(\boldsymbol{r}_i) > v(\boldsymbol{c}_i) \mid f(\boldsymbol{r}_i) f(\boldsymbol{c}_i)) \tag{5.20}$$

$$= p(\varepsilon - \varepsilon < f(\boldsymbol{r}_i) - f(\boldsymbol{c}_i)) \tag{5.21}$$

$$= \Phi(Z_i), \tag{5.22}$$

with

$$Z_i = \frac{f(\boldsymbol{r}_i) - f(\boldsymbol{c}_i)}{\sigma_n \sqrt{2}}, \tag{5.23}$$

and $\Phi(.)$ is the Cumulative Distribution Function CDF of the standard normal distribution. This stems from the observation that $\varepsilon - \varepsilon \sim \mathcal{N}(0, 2\sigma_n^2) = \mathcal{N}(0, (\sqrt{2}\sigma_n)^2)$

from the difference of two independent Gaussians (ε is *iid*). The likelihood of the model can then be written as

$$p(D \mid \boldsymbol{f}) = \prod_{i=1}^{M} p(\boldsymbol{r}_i \succ \boldsymbol{c}_i \mid f(\boldsymbol{r}_i), f(\boldsymbol{c}_i)). \tag{5.24}$$

Using Bayes' rule the posterior probability can be expressed below

$$p(\boldsymbol{f} \mid D) = \frac{p(\boldsymbol{f})}{p(D)} \prod_{i=1}^{M} p(\boldsymbol{r}_i \succ \boldsymbol{c}_i \mid f(\boldsymbol{r}_i), f(\boldsymbol{c}_i)), \tag{5.25}$$

where $p(\boldsymbol{f})$ is given by the Gaussian Process prior in equation 5.16, and $p(D)$ is the normalization constant. Unlike in section 5.2.2, the non-Gaussian likelihood function means that the posterior is no longer analytically tractable. A solution to find the posterior distribution consists in applying the Laplace approximation by Taylor-expanding the log-posterior around a point $\hat{\boldsymbol{f}}$:

$$\log p(\boldsymbol{f} \mid D) = \log p(\hat{\boldsymbol{f}} \mid D) + \boldsymbol{g}^T(\boldsymbol{f} - \hat{\boldsymbol{f}}) - \frac{1}{2}(\boldsymbol{f} - \hat{\boldsymbol{f}})^T H(\boldsymbol{f} - \hat{\boldsymbol{f}}), \tag{5.26}$$

where $g = \nabla_{\boldsymbol{f}} \log p(\boldsymbol{f} \mid D)$ and $H = -\nabla_{\boldsymbol{f}} \nabla_{\boldsymbol{f}} \log p(\boldsymbol{f} \mid D)$. By expressing the Taylor-expansion at the maximum a posteriori (MAP) $\hat{\boldsymbol{f}} = \boldsymbol{f}_{MAP}$, the gradient g disappears leaving the following approximation:

$$\begin{aligned} p(\boldsymbol{f} \mid D) &\approx p(\boldsymbol{f}_{MAP} \mid D) \exp\left[-\frac{1}{2}(\boldsymbol{f} - \boldsymbol{f}_{MAP})^T H(\boldsymbol{f} - \boldsymbol{f}_{MAP})\right] \\ &\propto \mathcal{N}(\boldsymbol{f}_{MAP}, H^{-1}). \end{aligned} \tag{5.27}$$

The expressions of the gradient $\boldsymbol{g}$ and the Hessian matrix H can be obtained algebraically, as detailed in [173, 167]. As the Hessian matrix is positive semi-definite, the MAP estimate $\boldsymbol{f}_{MAP}$ can then be found using Newton-Raphson recursion.

Finally, for a set of test points X_*, and using the same notation for the covariance matrices between the test and training sets as in section 5.2.2, the predictive posterior distribution $p(\boldsymbol{f}_* \mid D)$ is written as a convolution of two Gaussians:

$$p(\boldsymbol{f}_* \mid D) = \int p(\boldsymbol{f}_* \mid \boldsymbol{f}_{MAP}) p(\boldsymbol{f}_{MAP} | D) d\boldsymbol{f}_{MAP} \tag{5.28}$$

$$\propto \mathcal{N}(K_*^T K^{-1} \boldsymbol{f}_{MAP}, K_{**} - K_*^T H^{-1} K_*). \tag{5.29}$$

5.2.4 Bayesian optimisation

In an optimisation scheme, the aim is to find the best evaluation of the objective function f. This is achieved by placing a surrogate model on f, sampling the objective function at different inputs, and computing the expected maximum of f according to the surrogate model. In this thesis, Gaussian Process priors are placed and the predictive posterior distributions are computed as described previously. The predictive posterior distribution can then be sampled to find the optimal input $\boldsymbol{x}$ maximising f.

An important part of this process consists in determining where next to sample the objective function. To this aim, an *acquisition function* can be defined. For any given input $\boldsymbol{x}$, the acquisition function associates a value representing how beneficial it would be to select $\boldsymbol{x}$ as the next input to sample. The optimal next sample according to that acquisition function is the input $\boldsymbol{x}$ that maximises the acquisition function. The benefit associated with an input $\boldsymbol{x}$ depends on how the acquisition function is defined. In general, choosing an acquisition function corresponds to making a trade-off between *exploring* parts of the input space with high uncertainty, and *exploiting* the current best evaluations of f to generate an even better estimate. This section presents several common acquisition functions used in this thesis.

An intuitive acquisition function is the probability of improvement (PI) defined as

$$PI(\boldsymbol{x}) = p(f(\boldsymbol{x}) \geq f(\boldsymbol{x}^+)) \tag{5.30}$$

$$= \Phi\left(\frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+)}{\sigma(\boldsymbol{x})}\right), \tag{5.31}$$

where $\boldsymbol{x}^+ = \operatorname{argmax}_{\boldsymbol{x}_i \in \boldsymbol{x}_{1:t}} f(\boldsymbol{x}_i)$ is the sample corresponding to the highest past observation (considering t past observations). The mean $\mu(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$ are sampled from the predictive posterior distribution for a given test input $\boldsymbol{x}$. The main limitation of this acquisition function is that it will always favour exploitation over exploration: a test input with an infinitely small positive difference $\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+)$ will always be selected over one with a slightly lower expected mean but large variance. This can be remedied somewhat by introducing a coeffi-

cient ξ such that

$$PI(\boldsymbol{x}) = \Phi\left(\frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+) - \xi}{\sigma(\boldsymbol{x})}\right). \tag{5.32}$$

This has for effect to make the acquisition function select the input most likely to offer an improvement of at least ξ, as opposed to any improvement at all. Furthermore, the value of ξ can be varied over time, e.g. to favour exploration early on and exploitation later.

Another notable acquisition function is the expected improvement acquisition (EI) function, used to trade off exploration and exploitation. It can be shown [167, 165] that the expected improvement can be derived analytically from the posterior distribution such that:

$$EI(\boldsymbol{x}) = \begin{cases} (\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+))\,\Phi(Z) + \sigma(\boldsymbol{x})\phi(Z), & \text{if } \sigma(\boldsymbol{x}) > 0 \\ 0, & \text{if } \sigma(\boldsymbol{x}) = 0 \end{cases} \tag{5.33}$$

$$Z = \begin{cases} \frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+)}{\sigma(\boldsymbol{x})}, & \text{if } \sigma(\boldsymbol{x}) > 0 \\ 0, & \text{if } \sigma(\boldsymbol{x}) = 0 \end{cases}, \tag{5.34}$$

with $\phi(.)$ denoting the Probability Density Function (PDF) of the standard normal distribution. Unlike the default formulation of the probability of improvement acquisition function, the expected improvement acquisition function considers the magnitude of improvement achievable by a test input $\boldsymbol{x}$. Figure 5.4 illustrates the optimisation process for a one dimensional case. The red vertical line shows the next acquisition point provided by the EI acquisition function. Similarly to the the probability of improvement, a term ξ can be added to the expected improvement acquisition function to tune the trade-off between exploration and exploitation:

$$EI(\boldsymbol{x}) = \begin{cases} (\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+) - \xi)\,\Phi(Z) + \sigma(\boldsymbol{x})\phi(Z), & \text{if } \sigma(\boldsymbol{x}) > 0 \\ 0, & \text{if } \sigma(\boldsymbol{x}) = 0 \end{cases} \tag{5.35}$$

$$Z = \begin{cases} \frac{\mu(\boldsymbol{x}) - f(\boldsymbol{x}^+) - \xi}{\sigma(\boldsymbol{x})}, & \text{if } \sigma(\boldsymbol{x}) > 0 \\ 0, & \text{if } \sigma(\boldsymbol{x}) = 0 \end{cases}. \tag{5.36}$$

Different strategies and scheduling schemes have been explored to choose ξ [167],

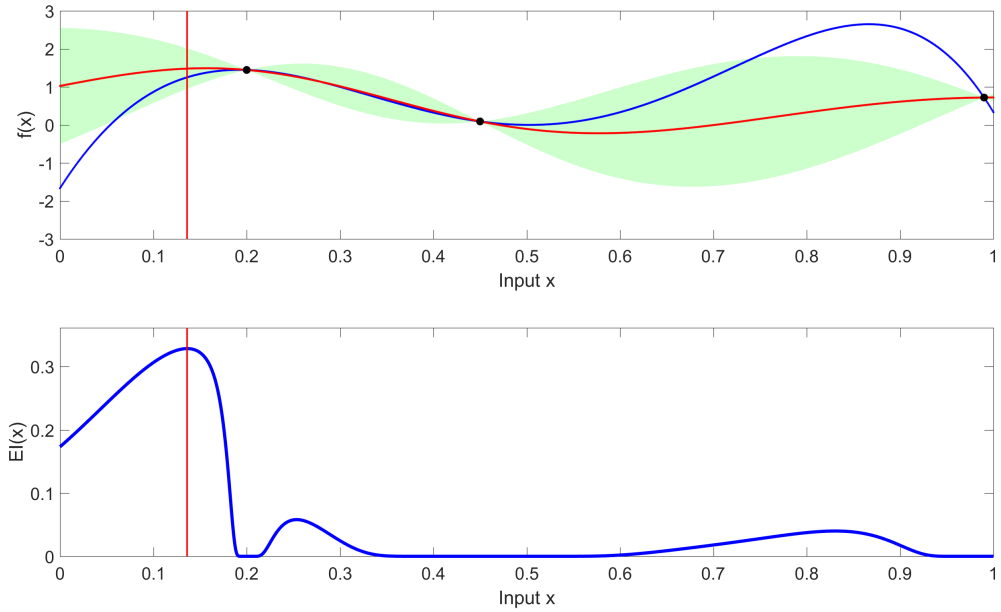


Figure 5.4: 1D Bayesian optimisation. Top: blue curve: true objective function, red curve: predicted mean, Green patch: area representing plus or minus 2 standard deviations from the mean. The bottom blue curve represents the values of the EI acquisition function. The red vertical line shows the optimal next point to sample suggested by the EI acquisition function.

including scaling ξ by the signal variance. Ultimately, these choices reflect design strategies for the optimizer in terms of weighing the different trade-offs, and experimental constraints can help to guide those decisions. As noted in [174], a single acquisition function is unlikely to be suitable in all cases, and so a portfolio of acquisition functions is typically used in practice. Finally, the following acquisition functions present a few different strategies: the maximum expectation function $ME(\boldsymbol{x}) = \sigma(\boldsymbol{x})$ represents pure exploration; Thompson sampling consists in drawing a random function from the Gaussian Process and using its maximum valued input as the next sample; and lastly the random acquisition function simply picks a random next sample.

A comparison of the optimisation process obtained when using both the direct sampling optimisation and the preference based optimisation methods is shown below on a 2D problem. These optimisation approaches will be referred to as the graded approach and preference approach respectively, since in our application direct sampling requires users to numerically grade the performance of the system

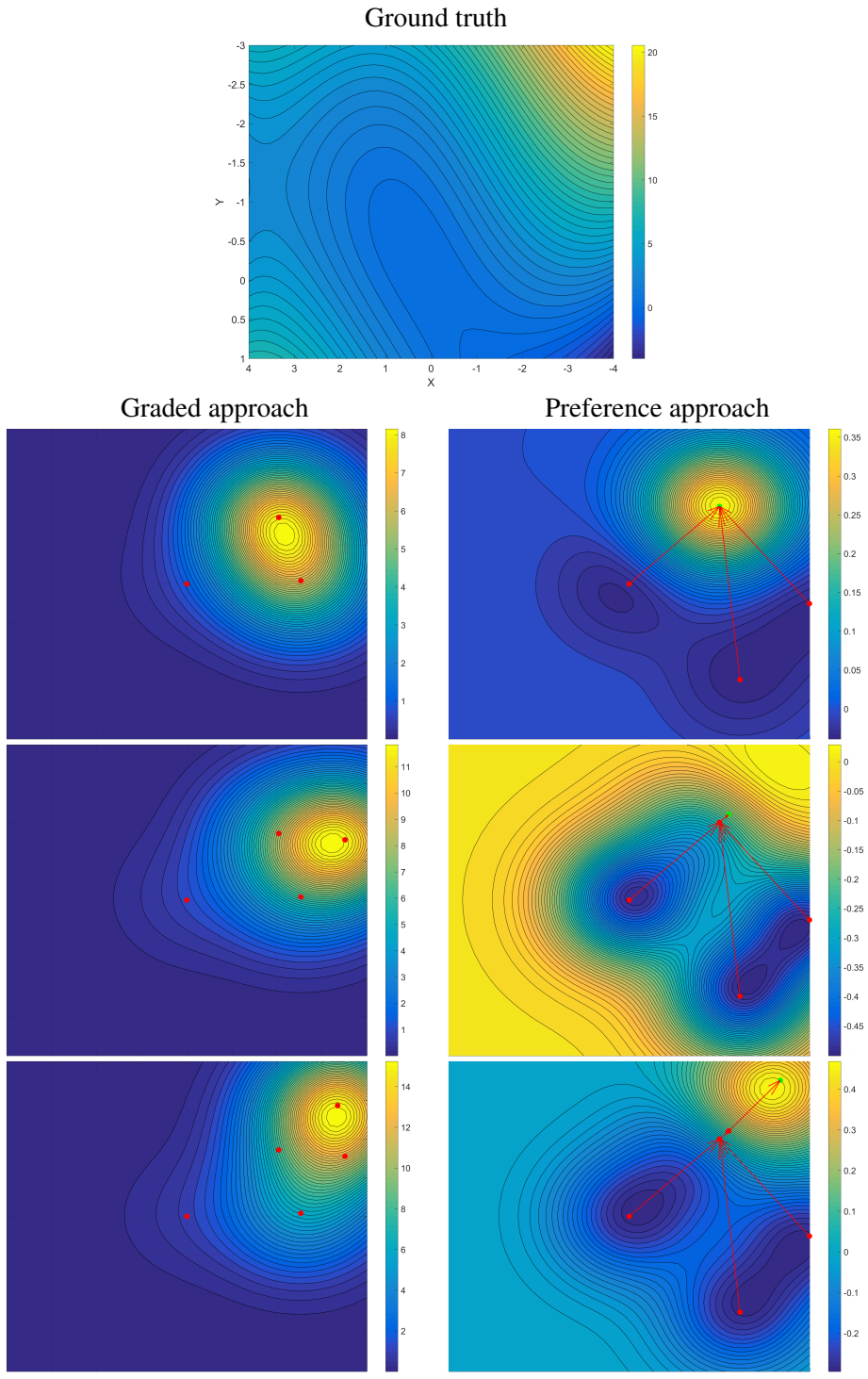


Figure 5.5: 2D Bayesian optimisation of an unknown objective function (top), using a graded approach (left), and a preference approach (right).

for a given set of parameters. Figure 5.5(top) shows the true objective function whose maximum the optimisation process is trying to find. Figure 5.5(left) shows the predicted distribution mean values over the sampled space obtained from the graded approach (i.e. where it is possible to sample function values directly). Figure 5.5(right) shows the predicted distribution mean values over the sampled space obtained from the preference approach. Note that the values of the predicted mean for the preference approach do not correspond to those of the true objective function, since those values are never sampled directly.

Both approaches perform well in this scenario, however an interesting observation can be made with regards to the preference approach. In the graded approach, samples are provided one by one and the acquisition functions naturally provide a suggestion for the next sample. In the preference approach, however, samples are provided two by two. In order to obtain completely new samples, the optimizer would need to specify two next samples. In practice one of the samples is drawn from the list of previous samples under some criteria. In this work, two different such criteria were considered. The first one involves always picking the last sample obtained. The advantage of this approach is that it avoids having to show a previous sample again to the user to refresh their memory, losing time in the process. The second criteria involves always picking the best previous sample out of all samples. While the first criteria seemed to offer practical advantages, the second criteria was found to perform better as the preference test at every new sample effectively reduces to asking the user whether this sample is his or her preferred sample of the entire dataset.

5.3 Application of Bayesian Optimisation to Adaptive Motion Scaling

Having established all the essential requirements for performing Bayesian optimisation using graded and preference approaches, this section presents the application of these methods to the optimisation of the adaptive motion scaling model. Details about the models used and notable differences with the setup from chapter 4 are presented first, followed by the experimental protocol and the results.

5.3.1 System Design and Improvements

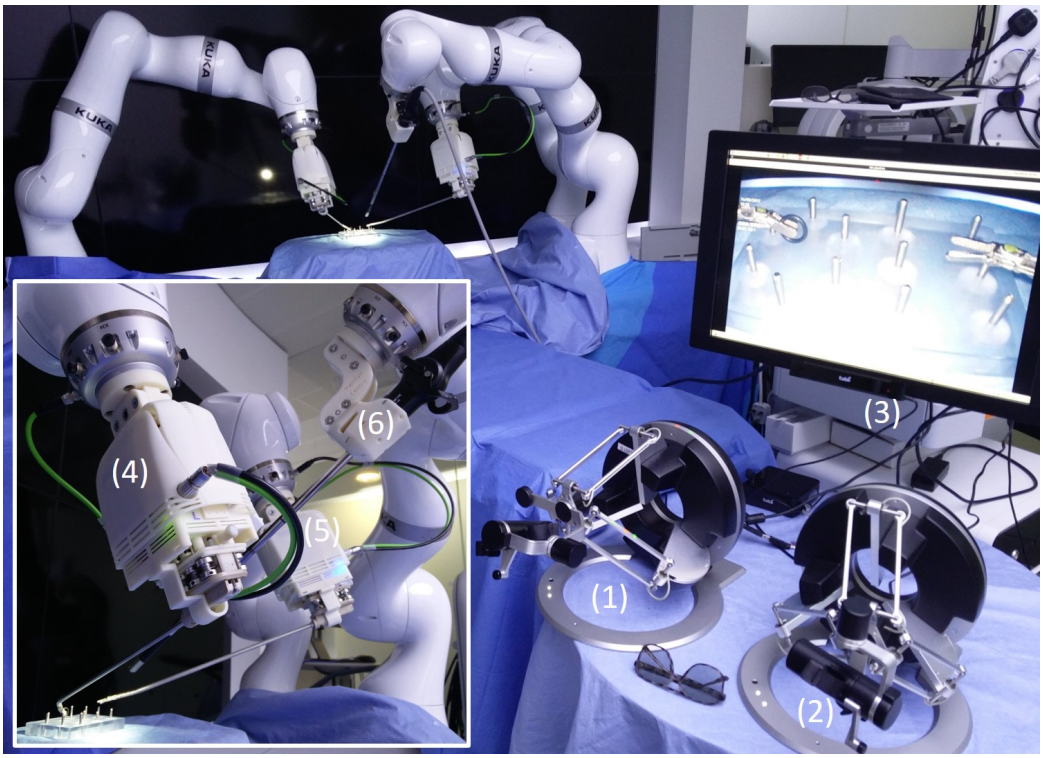


Figure 5.6: Overview of the bimanual master-slave system comprising: 2 Omega.7 devices (1) and (2), a Tobii eye-tracker (3), two actuation units (4) and (5) with their respective instruments, and an Olympus stereoscope (6). Three Kuka iiwa robotic arms are visible in the background.

The presented system is nearly identical to the one presented in chapter 4, allowing for a direct comparison of the manually optimised parameters against the ones obtained from the Bayesian optimisation process. The system is shown in Figure 5.6. An Olympus stereoscope, also illustrated in Figure 3.9 is mounted in place of the previously used custom stereoscope system. The two 6-DoF flexible instruments are actuated via custom-designed actuation units attached to the robotic arms, which recieved several improvements relative to those presented in chapter 4.

A supporting structure was added at the bottom of the instrument casing to provide a secure coupling between the instrument and the actuation unit. The adapter to the Kuka robotic arm was redesigned for additional strength. Together,

these changes serve to reduce the propagation of vibration to the instrument tip. Lastly, the Olympus stereoscope was used to provide a better image quality and depth perception. These changes serve to improve the repeatability and accuracy of the system without affecting it's behaviour. The graded and preference optimisation approaches were implemented from the scratch using a zero-mean Gaussian process as detailed in section 5.2. An SE kernel was used, with a lengthscale $l = \frac{\sqrt{d}}{10}$, where d is the dimension of the input samples ($d = 4$). This increase of the lengthscale relative to the dimensionality can be intuitively understood when considering the range of Euclidean distances possible for a given value of d. Since the SE kernel compares the Euclidean distance between two vectors, if the maximum Euclidean distance possible increases, so should the lengthscale. Given that all input dimension are normalized to the $[0, 1]$ range, the maximum Euclidean distance between two vectors of dimension d is $\sqrt{d}$. As such, a lengthscale of $l = \frac{1}{10}$ in one dimension is equivalent to a lengthscale of $l = \frac{\sqrt{d}}{10}$ in d dimensions. This makes it possible to specify to specify a desired one-dimensional behaviour and retain it for d dimensions. A portfolio of acquisition functions was designed for the optimisation process, containing the expected improvement, Thompson sampling, and maximum exploration acquisition functions.

5.3.2 Experimental Protocol

In order to test each optimisation approach and evaluate their impact on the performance of the system, a user study was conducted. A total of twelve experiments were carried out. Six non-clinical participants with experience using master-slave robotic systems were brought in to train with the adaptive motion scaling. The training lasted for as long as the participants felt was necessary, and until the experimenter felt the subject was proficient enough with the system (minimum training time of 5 minutes). The set of parameters used for the system during the training was the manually optimised parameters, although this was intentionally not revealed to the users.

Each participant performed two sets of experiments, one for the graded approach and one for the preference approach. The order of these sets of experiments was randomised between the participants. Each set of experiments itself consisted of two phases. The first phase was a sampling phase in which the system

was loaded with a specific set of parameters and the user was asked to perform a ring transfer task twice. At the end of the task the user graded the system performance, if during the graded approach set of experiments. If during the preference approach set of experiments, the user told the experimenter whether the system performance he experienced was his favourite so far. At the end of the sampling phase the system predicted the user's optimal set of parameters. The second phase was a performance assessment phase using the result of the optimisation. In a blind study whose order was randomised across participants, the user was then shown either the manual optimisation parameters, or the predicted optimal set of parameters. The user was asked to perform the ring transfer task four times with each set of parameters, during which their performance was recorded. After completing the tasks for both sets of parameters, the users were asked which set of parameters they preferred.

The ring transfer task is illustrated in Figure 5.7 and consisted of the following sequence:

- Both instruments start at the extremities of the peg board, above one of the side pegs.
- Both Omega.7 master devices also start at the outermost edge of their workspace.
- The left instrument starts with the ring.
- The left instrument brings the ring to the right instrument.
- The right instrument takes the ring and the left instrument goes back to its starting position.
- The right instrument brings the ring to the left instrument.
- The left instrument takes the ring and the right instrument goes back to its starting position.

Each sampling phase consisted of 12 samples. In addition, participants were allowed to repeat any given task as many times as they wanted if they felt unsure about how to rate the system's performance. Given that the ring transfer task had to be performed at least twice for each sample, every participant performed



Figure 5.7: Ring transfer task

the ring transfer task at least 64 times. The peg board used was the same one used in [175], thus preserving the dimensions of the task. Graded values were given between 0 and 20, where 20 is the best performance. These values where internally shifted to a -10 and 10 range so as to have a 0 mean. The 0 to 20 range was used with participants as they were observed to be reluctant to give negative values. The same metrics as in the previous chapter were used to quantify the performance of the system, namely the task completion time and the number of times the system was clutched.

5.3.3 Results

The outcome of the second phase of each set of experiments, in which participants blindly tested the result of the Bayesian optimisation against the manual optimisation, is that in all but one case users preferred the result of the Bayesian optimisation over the manually optimised parameters. This results in a 91.7% success rate (11 experiments out of 12) for the Bayesian optimisation methods in terms of system behaviour improvements. This is particularly noteworthy as the participants had been unknowingly biased towards the manually optimised parameters

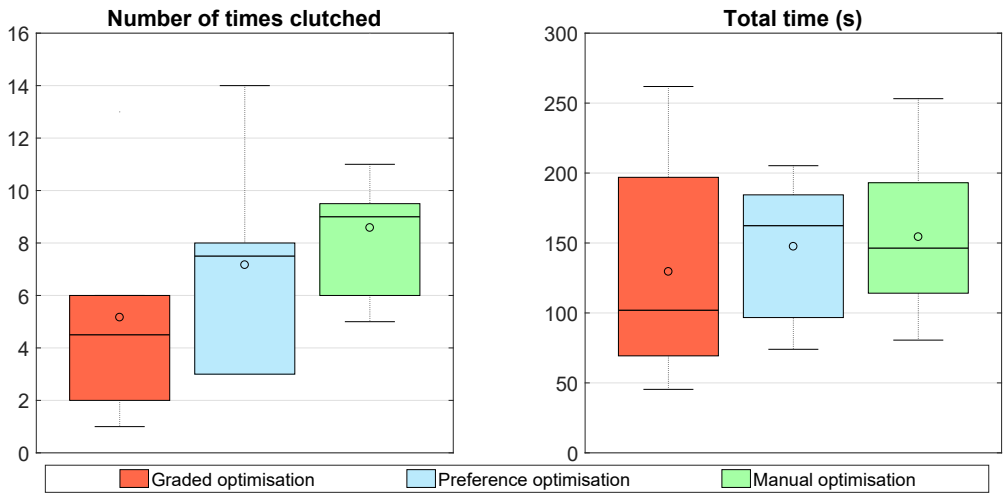


Figure 5.8: optimisation study performance results

by using them during the training session.

Additionally the performance assessment shows that the results of the Bayesian optimisation were at least as good as the results of the manual optimisation. Figure 5.8 shows the results of the performance evaluation for the different types of optimisation. Numerical values for the mean and median of the task completion time and number of times the system was clutched are listed in Table 5.1. The statistical significance between the different optimisation approaches for those metrics are listed in Table 5.2.

Note that no statistically significant difference is present in terms of time spent to perform the tasks. This is remarkable as the manual parameters were optimised for responsiveness and performance. In terms of number of times the system had to be clutched, the preference approach shows no statistically significant differ-

	Mean			**Median**		
	Manual	Preference	Graded	Manual	Preference	Graded
Number of times clutched	8.58	7.17	5.17	9.00	7.50	4.50
Total task time (s)	154.4	147.5	129.6	146.3	162.4	101.9

Table 5.1: Mean and median results from the blind study comparing the different optimisation approaches.

	Manual	Preference	Graded
Manual	-	0.282	0.047
Preference	0.282	-	0.292
Graded	0.047	0.292	-

	Manual	Preference	Graded
Manual	-	0.820	0.335
Preference	0.820	-	0.588
Graded	0.335	0.588	-

Table 5.2: p values from Wilcoxon ranksum tests for the different optimisation approaches. Left: values comparing the number of times the system was clutched. Right: values comparing task completion times.

ences compared to the manual optimisation. However, the graded approach shows a statistically significant improvement of 50% over the manual optimisation, with a p-value of 0.0477. This can be explained by the fact that the performance-oriented, manually optimised parameters would sometimes be the cause of small overshoots or imprecisions for some users, who would then have to readjust their position.

Interestingly, a common optimisation result seemed to involve a compromise between reducing the maximum scaling factor s_c, and increasing the sensitivity via k_d and k_s. In this manner users would reach the top speed more easily, but the reduction in top speed meant that users did not overshoot. Figure 5.9 shows

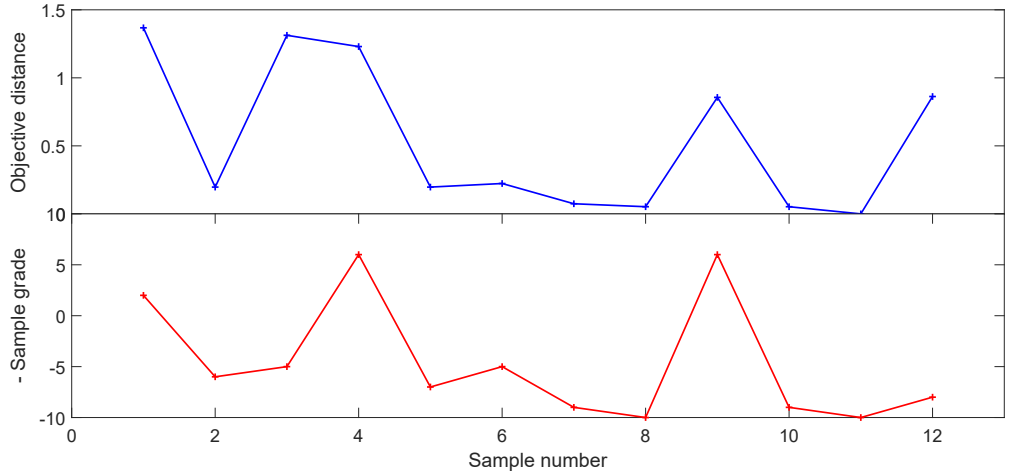


Figure 5.9: Coefficient progress for the graded approach. Blue: normalized distance to the optimal parameter set. Red: negative of grade given. The objective distance refers to the normalized Euclidean distance between the current parameter vector and the final parameter vector.

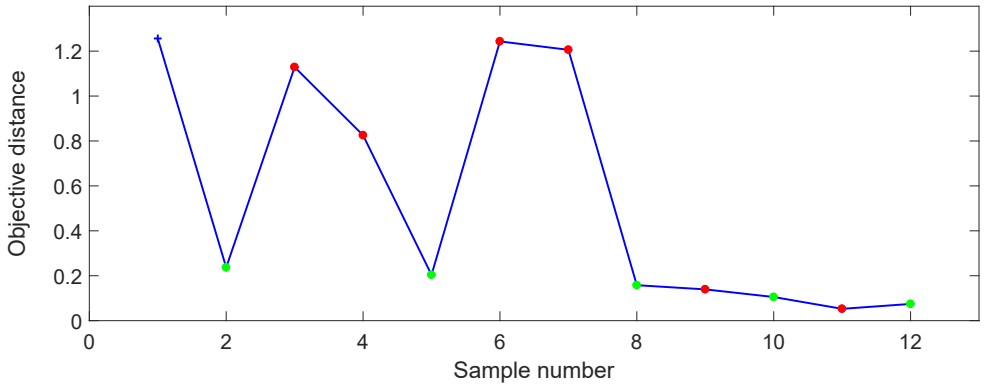


Figure 5.10: Coefficient progress for the preference approach. Blue: normalized distance to the optimal parameter set. Red dot: sample was not preferred to the previous best sample. Green dot: sample was chosen as the new best sample. The objective distance refers to the normalized Euclidean distance between the current parameter vector and the final parameter vector.

the graded optimisation process for one user. In Figure 5.9 and Figure 5.10, the normalized distance is obtained by considering the final result of the optimisation as the true optimal sample, and computing the distance between that sample and each other sample. As can be seen, Bayesian optimisation is well apt at finding a result that matches the observations.

The sole instance of a user preferring the manually optimised parameters over those obtained using Bayesian optimisation was obtained during the second phase of a preference approach. Due to the criteria chosen for the selection of the first sample of the next pair (i.e. take the previous best encountered), the system can be led to evaluate many pairs containing the same, 'best' sample. This can be seen in Figure 5.5, where a sample gets tested against many different samples. Combined with noisy measurements, this can lead to errors if a sample is erroneously preferred over one that has been itself preferred over a large population of samples.

Lastly, a survey was conducted with the participants after the study. While half of them preferred the process of the preference approach, none of the participants found that the results from the preference study performed better. This could be due to the fact that the preference approach only supports binary inputs for a given sample pair, thus limiting the user's capability to specify detailed gradients of preferences.

5.4 Discussion and Conclusion

This chapter presents a principled approach to the optimisation of human-robot interfaces. Often manually tuned by a single user, the performance of human-robot interfaces can vary significantly between users. This can be particularly detrimental when these interfaces impact directly the robot behaviour, potentially resulting in undesired motions. In this chapter, Bayesian optimisation is used to provide a meaningful way to perform user-specific tuning of the behaviour of the adaptive motion scaling scheme, previously presented in chapter 4. Two optimisation approaches are used: a graded approach in which users attribute a numerical value to a given system behaviour, and a preference approach in which users specify which system behaviour they prefer amongst those observed.

The results show that both optimisation approaches are able to provide at least similar system performance than the manual optimisation, while displaying more intuitive behaviour. In the case of graded optimisation, the system actually performed significantly better while also providing a more intuitive behaviour, and with only 12 samples needed. These results highlight the limitations of relying on manually optimised parameters when designing key aspects of human-robot interfaces. The fact that the manually optimised parameters, chosen to maximise performance, felt both less intuitive and showed worse performance than those obtained through a realistic optimisation procedure illustrates just how beneficial using optimisation approaches can be.

It is also interesting to note that the optimised parameters were typically significantly different from the manually chosen ones, highlighting the limitations of relying on intuition alone. In chapter 4 the parameters k_d, k_a, and k_s were all thought of as safety parameters intended to prevent the system from misbehaving, and so set to fairly strict values. To compensate, the parameter s_c dictating the maximum achievable scaling was set to be relatively high, reflecting the idea that if the conditions set by k_d, k_a, and k_s are met, a substantial boost to the scaling factor will be introduced. Contrarily, the result of the optimisation would typically lower the maximum achievable scaling s_c while relaxing the constraints dictated by k_d, k_a, and k_s. This resulted in a system more responsive to situations where the adaptive motion scaling might help, while substantially limiting any risk of overshoot. This result highlights the power of optimisation for finding new and

more optimal solutions than simply those available through intuition.

A limitation of the presented evaluation study is the small number of participants. Due to practical limitations regarding the availability of some of the system components, only six participants were able to be recruited for the study. A substantial amount of data was collected from each participant, allowing to draw meaningful conclusions with regards to the user-specific optimisation results. However, the small number of participants did not allow for the study of population behaviour. While most users seemed to gravitate towards parameters described in the previous paragraph, it is difficult to derive a meaningful analysis of whether different population groups would have emerged from the optimisation process.

This chapter also highlights the suitability of Gaussian processes for human-robot interface applications thanks to their flexibility, data-efficiency, and the fact that the structure of these models is formulated as high-level constraints through the kernels. In this chapter, the model presented in chapter 4 was used to control the motion scaling, and only its parameter values were modelled with Gaussian processes, used as surrogate models through the optimisation process. It would have been possible to forego the model presented in chapter 4, and instead model the motion scaling itself directly from the input values using Gaussian processes. This would have allowed more flexibility in the final model, perhaps yielding even better results. However, being able to provide a model structure, as in this application, allows the system to quickly reach efficient system behaviour with very few samples. In the next chapter, Gaussian processes are used to model directly desired system behaviour, and the necessary adjustments required are discussed.

6

Context-Aware Augmented Reality Display

6.1 Introduction

The previous chapters focused on the development of implicit robot control interfaces, leveraging gaze data to infer user intention and adapt system behaviour accordingly. This chapter makes use of, and expands on the same approach to tackle a key issue with augmented reality interfaces: when to display the augmented reality and how to control this display behaviour. Traditionally, augmented reality interfaces are controlled using an on/off switch, where the entire augmented reality overlay is either fully shown, or fully hidden. This display behaviour can be problematic depending on the type of surgical procedure being carried out, and has been demonstrated to cause inattentional blindness [139, 131]. An example of this is tumour resection in neurosurgery. Due to the critical importance of healthy brain tissue, it is imperative for the surgeon to observe the narrowest resection margins possible, while at the same time making sure to remove the tumour completely to avoid another traumatic procedure. Under these circumstances, accurate image guidance is invaluable. Augmented reality is an ideal candidate to provide this intra-operative image guidance, as it allows the surgeon to benefit from the

Content from this chapter was published and is reproduced with permission from:
Context-Aware Modelling for Augmented Reality Display Behaviour. Gras G., Yang G-Z. In Robotics and Automation Letters 2019.

image guidance while operating. This is in contrast to existing solutions in which the surgeon is required to stop operating to check the current progress using a tri-planar display of the registered pre-operative scan. However, due to the powerful magnification of surgical microscopes, the field of view is typically heavily narrowed during these operations, such that the tumour takes up a substantial portion of the screen. As such, a traditional augmented reality display would cover most of the field of view when activated. While different trade-offs can be made on the appearance of the augmented reality overlay, as discussed previously in section 2.4.3, an augmented reality overlay covering a substantial portion of the screen will necessarily hide some information from the surgeon. This can be particularly dangerous if events that would require immediate attention, such as uncontrolled bleeding, are not noticed right away.

This chapter exploits the gaze-contingent methods developed previously in chapters 3, 4, and 5 to infer the ideal augmented reality display behaviour based on the current context. By training a model making use of gaze data, instrument tracking, and tumour registration information, the proposed system is able to automatically select the augmented reality display mode that best matches the user's actions. This approach removes the need to manually switch the augmented reality on and off, thus making sure the augmented reality is always activated when image guidance is required, while not occluding the field of view otherwise.

Furthermore, the fact that the augmented reality display mode is automatically selected by the model opens up the possibility to introduce more specialised augmented reality display options. These specialised display modes can provide more advantageous visualisation options, but are only suitable in specific conditions. An example of this, used in this work, is a partial augmented reality display mode highlighting only a fraction of the tumour contour close to the cutting instruments. This display mode allows the user to visualise the tumour margins during cutting motions without obstructing most of the surgical view. Such specialised augmented reality displays would be unwieldy to control manually, as users would either need to toggle through multiple modes or use several buttons. Using a context-based model, however, additional display modes can be added without additional burden to the surgeon while providing valuable new visualisation options automatically.

Building on the work of the previous chapters, this chapter presents a data-

efficient multi Gaussian process model to control the augmented reality behaviour. The multi Gaussian process model acts as a multi-agent voter model where each agent votes for a specific augmented reality display mode. This model is capable of integrating meaningful priors for the different modes, able to specify default clinically-friendly behaviour even when substantially outside of the training set. Additionally, this work illustrates how uncertainty propagation in the model can be used to improve the training efficiency, helping the user identify areas of high model uncertainty during training. A detailed simulated robotic environment is presented, used to replicate conditions similar to a well-registered robotic tumour contour marking task. The results of a detailed simulated experiment for this task with 15 users demonstrate the usefulness the proposed approach, with clear reductions in inattentional blindness and task times.

6.2 Simulator and System Overview

6.2.1 Simulator

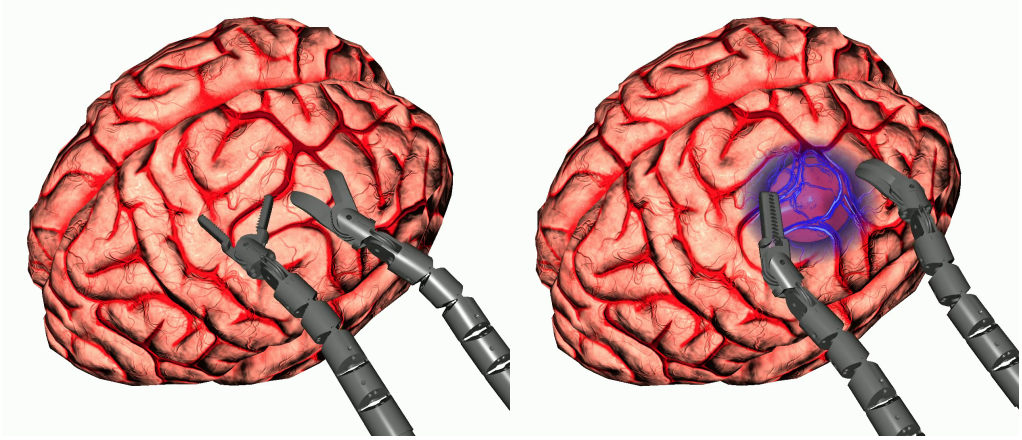
The aim of this work is to explore how different forms of augmented reality display behaviour impact the performance of surgical tasks. More specifically, this work aims to develop and study the advantages of context-based augmented reality display behaviour for surgical robotics. For this purpose, a system possessing the following is necessary:

1. Controllable robotic instruments, using accurate kinematic behaviour. This is important to replicate the same interaction limitations that would be present in a surgical procedure.

2. A dynamic environment, resulting from the interaction between the surgical instruments and a deformable tissue surface. This is a key requirement as a static environment would trivialize most image guidance-based tasks.

3. A segmented and registered tumour.

To meet the above requirements in an easily controllable and repeatable setting, the presented augmented reality display behaviour model is integrated and tested in a surgical simulator. The simulated environment consists in a pair of simulated



(a) (b)

Figure 6.1: Illustrations of the simulator (a) without augmented reality, (b) with augmented reality displaying a small tumour.

robotic instruments and a deformable brain model. The simulated robotic instruments are virtual replicas of the 7-DoF instruments presented in [160, 159], and used in the previous chapters 4 and 5. The same kinematic solver used to drive the real robot is used to solve the kinematics of the simulated instruments, such that their behaviour matches closely the behaviour of the real system. The only difference with the instruments presented in the previous chapters is that the end-effectors mounted on the simulated instruments are standard surgical grippers. In this work, the left instrument is equipped with a surgical sucker, and the right instrument is equipped with a scissor. Both of these are standard surgical instrument end-effectors that also exist in the robotic system.

The instruments can interact with the simulated brain model using deformable physics. In this work, the Bullet Physics engine is used to handle the dynamic collisions between the instruments and the deformable brain. Additionally, a virtual tumour is contained within the deformable brain. The tumour deforms with the brain, such that the simulated surgical scene stays realistic during deformations without having the risk of the tumour piercing through the deformable brain. Figure 6.1 shows the simulated environment. Note that the deformable model obtained using the Bullet Physics dynamic collisions provide no guarantee to accurately model the deformations of real brain tissue resulting from the forces exerted by the simulated surgical instruments. This, however, is not an issue for the

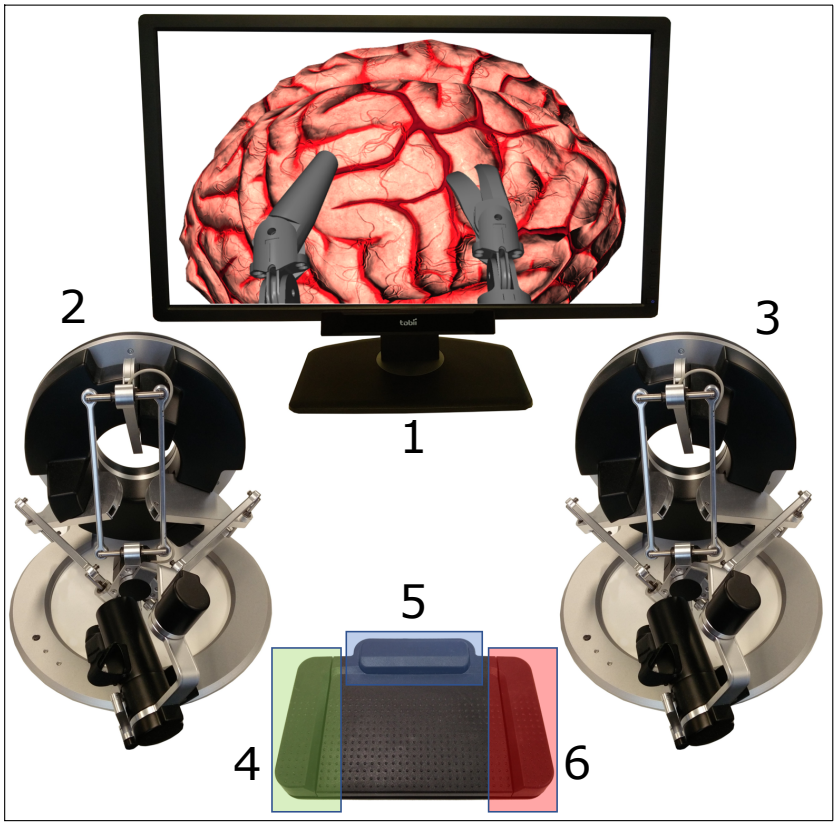


Figure 6.2: System overview.

purpose of this work. The deformation modelling is only used to simulate the difficulty in following precise moving tumour margins posed by deformable tissues during a real procedure. A lack of dynamic deformations would be unrealistic and could trivialize the simulated tasks, as the simulated scene would be perfectly (and unnaturally) static. This could allow for the memorization of the tumour margins, potentially skewing the results and bypassing challenges encountered in real systems. Note that this assumes the availability of intra-operative image guidance to update the tumour position. This could be intra-operative ultrasound, MRI, or vision-based modelling, for example.

6.2.2 System Overview

The simulated robotic system is actuated using two Omega.7 haptic devices (Force Dimension, Switzerland), similarly to the real robotic system. Likewise, a footpedal

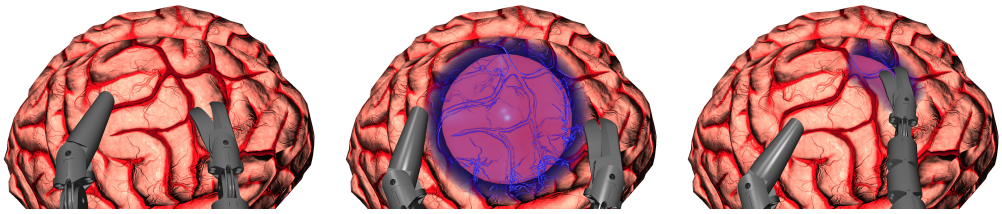
is used to activate and clutch the robotic system. Figure 6.2 illustrates the system components. The right footpedal 6.2(6) is used to activate/deactivate the robot, and the left footpedal 6.2(4) is used to clutch/unclutch the robot. The top footpedal 6.2(5) is used to toggle the augmented reality display on or off when in manual mode, as described in the user study section 6.4.2. Additionally, both Omega.7 haptic devices simulate a "gripper button" by exerting a small force opposing the gripping motion when the gripper is nearly closed. Overcoming this force registers the "gripper button" as pressed down. The user can activate the simulated sucker mounted on the left instrument using this feature by fully pressing the left Omega.7 gripper. Likewise, fully pressing the right Omega.7 gripper will activate the "cutting mode" of the scissor, used to mark the contour of the tumour. Lastly, a Tobii Pro X2-60 eye tracker (Tobii AB, Sweden) is used to detect the user's gaze with an update rate of 60 Hz.

6.2.3 Model Inputs and Outputs

The purpose of the presented system is to determine the desired augmented reality display mode based on the scene context. This effectively requires the creation a model of the augmented reality display behaviour, where the input consists of data from the surgical scene and the output is the augmented reality display mode. Three augmented reality display modes are chosen in this work:

1. No augmented reality.
2. Full augmented reality, where the entire tumour is shown.
3. Partial augmented reality, where only a fraction of the tumour is revealed with the augmented reality.

These modes are illustrated in Figure 6.3. The partial display mode is introduced to exploit the advantages of the context-based nature of the proposed method. As the augmented reality requires the tumour to be segmented and registered, the tumour contour and its projection on the surface tissue are also known. As such, during contour marking or cutting motions, it could be beneficial to only reveal the area surrounding the tumour edge with the augmented reality. The partial augmented reality display mode does exactly this. When enabled, it locates the point



(a) (b) (c)

Figure 6.3: Illustration of the three augmented reality display modes with the simulator view used in the tumour contour marking task. (a) No augmented reality. (b) Full augmented reality. (c) Partial augmented reality.

on the tumour contour closest to the cutting tool and highlights the tumour edge around that area. This mode allows the user to visualise the tumour edge where they are performing the cutting motion, while not obstructing the line of sight to other parts of the surgical workspace. The method used for the display of the augmented reality is an *inverse realism* method, based on [137]. An efficient low-latency pipeline for augmented reality stereo rendering was developed, as well as improvements regarding the augmented reality texture modulation. These are presented in detail in chapter 7, sections 7.3 and 7.4, where they play an important role for force visualisation and depth perception.

In this work, the model inputs consist of values describing the position of the surgical instruments, gaze fixation point, and relative tumour position. To limit the input dimensionality, these were processed into five values. These are illustrated in Figure 6.4 and described below:

- The distance between the projection of the tumour centre on the tissue surface (labelled as (1) in Figure 6.4) and the suction tool end-effector (labelled as (2) in Figure 6.4). This distance is labelled (a) in Figure 6.4.
- The distance between the projection of the tumour centre on the tissue surface (labelled as (1) in Figure 6.4) and the cutting tool end-effector (labelled as (4) in Figure 6.4). This distance is labelled (d) in Figure 6.4.
- The pixel distance between the 2D gaze fixation point (labelled as (3) in Figure 6.4) and the 2D projection in screen coordinates of the suction tool end-effector (labelled as (2) in Figure 6.4). This distance is labelled (b) in Figure 6.4.

- The pixel distance between the 2D gaze fixation point (labelled as (3) in Figure 6.4) and the 2D projection in screen coordinates of the cutting tool end-effector (labelled as (4) in Figure 6.4). This distance is labelled (c) in Figure 6.4.
- The distance between the cutting tool end-effector (labelled as (4) in Figure 6.4), and the point on the tumour contour closest to the cutting tool end-effector (labelled as (5) in Figure 6.4). This distance is labelled (e) in Figure 6.4.

The distances a, b, c, d and e from Figure 6.4 are the model inputs, grouped into the input vector $\boldsymbol{x}$ evaluated at every time step. This input representation was chosen as it contains the key variables necessary to specify the desired augmented reality display behaviour chosen in this work, as detailed in section 6.4.1. Alternatively it would be possible to use dimensionality reduction techniques as

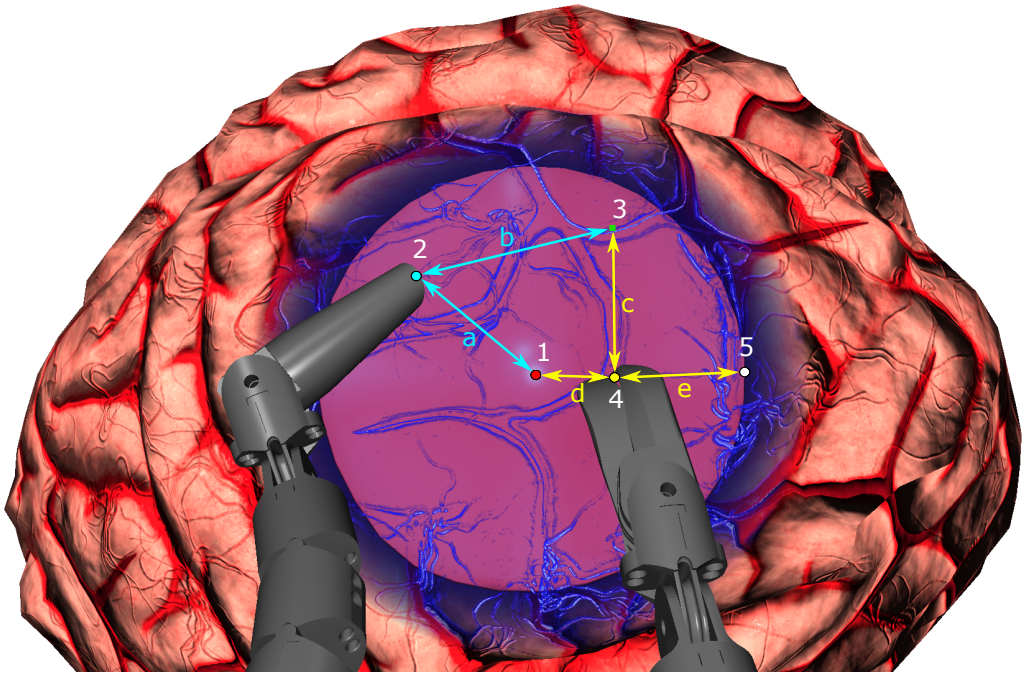


Figure 6.4: Definition of distances used as inputs in the augmented reality display behaviour model. 1 represents the projection of the tumour centre on the brain tissue surface. 2 represents the suction tool end-effector. 3 represents the 2D gaze point in screen coordinates. 4 represents the cutting tool end-effector. 5 represents the point on the tumour contour closest to the cutting tool end-effector.

part of the learning process of the model, so as to remove this design choice. However, this would likely increase the amount of data required for the resulting model to achieve a similar performance, thus increasing the training time of the model. Additional alternatives are discussed in section 6.5. Note that, as in the work presented in chapter 5, the values of the input vector $\boldsymbol{x}$ are normalized to a $[0, 1]$ range.

6.3 Multi Gaussian Process Modelling

6.3.1 Gaussian Process Voters with Meaningful Priors

In order to be able to select the appropriate augmented reality display mode depending on the scene context, a model of the augmented reality display behaviour is necessary. Using the model inputs and outputs defined previously in section 6.2.3, several important considerations can be noted to help with the choice of the model.

Firstly, the model must be highly data-efficient. The reason for this requirement is similar to the one in the approach presented in chapter 5. In that previous work, the desired robot behaviour was learned in a supervised manner through user input. In order to rapidly generate a user-specific model, it was key to use a highly data-efficient model since users were required to assess each robot behaviour individually during the training process. However, in that work the model structure was already chosen and only the parameters of the model were learned using Bayesian optimisation. This was because reasonably strong assumptions could be made intuitively to guide the design of the model, leaving key parameters as a way to tune the robot behaviour. Instead, the work proposed here aims to learn the model entirely, relying on as few assumptions as possible. This is in part motivated by the fact that it is harder to intuitively come up with a model structure for the augmented reality display behaviour. Unlike in chapter 4 where the adaptive motion scaling could intuitively be thought of as a combination of independent driving and safety factors, each of which could be modelled linearly, it is much more challenging to find a similar logic in this application. For example, it is clear the partial augmented reality display should be displayed during cutting motions, e.g. when the user is fixating the cutting tool while the cutting tool is

close to the tumour. But what if the suction tool is close by? What if the user is fixating something else? Most importantly, while it is possible (and necessary, to train the model) for a given user to come up with a specific desired display behaviour, another user might have completely different preferences.

The second consideration is that the resulting model should possess a safe default behaviour, even when operating substantially outside of the training set. This is particularly important considering the comparatively low target amount of training data relative to the dimensionality of the problem. In this work, the safe behaviour is defined as the "No augmented reality" display mode, as this is the default view used in the operating theatre. Finally, the model should ideally be able to cope with a training set containing substantially varying quantities of data for the different labels (i.e. augmented reality display modes). This consideration has similarities with the desire to enforce a safe default behaviour without requiring the user to parse a substantial portion of the feature space.

Gaussian processes exhibit many attractive properties relative to the above considerations. As detailed previously in chapter 5, covariance functions provide an intuitive way of specifying high-level beliefs about the underlying functions, allowing for highly flexible models. Additionally, the analytically tractable nature of the posterior predictive distribution in Gaussian process regression makes it a very practical model. The work presented in chapter 5 demonstrated the highly data-efficient nature of Gaussian processes, making them useful priors for Bayesian optimization. Additionally, mean priors are an effective way to specify default behaviour outside of the training set.

The proposed model consists of three Gaussian processes that act as voting agents. Each Gaussian process agent encodes an unknown objective function related to a specific augmented reality display mode, representing the suitability of that mode to be the correct display mode for the system. For any given input sample $\boldsymbol{x}$, each agent votes individually using the predictive posterior distribution for their corresponding Gaussian process, and the display mode of the agent with the winning vote is applied. When a new input sample is labelled by the user, each of the three Gaussian processes is updated with a new observation. The Gaussian process for the matching mode is attributed an observation of value $+1$, and the other Gaussian processes are attributed an observation with the value 0. The resulting architecture is illustrated in Figure 6.5.

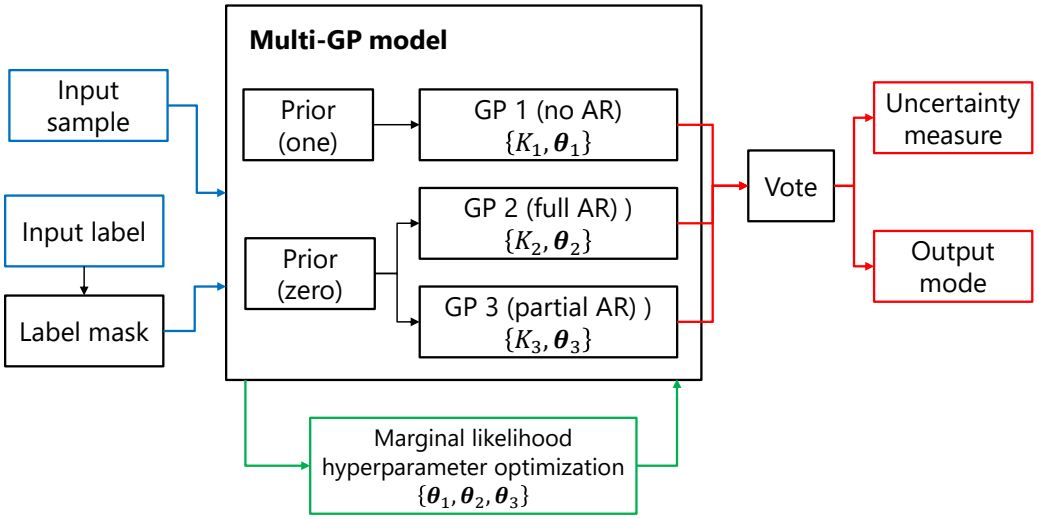


Figure 6.5: Multi Gaussian process model overview.

Each Gaussian process agent has its own kernel and mean prior. To enforce the "No augmented reality" default behaviour, the Gaussian process corresponding to that mode is attributed a constant mean prior of value one, while the other Gaussian processes are attributed a mean prior value of zero. Note that in this work a Gaussian noise likelihood was used to reflect the continuous voter-agent design, as opposed to a probit classification model, for example, such as the one used in [173]. Finally, automatic relevance determination (ARD) squared exponential kernels were used for each of the Gaussian processes. This results in a total of seven hyperparameters for each Gaussian process: one for each of the five input dimensions as part of the ARD kernel, the kernel signal variance hyperparameter, and the noise variance hyperparameter. The hyperparameters for each of the Gaussian processes are independent.

Figure 6.6 illustrates the behaviour of the three Gaussian process voter agents for a one dimensional example model. The example unknown ground truth function is illustrated in Figure 6.7(top), where values of 0 correspond to choosing GP 1, values of 1 correspond to choosing GP 2, and values of 2 correspond to choosing GP 3. The behaviour of each of the individual Gaussian processes shown in Figure 6.6 is simply that of a standard Gaussian process regression. For any given observation, the Gaussian process of the corresponding mode is attributed an observation value of $+1$, and 0 for the other Gaussian processes. Note that the mean

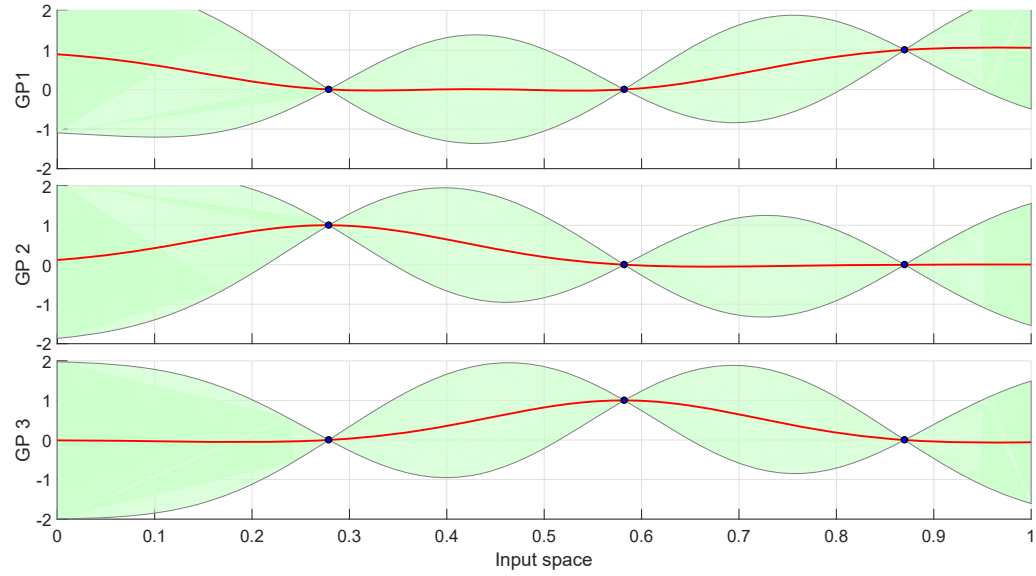


Figure 6.6: Gaussian process voter agents for three observations. No hyperparameter optimization.

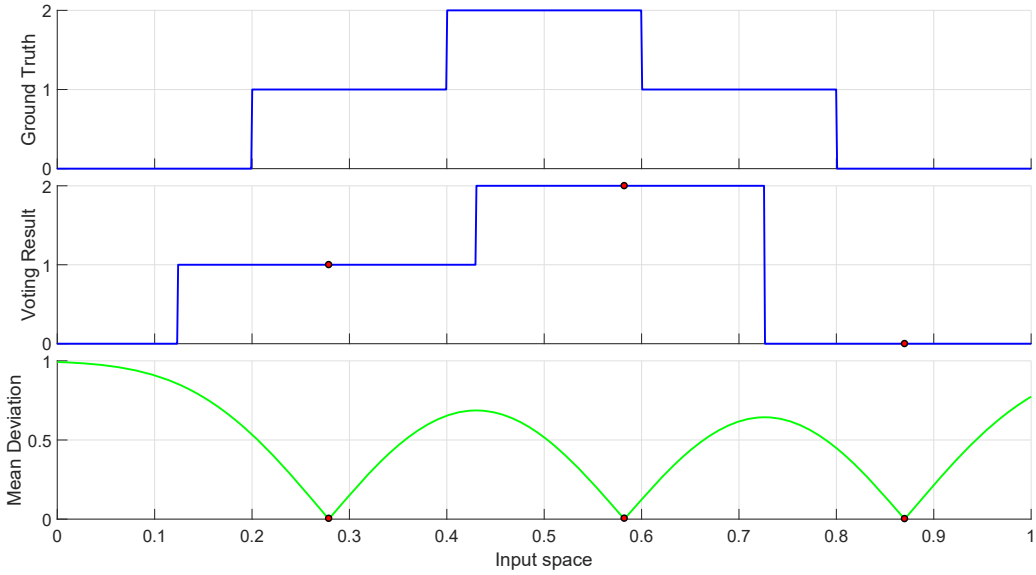


Figure 6.7: Ground truth and result of the voting process for the Gaussian process voter agents shown in Figure 6.6.

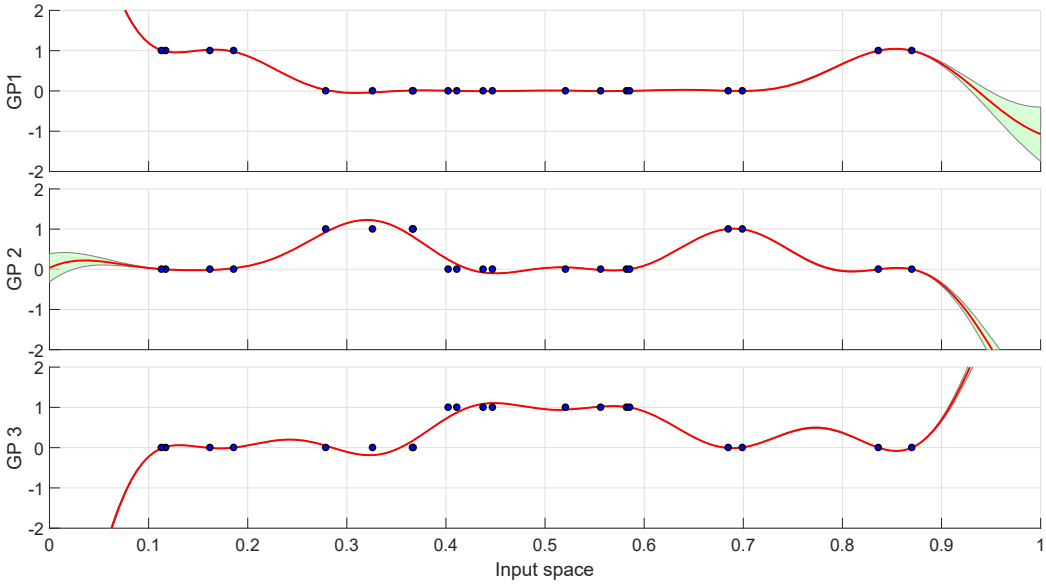


Figure 6.8: Gaussian process voter agents for twenty observations. No hyperparameter optimization.

priors naturally bring back the Gaussian processes to a default value outside of the observations. The result of the combined voting process is displayed in Figure 6.7(middle). This result describes the model's approximation of the ground truth function displayed in 6.7(top), based on the observations collected. Finally, Figure 6.7(bottom) shows the mean deviation of the three Gaussian processes.

While at first glance the example model illustrated in Figures 6.6 and 6.7 seems to perform well, there are some important limitations to point out. Figures 6.8 and 6.9 illustrate the results of the same model for 20 observations. The first observation that can be made from Figure 6.8 is that the model does not seem to match the observations really well. This is particularly visible in Figure 6.8(middle: GP 2). This would be understandable if the noise parameter of the model was high, however in this figure all the model parameters are identical to those used in Figure 6.6. Additionally, if this was the case the uncertainty measure would be higher to reflect the added stochastic variation coming from the noise. On the contrary, the model displays an extremely low uncertainty nearly everywhere. Lastly, even the mean prior does not seem to be respected. This is due to a saturation of the model, as the fixed hyperparameters chosen do not allow the model to accurately represent the observed data. Under the assumptions imposed

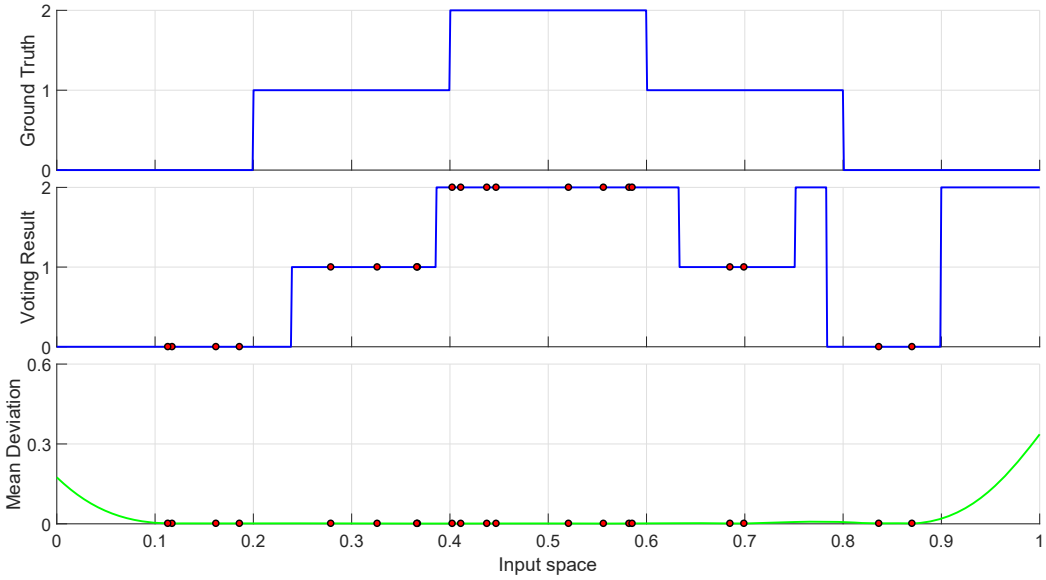


Figure 6.9: Ground truth and result of the voting process for the Gaussian process voter agents shown in Figure 6.8.

on it by the hyperparameters, the model effectively "runs out" of better options to describe the data, concluding with high certainty that it has an accurate representation of the underlying function. This saturation also causes the model to fluctuate strongly outside of the training data before returning to the mean prior. This is observable in Figure 6.10, in which a zoomed out view of Figure 6.8 is displayed. The mean prior is eventually respected, but the saturation of the model initially causes undesired behaviour.

6.3.2 Hyperparameter Optimisation

To address the issues observed in section 6.3.1, the hyperparameters of the model must be selected based on the observed data. Fortunately Gaussian processes exhibit some valuable properties related to hyperparameter optimization. As mentioned in chapter 5, a common way to select the hyperparameters of a model is to maximise the marginal likelihood of the model $p(\boldsymbol{y} \mid X, \boldsymbol{\theta})$. The marginal likelihood is defined as

$$p(\boldsymbol{y} \mid X, \boldsymbol{\theta}) = \int p(\boldsymbol{y} \mid \boldsymbol{f}, X, \boldsymbol{\theta}) p(\boldsymbol{f} \mid X, \boldsymbol{\theta}) d\boldsymbol{f}. \tag{6.1}$$

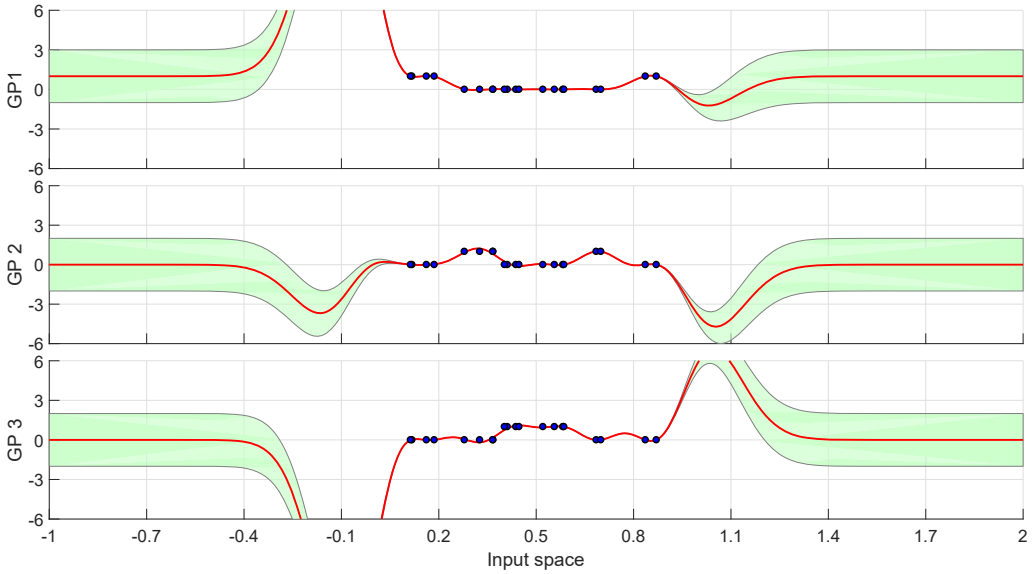


Figure 6.10: Zoomed out view of the Gaussian process voter agents for twenty observations. No hyperparameter optimization.

Note that equation 6.1 represents a slight abuse of notation as $p(\boldsymbol{y} \mid \boldsymbol{f}, X, \boldsymbol{\theta})$ and $p(\boldsymbol{f} \mid X, \boldsymbol{\theta})$ may only depend on a subset of the hyperparameters $\boldsymbol{\theta}$ depending how they are defined. Under the Gaussian noise likelihood, the results presented in section 5.2.2 can be used to obtain the expression of the log marginal likelihood:

$$\log p(\boldsymbol{y} \mid X, \boldsymbol{\theta}) = -\frac{1}{2}(\boldsymbol{y}-\boldsymbol{m})^T(K+\sigma_n^2 I)^{-1}(\boldsymbol{y}-\boldsymbol{m}) - \frac{1}{2}\left|K+\sigma_n^2 I\right| - \frac{n}{2}\log(2\pi), \tag{6.2}$$

where the covariance matrix K the mean prior $\boldsymbol{m}$, and noise variance σ_n can all vary with $\boldsymbol{\theta}$ depending on the definition of the hyperparameters. This tractable expression of the marginal likelihood is highly advantageous, as it can be evaluated for different values of the hyperparameters $\boldsymbol{\theta}$. The value of $\boldsymbol{\theta}$ maximising equation 6.2 also maximises the likelihood of the model, thus providing a method to determine the optimal hyperparameters of the model. At this stage different global optimization techniques can be used to determine the optimal value of the hyperparameters. Additionally, equation 6.2 is differentiable with regards to the hyperparameters $\boldsymbol{\theta}$ so long as the kernel K is also differentiable with regards to $\boldsymbol{\theta}$. As this is the case with kernels of the squared exponential family used in this work, it becomes possible to use gradient descent optimizers to find the optimal value of

the hyperparameters. In the presented work, five different optimizers were used in a parallel thread pool to solve this optimization problem. Three derivative-free algorithms: DIRECT (Dividing Rectangles) [176], DIRECT-L (locally biased version of DIRECT) [177], and BOYQA [178]; and two gradient-based algorithms: LBFGS (Low-storage BFGS) [179] and Conjugate gradients [180]. The conjugate gradients algorithm was re-implemented from [166], and the other algorithms were used through the NLopt library [181]. Multiple randomized starts were used to avoid local minima.

The capability of Gaussian processes to automatically derive the hyperparameters from the observed data is highly advantageous. The standard approach to compare the performance of parameter-based models consists in using cross-validation across different parameter sets. This method can be highly computationally expensive and time consuming for large data sets and/or models with a high number of parameters, as the model must be re-trained completely to allow comparison for each parameter set. Using Gaussian processes the hyperparameters can be optimized directly by maximising equation 6.2.

Optimization of the hyperparameters allows for far greater flexibility in the Gaussian process models. As mentioned in chapter 5, varying the lengthscale of the squared exponential kernels can significantly change the form of function drawn from the corresponding Gaussian process. When applied to ARD kernels, possessing individual lengthscales for each input dimension as described in equation 5.13, this indeed allows performing automatic relevance determination by effectively weighing the effect of variations along each input dimension. Finally, being able to estimate the noise variance is also highly advantageous, particularly so in this application where the exact boundaries between the different augmented reality display modes may not perfectly consistent during training.

Figures 6.11 and 6.12 show the Gaussian process models resulting from hyperparameter optimization using the same observed data as in Figures 6.8 and 6.9. Notice in Figure 6.11 that the data seems to be much more accurately described by the model, resulting in part from the shorter lengthscales found by the optimization process. The combined voting result showed in Figure 6.12(middle) is also much closer to the ground truth shown in Figure 6.12(top). The key remaining inaccuracies in the voting model lie around the boundaries between the different modes, as data is sparse around those boundaries. However, note that the

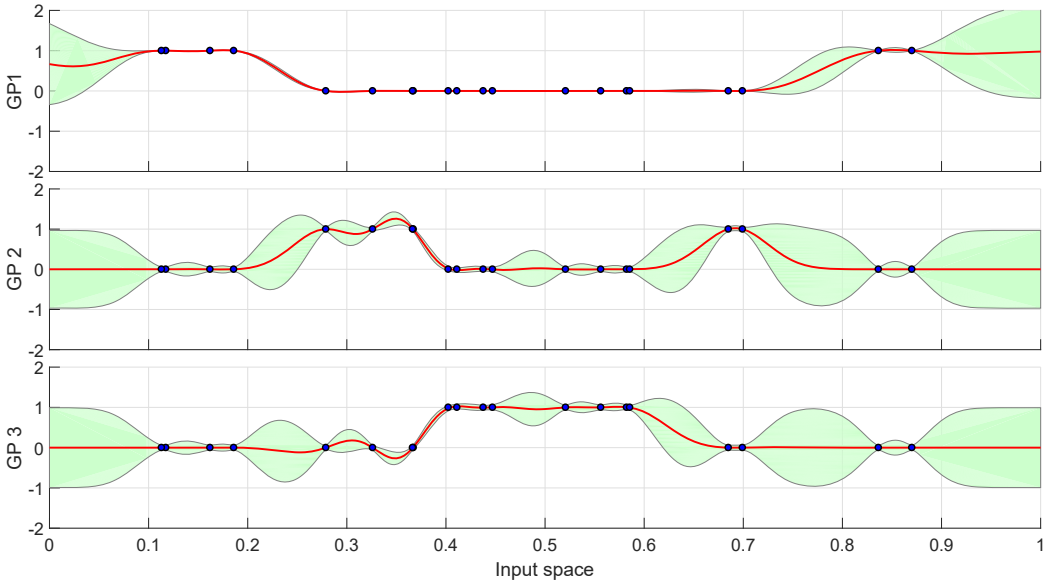


Figure 6.11: Gaussian process voter agents for twenty observations, with hyperparameter optimization.

uncertainty of the model accurately reflects this lack of data. For example, Figure 6.12(bottom) shows a peak in the mean standard deviation around $x = 0.65$, reflecting the model's low certainty as to the exact placement of the boundary between the two modes of the ground truth function.

Finally, while the hyperparameter optimization described above offers clear advantages, it can also present drawbacks in some circumstances. The most notable limitation is tied to the quantity of observations. If not enough observed data is available, the presented hyperparameter optimization approach can lead to overfitting of the hyperparameters [165, 182, 183]. This is because if the hyperparameter dimensionality is too high relative to the training data, maximising the marginal likelihood may result in extreme hyperparameter values able to describe "well" all the observations. For example, this could result in models with really high lengthscale values and high noise variance, effectively fitting the observation to a line with a high estimated noise. Conversely, the data could also be described with a very low noise model with extremely small kernel lengthscales, where the resulting model is almost equal to the mean prior everywhere except at the observed data points. As such, in applications with very few observations, it is best to resort to other methods for the estimation of the hyperparameters of the system.

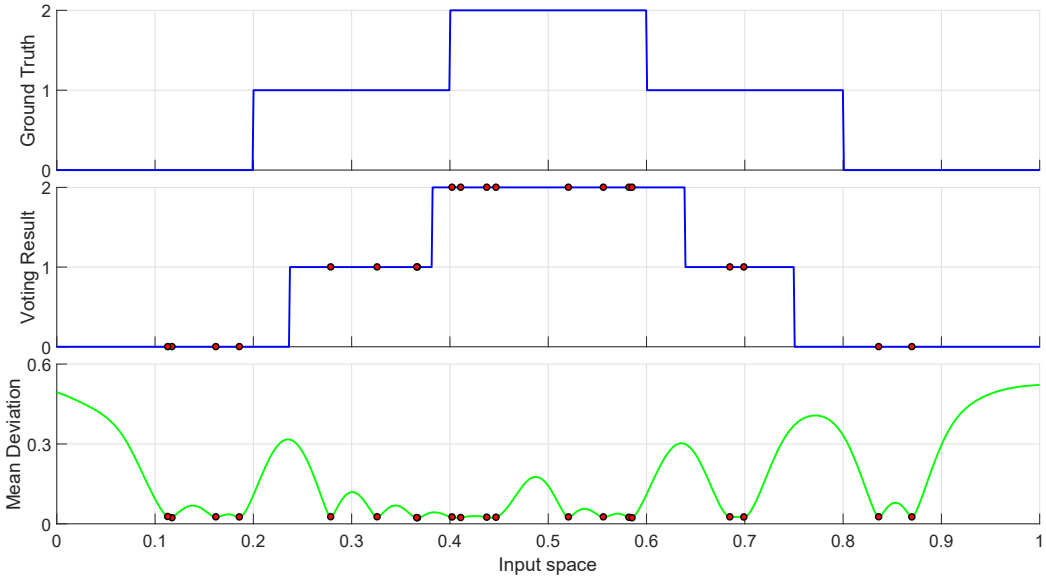


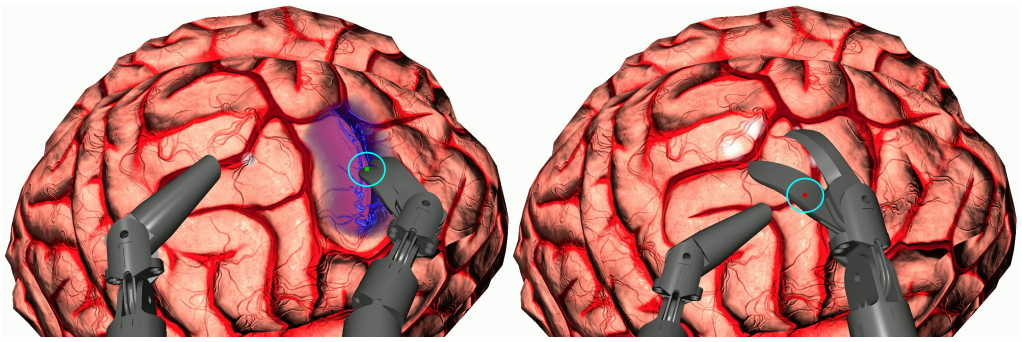
Figure 6.12: Ground truth and result of the voting process for the Gaussian process voter agents shown in Figure 6.11.

This was the case in the work presented in chapter 5, where user knowledge from the system behaviour could be used to set reasonable values for the hyperparameters without risking to saturate the model. In this chapter, however, the acquisition of the training samples does not require the robot behaviour to be tested through a task. The training consists in labelling directly the desired augmented reality display mode, as described further in section 6.4.1. This makes it possible for a much quicker acquisition of training samples, allowing for the optimization of the hyperparameters using the marginal likelihood.

6.4 User Study

6.4.1 Model Training

The model presented in section 6.3 is trained in a supervised manner. The user performing the training operates the system, and can request a specific augmented reality display mode at any given moment. As all the model inputs are evaluated individually at every time step, the system can be paused while the requested augmented reality display mode is used as a label for the current system inputs. The



(a) (b)

Figure 6.13: Visualisation of the uncertainty during model training. The uncertainty is visualised through the colour of the gaze point, located at the centre of the light blue circle. The light blue circles are displayed purely to draw the reader's attention to the correct points in the image. (a) Low uncertainty: the model has been trained to recognize the context of the cutting motion. (b) High uncertainty resulting from an unusual scene configuration.

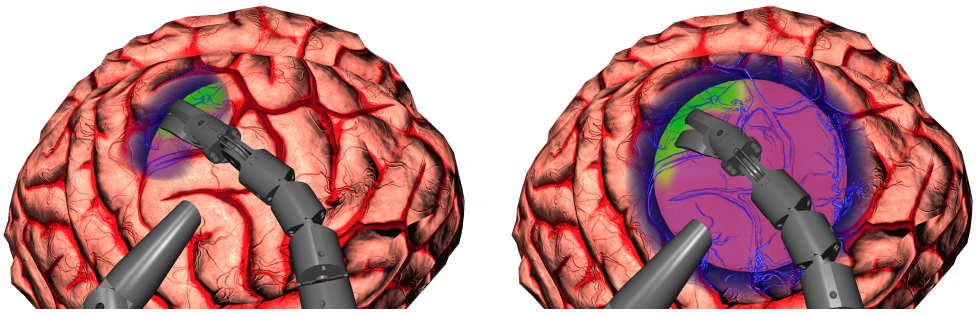
multi Gaussian process model learns incrementally, meaning the label is instantly taken into account and used from that point onwards. Optimization of the hyperparameters can be performed at arbitrary points during the training procedure, but does not need to be performed after every new label. Note that the result of a hyperparameter optimization does not affect the result of the next one, as multiple randomized starts were used during the optimization process. The main purpose of the optimization during training is to evaluate the current performance of the system, and potentially give more meaningful uncertainty measures.

A valuable property of the multi Gaussian process model employed is the ability to estimate the uncertainty of the model at any give time step. This uncertainty can be relayed to the user during training, assisting them to improve the data acquisition. By enabling the user to visualise where the model is more uncertain, the user is able to seek out areas of high uncertainty and provide labels to these areas. This allows the user to quickly improve the predictions generated by the model, by providing more meaningful data and avoiding duplicate labelling. In this work, the visualisation of the uncertainty is performed by marking the user's gaze point on the screen, and modulating the colour of that point from red to green depending on whether the model's uncertainty is high or low respectively for the current system configuration. This is illustrated in Figure 6.13.

The model was trained by an expert user, to enforce a particular desired behaviour. While it was initially considered to have every user train the model themselves, as was done in 5.3, this idea was discarded for the following reason. Because the model presented in this work requires the user to specify which mode they prefer in different contexts, it is important for the user training the model to have a consistent idea of how they want the model to behave. It is difficult for the model to generate meaningful results when trained by a user who does not have a desired system behaviour in mind. Note that this does not mean the trainer is required to know precisely the boundaries of each of the modes; the whole purpose of the model is to be able to generate that behaviour automatically from the observed data, and is able to cope noise. This requirement simply expresses the need for limited labelling noise and is not inherent to this model: a neural network trained to differentiate between cats and dogs with a substantial amount of wrong labels would also perform poorly. Since the user study was carried out with novices experienced with using a surgical robot, but not with an image guidance system, this would have required each user to spend time discovering what their own preferred augmented reality display behaviour was, drastically increasing the time necessary for the whole study. An alternative option would have been to guide the users during training towards a pre-selected behaviour. However this solution is very similar to running the study with a pre-trained model, and does not offer meaningful advantages. A total of 89 labels were collected to train the model used in the user study, and the hyperparameters were optimized as described in 6.3.2. The number of labels was purposefully limited to simulate the behaviour obtainable after a 15-20 minute training session.

6.4.2 Experimental Task and Protocol

A neurosurgical tumour contour marking task was designed to test the proposed system. The primary aim of the task is to mark the contour of the tumour as accurately as possible, with the secondary objective of performing the task as quickly as possible. Tumour contour marking is an essential step in the tumour resection process, as it is when surgeons decides the margins they will use during the resection. The importance of having positive margins as thin as possible combined with the delicate nature of the operation make it a prime application for augmented re-



(a) (b)

Figure 6.14: Visualisation of the marking of the tumour contour (a) for the partial augmented reality display mode, (b) for the full augmented reality display mode.

ality based image guidance. In neurosurgery the marking of the tumour contour can be performed with bipolar diathermy or with surgical scissors. In this work the simulated right instrument is equipped with a scissor end effector. This allows the user to simulate cutting motions by fully pressing the right Omega.7 gripper as described in section 6.2.2. The cutting action is visually represented by progressively changing the colour of the tumour model from its default pink colour to green around the cutting location. This is illustrated in Figure 6.14. The cutting in any given area is considered complete when the colour of the tumour in that area is purely green. This requires users need to maintain the cutting motion for a small amount of time in any given area, preventing them from unrealistically quickly completing the tumour marking. The task is complete when the entire contour of the tumour is purely green.

While the task above replicates the key functionality required to simulate a contour marking task, it does not capture the risks that arise from the augmented reality potentially hiding crucial information from the surgeon. Without simulating changes in the tissue texture that arise from potentially critical events such as bleeding around the operating site, the simulated task would be unrealistically static and could trivially be completed by simply keeping the "full augmented reality" view on all the time. To simulate this additional difficulty, the brain texture is modulated to make spots appear at random time intervals. These spots can either be red or white, and pulsate over time. They are illustrated in Figure 6.15.

The red spots represent bleeding and must be removed as soon as possible by the user. The white spots represent a benign variation in the brain tissue surface,

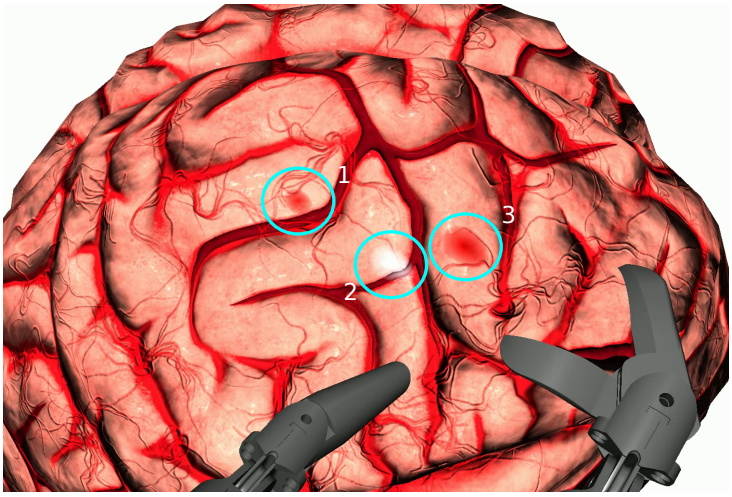


Figure 6.15: Visualisation of bleeding spots (in red, labelled 1 and 3 in the figure), and benign spots (in white, labelled 2 in the figure). The light blue circles are displayed purely to draw the reader's attention to the correct points in the image. The spots pulsate according to their own timers, growing larger and shorter periodically. Their shapes differ slightly but they are all roughly circular.

such as from water pooling up after irrigation of the surgical scene, and can be ignored. The spots can be removed using the sucker mounted on the left instrument. The sucker is activated by fully pressing the left Omega.7 gripper, as discussed in section 6.2.2. By pressing the sucker against the surface of the simulated brain where the spot is and activating it, the spot will be removed. Note that spots do not disappear on their own, and will keep pulsating until the experiment is over or until they are removed with the sucker. In the user study, spots appeared randomly every six to nine seconds. The type of the spot (bleeding or benign) was generated using a Bernoulli distribution starting with an equal probability ($p = 0.5$) for bleeding and benign spots. To limit the effect of randomness, successive generations of the same type of spot incrementally lowered the probability of drawing that type of spot by 0.15, making it impossible to generate the same type of spot more than four times. Upon generation of the opposite type of spot the probability of drawing either was reset to 0.5.

By order of importance, the task metrics, presented as such to the users, are thus as follows:

1. Average bleeding time: the average lifetime of a bleeding spot.

2. Contour marking error distances: the average distance from the scissor end-effector to the tumour contour during cutting motions.

3. The task completion time.

Two variations of the task were performed. In the first one users were able to control the display of the augmented reality manually using a footpedal. Pressing the top footpedal a first time, illustrated in Figure 6.2(5), would activate the full augmented reality display. Pressing it again would hide the augmented reality. This manually controlled on/off approach is currently the most widespread approach to control the display of the augmented reality in the operating theatre. The second variation of the task uses the proposed gaze-assisted multi Gaussian process model to control the display of the augmented reality, also referred to as the smartAR mode below.

The study was conducted with 15 participants, 13 male, 2 female. Participants were novices without significant prior experience using a surgical augmented reality image guidance system. Participants were first trained with the simulator without any augmented reality, to familiarize themselves with the control of the simulated robot. They then trained with the manually controlled augmented reality. The eye tracker would then be calibrated for that participant, and they would train with the gaze-assisted augmented reality. Finally, the users were given any free time necessary to be sufficiently proficient with doing both variations of the task. This would typically involve repeating the task once again for both variations. During this process, all participants received the same helpful advice for how to best perform the task. This included advice on how to best control the robot and best practice for marking the tumour contour. All participants were required to start marking the tumour contour from the same position, and to mark the tumour contour following the same direction. The position of the simulated instruments was reset before every task, as well the user's hand positions on the Omega.7 master devices. Finally, every participant performed each version of the task three times in a randomized order.

6.4.3 Results

The experimental results for the task metrics are reported in Figure 6.16. Numerical values for the mean, median, and statistical significance of the task metrics are

	Mean		Median		p-value
	Manual	SmartAR	Manual	SmartAR	
Bleeding time (s)	11.05	5.74	9.78	5.43	8.32×10^{-9}
Total task time (s)	108.1	93.99	94.93	80.73	8.42×10^{-5}
Position error (cm)	2.81	2.73	2.76	2.70	0.33

Table 6.1: Mean, median, and statistical significance results from the contour marking study.

also listed in Table 6.1.

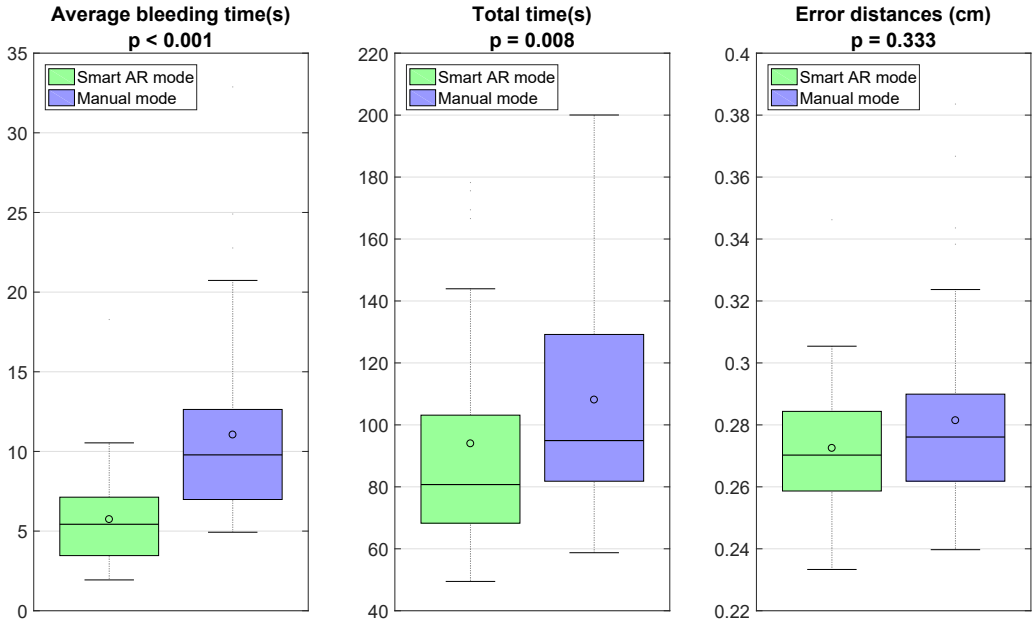


Figure 6.16: Results of the user study.

As can be seen in Figure 6.16, both the average bleeding time and the task completion time show statistically significant mean and median improvements for the gaze-assisted augmented reality relative to the manual mode. The average bleeding time shows a median reduction of 44.5% in favour of the gaze-assisted augmented reality. Likewise, the task completion time shows a median reduction of 10.7%. The contour marking error distances only show non-statistically significant negligible differences between the smartAR and manual modes.

The substantial reduction in bleeding time with the gaze-assisted augmented reality can be in large part attributed to the difficulty in identifying bleeding when the "full augmented reality" mode is activated. While the spots themselves are easily noticeable, it is impossible to see whether they are red or white through the augmented reality. In the manual mode, this forces the user to periodically disable the augmented reality to be able to check for bleeding in the areas where change was detected. The partial augmented reality display mode, however, leaves most of the field of view free from any augmented reality overlay, only providing guidance around the area where the user is cutting. The main limitation of this display mode is that it is only suited for cutting motions. Outside of cutting motions, for example if the scissor tool is distant from the tissue surface, the partial augmented reality mode risks flickering quickly back and forth as the closest contour point to the scissor end-effector can vary rapidly. Additionally, if a spot appears close to the scissor during cutting motions, the partial mode risks causing the same problem as when the full augmented reality mode is used, by hiding the colour of the spot. The multi Gaussian process model alleviates all these issues by automatically switching between the different modes to make sure that the most appropriate mode is always used. During the training of the model these different cases where encountered and identified, allowing the user to select the most suitable mode depending on the context. The improvement in the task completion time is a consequence of this more intuitive interface, removing the need to manually switch the augmented reality display mode.

The lack of significant improvement in the contour marking error distances is not surprising, as the multi Gaussian process model was not expected to bring any improvements in that regard. However, this result is still meaningful as it shows that a partial view of the augmented reality can be used without reducing the accuracy of the image guidance. It should be noted that in an original prototype of the simulator, the partial augmented reality view was much thinner. Although a full study was not conducted at the time to assess the performance of that prototype, early tests seemed to indicate a small loss of accuracy, implying that the partial view can only be made so small before a loss of accuracy is incurred.

6.5 Discussion and Conclusion

In this chapter, a novel augmented reality display control scheme is introduced. Using context from the surgical scene and the user's gaze, a multi Gaussian process model is trained to infer the ideal augmented reality display mode at any given time step. This allows for the use of specialised augmented reality views, such as a partial augmented reality display around the tumour contour cutting motions. This approach tackles a long-lasting issue in augmented reality applications, namely the need to toggle the augmented reality on or off manually to avoid obstructing the field of view at important moments. A detailed user study was performed to evaluate the performance of the system.

Results from the user study show clear improvements in user perception of the surgical scene and task times during a tumour marking task. The introduction of the partial augmented reality display mode plays a significant role in lowering the average simulated bleeding time to nearly half of what is achieved in the manual mode. The capacity of the model to switch the augmented reality display automatically enables this additional mode to be used without concerns for erratic augmented reality behaviour. Additionally, the automatic turning off of the augmented reality when the surgical sucker is used allows further improvements in task completion times. Building on the work of the previous chapter, this chapter further highlights the strong gains that can be made by developing surgical interfaces that take advantage of the scene context, and in particular how eye tracking data can be leveraged to great effect as a component of these interfaces.

While the model used in the presented approach here performed well, there are a number of ways in which it could be augmented in future work. A main limitation of the presented model is that it does not take into consideration dynamic effects. This makes it impossible to consider whole gestures as part of the context. These could be highly advantageous to use, particularly to switch the augmented reality display mode in cases where the context would otherwise not indicate such a switch is necessary. Indeed, a limitation of the current model is that a given scene context is described entirely by the input values defined in 6.2.3, which are all distance measures. This formulation increases the possibility of having two highly different contexts generate the same input vector. This problem can be resolved by increasing the input dimensionality, or enhancing the model to account

for dynamic effects. Finally, other kernels could be used to improve the model behaviour, in particular non-stationary kernels to better handle the sudden variations between modes.

7

Visual Force-Feedback

7.1 Introduction

The robotic interfaces explored so far in this thesis have all been designed for master-slave systems. While the master-slave paradigm presents numerous advantages in terms of precision and control capabilities for highly dexterous instruments, traditional master-slave systems are expensive and have a large footprint in the operating theatre. As illustrated in Figure 7.1, recent work has focused on miniaturising master-slave systems and developing new paradigms for low-footprint systems. In particular, ungrounded hand-held robots allow the surgeon to benefit from robotic enhancements while operating similarly to how they would with conventional instruments [184]. These instruments represent simple, low cost, yet intrinsically robust alternatives to conventional master-slave and co-operative control platforms.

These hand-held robots present a completely different interfacing paradigm than master-slave systems. For example, technological advances in hand-held instruments have allowed the integration of force sensing capabilities into surgical tools, resulting in the possibility of force feedback during an operation [185, 186, 187]. Other hand-held robotic probes have incorporated flexible tip-

Content from this chapter was published as:
Visual Force Feedback for Hand-Held Microsurgical Instruments. Gras G., Marcus H.J., Payne C.J., Pratt P., Yang G-Z. In Medical image computing and computer-assisted intervention 2015. Reproduced with permission from Springer.

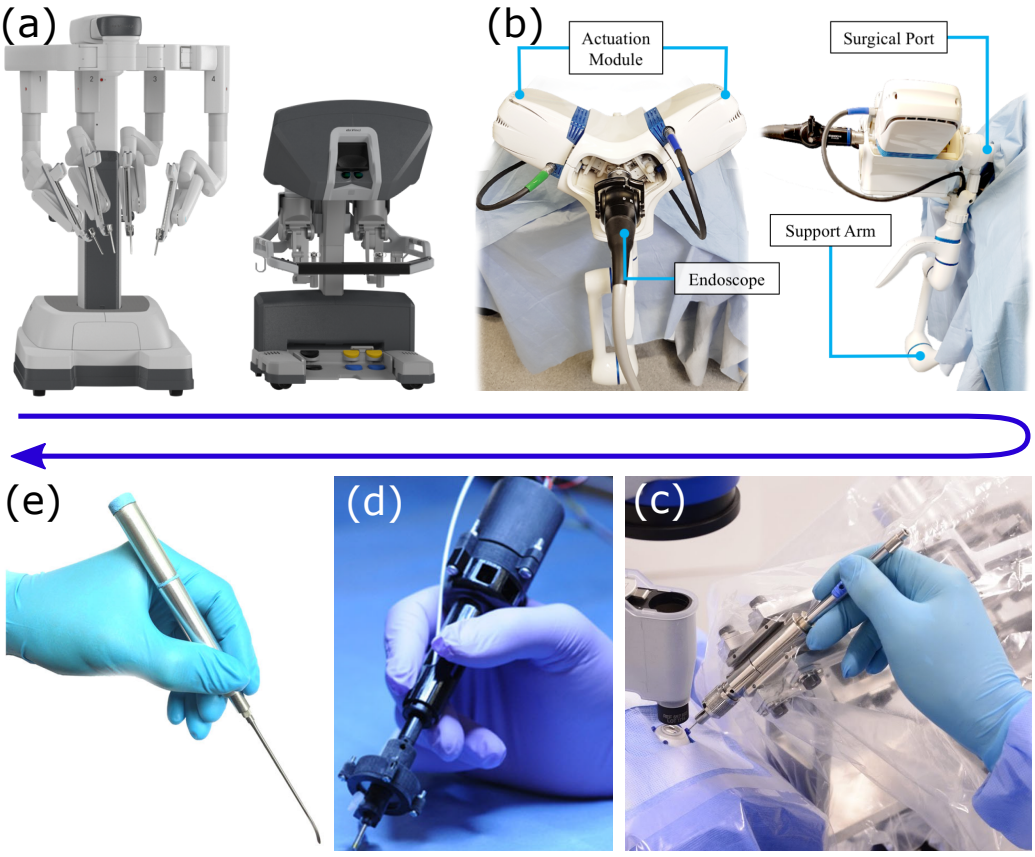


Figure 7.1: From large to small: surgical robots at different scales. (a) Da Vinci XI® and surgical console, ©Intuitive Surgical (b) Micro-IGES single port system, ©2017 IEEE [160] (c) EurEyeCase hands-on robot for eye surgery [189] (d) Hand-held instrument for confocal laser endomicroscopy, ©2011 IEEE [190] (e) Hand-held haptic device for neurosurgery, reproduced with permission from Springer [187].

mounted cameras, allowing free-hand exploration of hard-to-reach anatomical cavities [188].

The work presented in this chapter was motivated by the need for improved force feedback and image guidance interfaces with hand-held instruments, particularly regarding aneurysm clipping and tumour resection in microsurgery. Microsurgery is technically challenging, demanding both rigorous precision under the operating microscope and great care when handling tissue. Applying excessive force can result in irreversible tissue injury, but sufficient force must be exerted to carry out manoeuvres in an efficient manner.

Various modes of force-feedback have been used in both hand-held and robotic devices including tactile, auditory, and visual cues [184, 191]. Previous work has suggested that all of these forms of feedback have a comparable influence on surgical performance [192]. However, these modes differ considerably in the nature in which they may be integrated into the operative workflow. Tactile feedback may represent the most intuitive method to indicate when a force threshold is breached, but is best suited to conveying binary rather than graduated information. Auditory feedback does allow for more granular feedback, but may not be recognised above the ambient noise of a busy operating room. Visual feedback can provide detailed information in an unambiguous fashion, but similarly to augmented reality it risks inattention blindness - cues located away from the operating site may distract the surgeon, and cues overlying the operating site may obstruct the surgical scene.

This chapter presents a novel visual force-feedback approach for hand-held microsurgical instruments that allows graduated force feedback to be presented over the operating site while simultaneously providing image guidance. The proposed technique monitors forces exerted by surgical instrument on the brain tissue. When excessive forces are detected, an augmented reality overlay modulated by the detected forces is gradually shown, revealing the underlying anatomical structures. Unlike in chapter 6, no model needs to be trained to control the augmented reality display behaviour. The usage of the force information allows the augmented reality to appear when needed, i.e. when the surgeon is exerting forces at risk of damaging critical anatomical structures underneath the instrument. An *inverse realism* Non-Photorealistic Rendering (NPR) method [137] is used to preserve salient anatomical features and improve depth perception. The real-time pipeline developed for the efficient stereo augmented reality rendering is presented. Additionally, the force information is further exploited to displace the virtual elements using finite element modelling, generating motion cues to improve depth perception. The resulting technique is a novel context-aware augmented reality scheme where the augmented reality display is shown to the surgeon when relevant, and adapts based on the forces applied. The efficacy of the method is assessed by detailed user studies involving a retraction and dissection study to quantify the forces exerted with and without the visual force-feedback. Lastly, a depth study is conducted to quantify the effect of the motion cues generated by the force-based displacement of the virtual elements on depth perception.

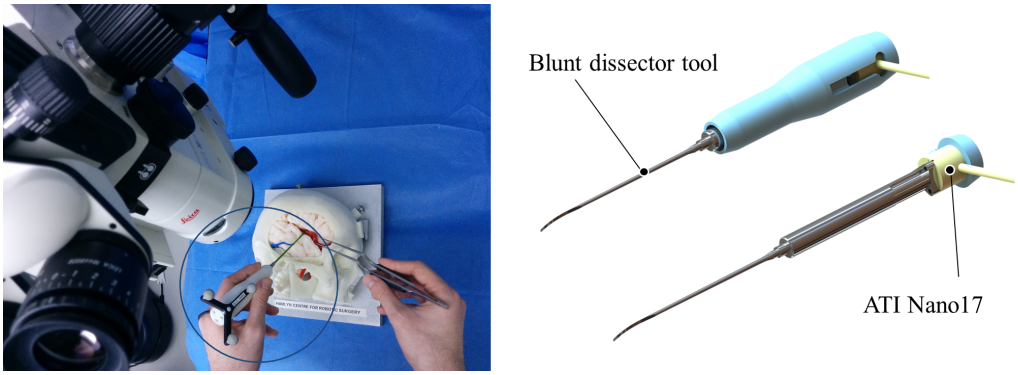


Figure 7.2: Left: surgical setup under a microscope with a blunt dissector with force feedback (left hand), and forceps (right hand) Right: design of the blunt dissector

Results demonstrate the efficacy of the proposed system for visual force feedback, as well as the importance of motion cues for depth perception.

7.2 System Overview

A blunt surgical dissector (Yasargil FD304R, Braun Aesculap) was axially affixed to a force-torque (F/T) sensor (ATI Nano17, ATI Industrial Automation, Inc) that can sense both lateral and axial forces that are applied to the dissector tip. This dissector and F/T sensor assembly were integrated into a hand-held casing that incorporated optical tracking markers, as shown in Figure 7.2. The force signal was acquired by a realtime linux computer through a PCI interface. Images from the Leica M525 OH4 (Leica Microsystems Ltd., UK) surgical microscope were acquired with an AJA HA5-Plus HDMI to SDI converter (AJA Video Systems Inc., California, USA) linking to an AJA Kona 4 PCIe framegrabber. A Polaris Vicra optical tracking system by NDI (Waterloo, Ontario, Canada) served to track the microscope, the hand-held tool, and register the virtual scene.

The images acquired by the surgical microscope are first used to detect the tip of the blunt dissector. This information, combined with the force reading, is used to perform the NPR processing on the images for the desired affect. The resulting image is used as a texture in a virtual scene containing the virtual anatomical elements used for the augmented reality guidance. The virtual elements are updated using the infrared tracking, jointly with the displacement modelling updated with

the force information. Figure 7.3 shows a high-level overview of the system.

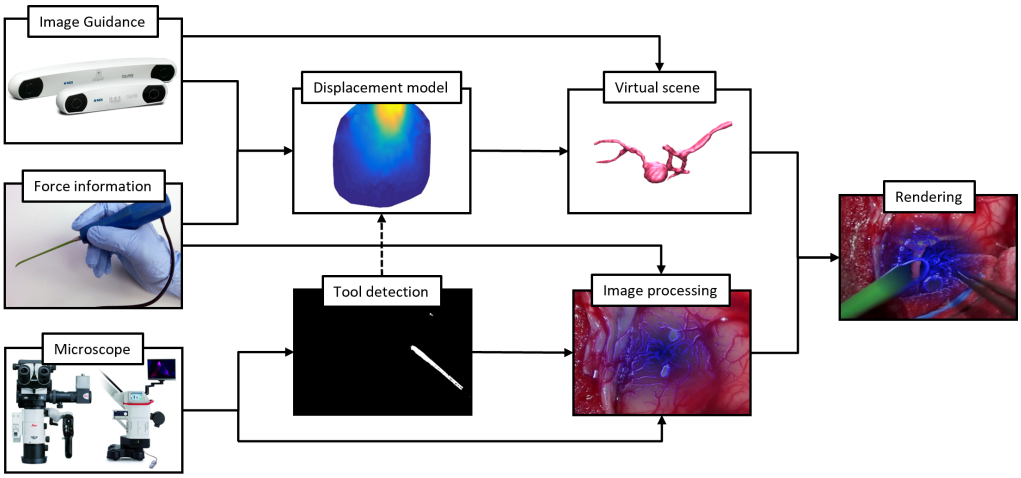


Figure 7.3: High-level system overview.

7.3 Augmented Reality Modulation

As noted in chapter 2, there are numerous ways of choosing how to process images for the purpose of augmented reality. These trade-offs are resolved by making specific design choices regarding how the overlay should appear. The principal purpose of NPR is to render the surface tissue as a translucent layer, while keeping salient anatomical features distinct enough to provide navigation and depth cues [137]. To accomplish this NPR computes p-q values representing a measure of the local surface slope of the tissue. The model used in [137] considers a simplified set of conditions in which the camera and the light source are close to the tissue. The light source is assumed to behave as a point light source, and the surface is assumed to behave as a smooth Lambertian surface. Under these assumptions, the image irradiance at pixel coordinates x and y is given by:

$$E(x,y) = s_0 \rho_m \frac{(1 - p_0 x - q_0 x)^3}{Z_0^2 (1 - p_0 x_0 - q_0 y_0)^2 (1 + p_0^2 + q_0^2)^{\frac{1}{2}} (1 + x^2 + y^2)^{\frac{3}{2}}}, \qquad (7.1)$$

where s_0 represents the point light source intensity, $(x_0 Z_0, y_0 Z_0, Z_0)$ are the coordinates of the surface point with surface normal $(p_0, q_0, -1)$, and ρ_m is the mean

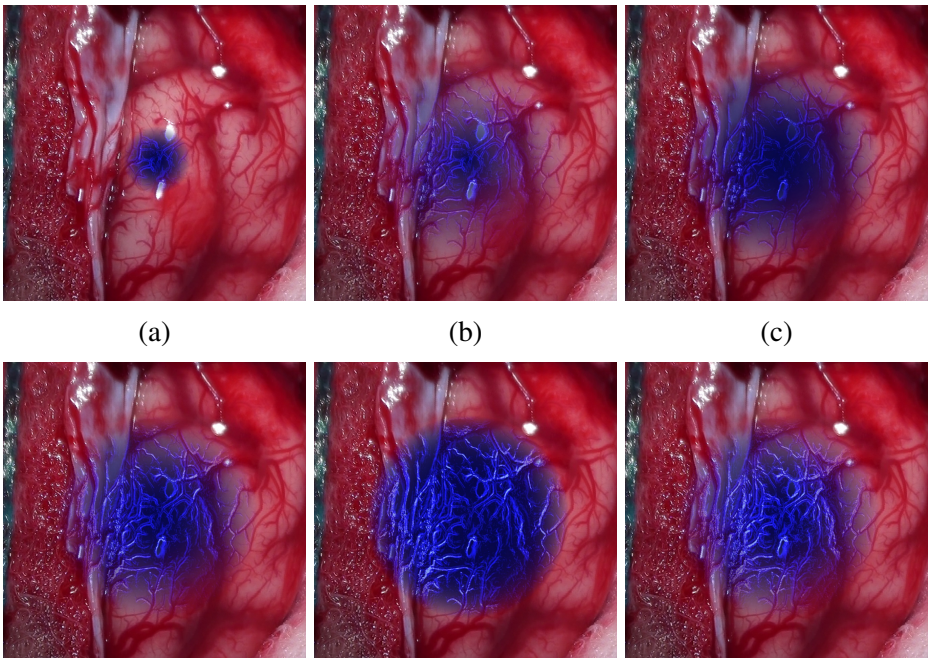
albedo coefficient of the neighbourhood of the considered surface point. By partially differentiating the irradiance equation 7.1 relative to x and y we can solve for p_0 and q_0:

$$p_0 = \frac{[R_x\alpha + 3x]\,[(-yR_y + 3)\alpha - 3y^2] + [yR_x\alpha + 3xy]\,[R_y\alpha + 3y]}{[yR_x\alpha + 3xy]\,[xR_y\alpha + 3xy] - [(-xR_x + 3)\alpha - 3x^2]\,[(-yR_y + 3)\alpha - 3y^2]}$$

$$q_0 = \frac{[xR_y\alpha + 3xy]\,[(R_x\alpha + 3xy] + [(-xR_x + 3)\alpha - 3x^2]\,[R_y\alpha - 3y]}{[yR_x\alpha + 3xy]\,[xR_y\alpha + 3xy] - [(-xR_x + 3)\alpha - 3x^2]\,[(-yR_y + 3)\alpha - 3y^2]}$$

$$\alpha = 1 + x^2 + y^2, \quad R_x = \frac{1}{E}\frac{\partial E}{\partial x}(x, y), \quad R_y = \frac{1}{E}\frac{\partial E}{\partial y}(x, y). \tag{7.2}$$

To use NPR for visual force feedback, the p, q values are used to generate a mod-



(a) (b) (c)

(d) (e) (f)

Figure 7.4: Illustration of the effect of the main visual force feedback parameters: (a) small radius, (d) large radius; (b) low blending coefficient, (e) high blending coefficient; (c) low IR detail level, (f) high IR detail level.

ulated texture relative to the force exerted, increasing the level of details in the highlighted features when higher forces are applied. The output of the rendering

is a texture referred to as the inverse realism texture (IR), of width W_{ir} and height H_{ir}, and centred around a point $P_c(x_c, y_c)$. Smooth surfaces are represented as uniform dark blue patches, and high surface gradients are brightened proportionally to their gradient value. Furthermore, pixels shift from the IR texture values to the original image values proportionally to their distance from the texture centre. For NPR to be useful for visual force-feedback, these properties must be modulated by the forces detected. The parameters used to modulate the IR texture are:

- The detail level of features during the pq value generation.
- The radius R_{ir} of the circular texture, when $W_{ir} = H_{ir} = R_{ir}$.
- The blending between the input image and the IR texture, G_{ir}.
- The colour palette of the IR texture.

Let A be the output pixel, B the input pixel, and C the IR texture pixel at the point $P(x, y)$. Let G_{ir} be the blending coefficient between B and C. The blending is modulated by:

$$A = S_{ir} \times B + (1 - S_{ir}) \times C, \quad S_{ir} \in [0, 1] \tag{7.3}$$

with

$$S_{ir} = \left(\frac{(x - x_c)^2}{W_{ir}^2} + \frac{(y - y_c)^2}{H_{ir}^2} \right)^{G_{ir}}. \tag{7.4}$$

And so:

$$A = C + (B - C) \times \left(\frac{(x - x_c)^2}{W_{ir}^2} + \frac{(y - y_c)^2}{H_{ir}^2} \right)^{G_{ir}}. \tag{7.5}$$

Figure 7.4 illustrates the effect of the different parameters on the output IR texture, and Figure 7.5 shows the modulation of the texture for different forces applied by the surgical instrument. Recent studies have demonstrated that the use of inverse realism techniques using NPR improve depth perception [131]. The proposed method makes use of this property to combine visual force feedback with image guidance techniques. When exerting excessive force near critical pathology, the proposed method reveals anatomical structures hidden under the tissue surface, alerting the surgeon to their presence. A virtual surgical scene containing

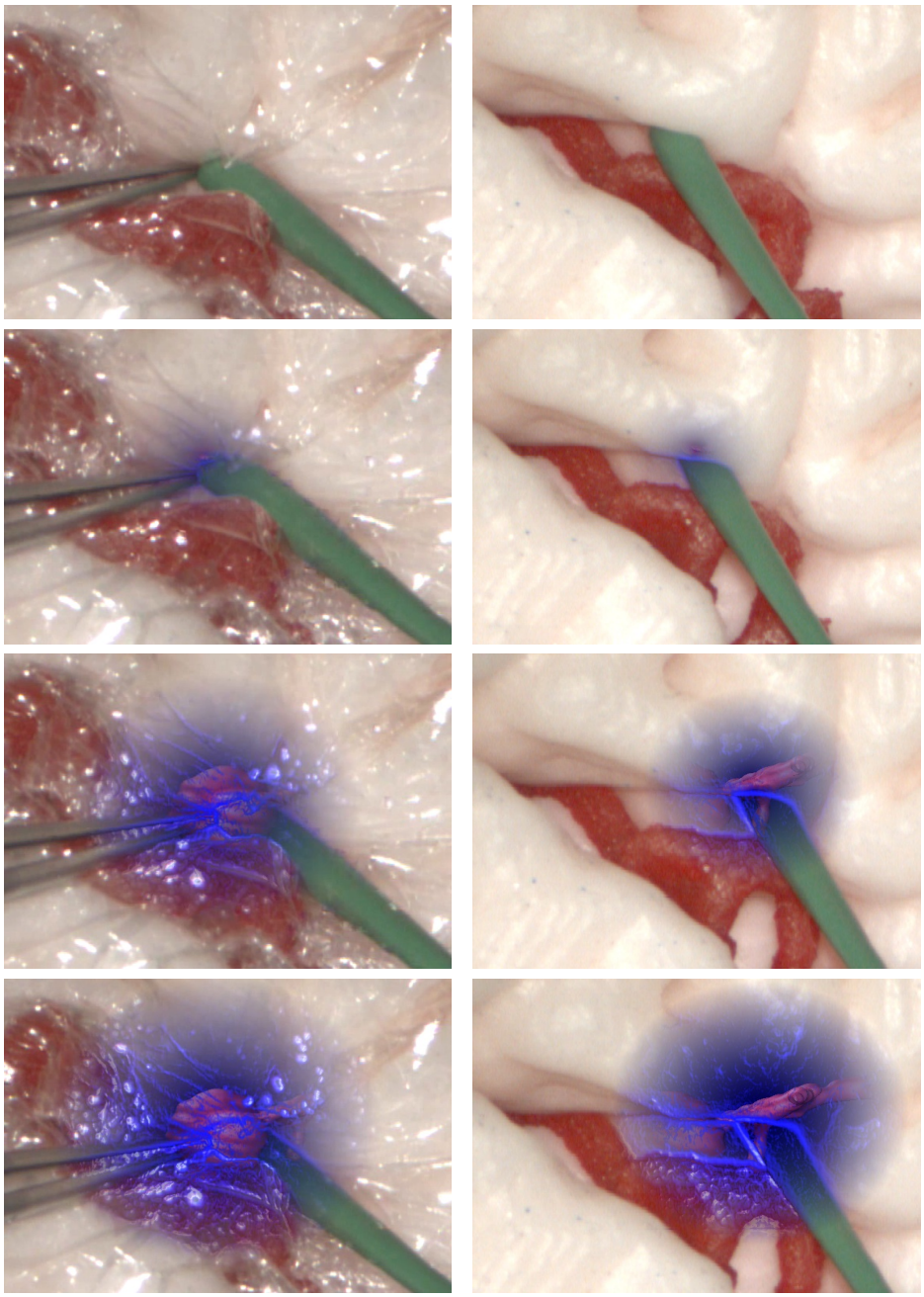


Figure 7.5: From top to bottom: gradual increase in forces exerted; Left: with plastic membrane to simulate the dura; Right: without the plastic membrane.

triangle meshes of the structures of interest is registered to the real-world operating scene, allowing correct overlaying of the virtual elements on the operative images. The resulting combination of the IR texture modulation with the augmented reality on in vivo data is shown in Figure 7.6.

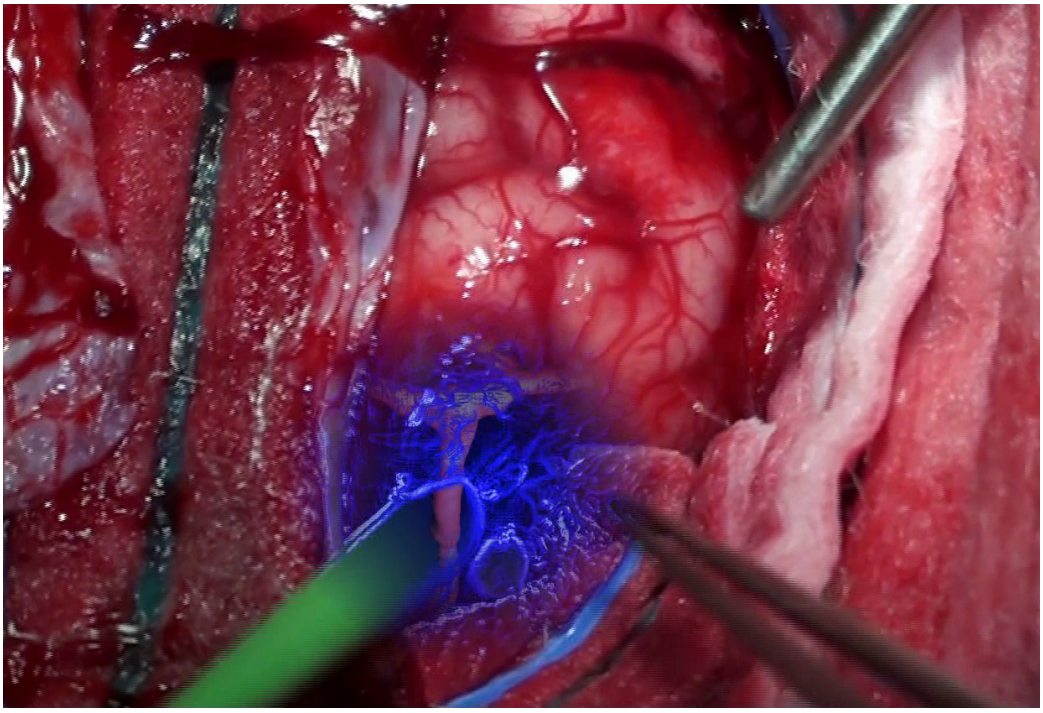


Figure 7.6: Example of the visual force feedback using in vivo data from a porcine brain dissection task.

7.4 An Efficient Rendering Pipeline for Stereo Augmented Reality

7.4.1 Real-Time Pipeline

In order for the augmented reality scheme presented in section 7.3 to be useful in the operating theatre, the combined rendering operations must be performed with the little latency and with the highest frame rate possible. Additionally, it is important to preserve the 3D vision provided by the surgical microscope. This

section describes the different optimizations and implementation choices made to obtain a real-time rendering pipeline for the visual force feedback.

The augmented reality pipeline for the visual force feedback can be broken down into three main phases. First the IR texture is created by processing the input images. Second, the IR texture is used as part of the rendering for a virtual scene. Finally, the resulting left and right images rendered from these virtual scenes are combined to obtain a row-interleaved stereo image. A key aspect of reducing latency for such a pipeline consists in minimizing image copies. As the input video frames are stored in a 4-channel RGBA 32 bits per pixel format in memory, a single 1920x1080 high definition (HD) stereo frame represents 16.6 Megabytes (MB) of memory. For indicative purposes only, CPU-based copies (RAM to RAM) take approximately $10\mathrm{ms}$, while CPU to GPU copies can take up to 30-40msdepending on hardware. Due to the computational requirements of the IR processing, GPU computing kernel operations using the Open Computing Language OpenCL are much more efficient than trying to use custom shaders from the Open Graphics Library (OpenGL). As such, a naive pipeline performing the image processing with OpenCL and rendering with OpenGL separately can quickly reach over $140\mathrm{ms}$of latency (one CPU-based copy and 3 CPU-GPU copies), without even considering any processing time. Limiting these exchanges are key to reducing latency. This is achieved in large part by using shared GPU memory buffers that can be used both for formal image processing, and for the rendering of virtual scenes. Figure 7.7 illustrates the optimized rendering pipeline.

The images are first copied from the AJA Kona 4 PCIe framegrabber into the workstation's RAM. The images are then copied into OpenCL buffers, in GPU memory. Note that it is possible to copy the images directly from the framegrabber to the GPU memory via DMA transfer. However, in the presented system some minor operations such as instrument tip detection were performed in parallel on the CPU, and so it was decided to retain this one extra copy. The image processing required to obtain the IR texture is performed using the images from these OpenCL buffers, along with external information such as the instrument tip position and force exerted obtained from the CPU. The images resulting from this processing are stored into shared OpenGL-OpenGL buffers. These buffers are typically OpenGL texture buffers shared with an existing OpenCL context. By switching ownership of the buffer from OpenGL to OpenCL and vice-versa, the

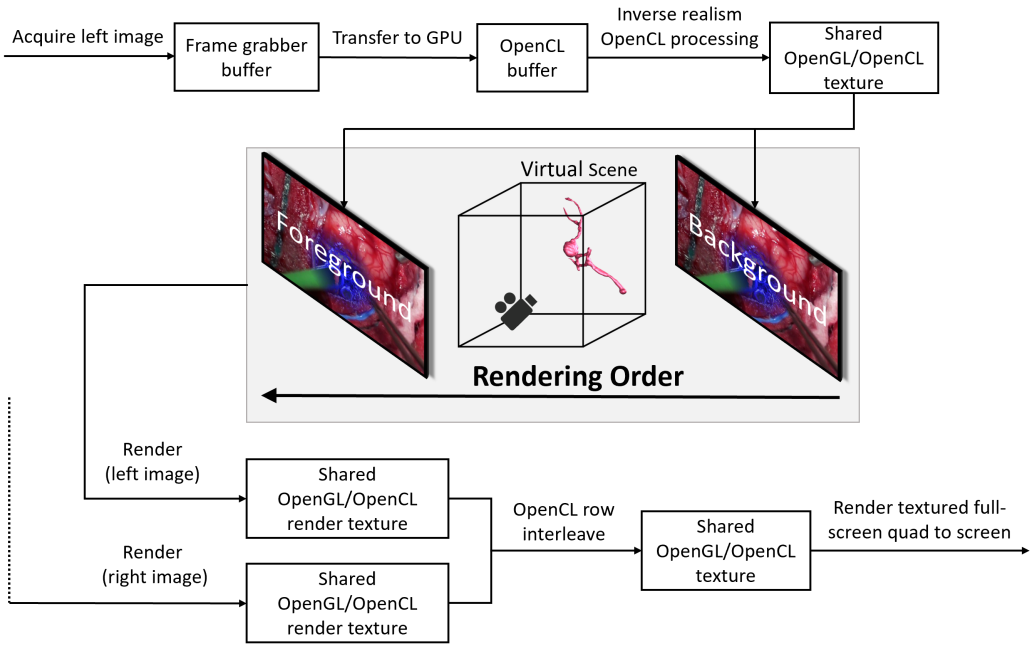


Figure 7.7: Real-time augmented reality pipeline. The dotted line represents a similar line of operations being performed for the right stereo image as for the left image.

same memory space can be used by the different libraries without requiring expensive copies. The newly filled OpenGL texture buffers containing the result of the IR processing can now be used as part of the rendering of virtual scene. The desired transparency behaviour is that the IR texture should appear on top of the virtual elements (e.g. aneurysm) where there are salient features, and underneath otherwise. This is reflected in the alpha-channel (transparency) of the IR texture, where areas of high opacity should obstruct vision of the aneurysm, and areas of low opacity should not. For this to be displayed correctly, the IR texture must be applied as both the foreground and the background for the rendering of the virtual scene: if the IR is only applied as the background, the virtual elements will always appear on top of the texture; if the IR is only applied as the foreground the correct opacity behaviour will be obtained, but areas of high transparency in the IR texture where the virtual element is not visible (due to its position or size) will be blended with the default OpenGL background colour, ruining the immersion of the scene. The final rendering order is: background, virtual scene, foreground. The resulting images obtained are stored into an OpenGL render texture, shared with OpenCL.

The desired stereo format, typically row-interleaved, can then computed using OpenCL, stored into a shared buffer, and finally rendered to the screen.

7.4.2 Stereo Rendering

Another important design decision is the choice of how to place the virtual cameras for stereo rendering. This decision impacts how depth is perceived throughout the scene by defining the difference between left and right images for different virtual objects. A useful metric is the distance in pixels between the position of the same 3D point in the left and right images, called parallax. Another useful definition is that of the screen plane. The screen plane of the system refers to the virtual plane in which elements have zero parallax. When looking at objects, human eyes automatically adjust their focus and angle to converge on their visual target, placing the object at zero parallax. Other objects not at that same depth will appear slightly out of focus and generate a negative or positive parallax, depending on whether they are in front of behind the target object. When looking at a stereo monitor, the user's eyes will focus on the screen. Objects rendered with zero parallax will appear to be "on" the screen, hence why the zero-parallax plane of a virtual scene is referred to as the screen plane. Real camera rigs typically choose between placing both cameras forward axis parallel to each other, or angling them inwards so that their forward axis converge at some distance. In a converged configuration there exists a point where both camera focal planes converge, generating an approximate zero-parallax plane. However this configuration suffers from artefacts due to the angle of the cameras. These include "keystoning", distortions at the screen corners where the different views of the two cameras are particularly noticeable, and stereo window violation, where objects in front of the focal plane (and thus appear in "front" of the screen) look like they are clipped by the edge of the screen if the virtual element reaches out of the field of view. While the parallel configuration does not suffer from these artefacts, it has the disadvantage of effectively setting the screen plane at infinity, making everything appear in front of the screen. This removes the possibility to make a target object appear on the screen, and creates limitations on how much depth can be introduced in a scene. Note that in real systems, most of these artefacts can be corrected in post-processing. In a virtual system, however, it is possible to create a third con-

figuration that natively corrects for these artefacts. This involves using a parallel camera configuration with off-axis frustums for both cameras to create a perfectly aligned screen plane, as illustrated in Figure 7.8. In determining how to construct

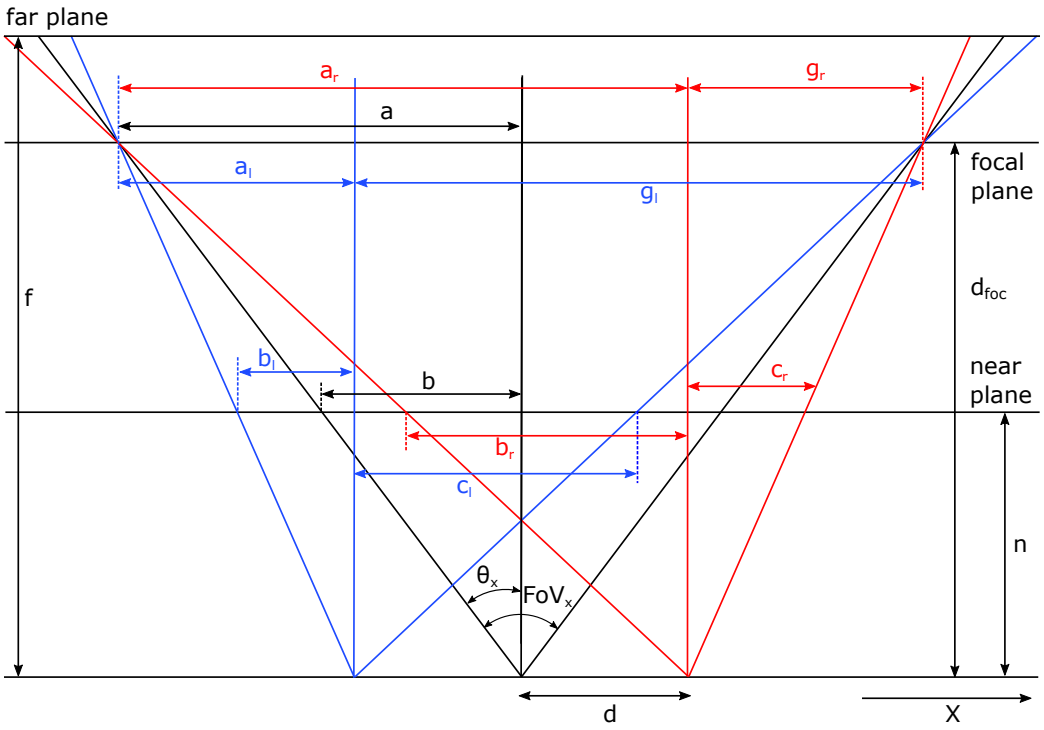


Figure 7.8: Stereo off-axis frustum with converging focal planes. The left-shifted frustum (corresponding to the left camera) is shown in blue, and the right-shifted frustum (corresponding to the right camera) is shown in red. The black frustum represents a baseline symmetrical frustum, and is used for geometrical purposes.

the off-axis frustums for the stereo system, represented in blue and red in Figure 7.8, we are mainly interested in the distances between the camera axis and the intersection point between the near plane and the frustums. These are b_l, c_l for the left camera frustum, b_r, c_r for the right camera frustum, and b for the symmetrical baseline frustum (represented as the black frustum in Figure 7.8). Furthermore, it is assumed that the baseline properties of those frustums are known, and so we are interested in the difference of these distances between the baseline frustum

and the shifted frustums. For the left-shifted frustum, these are defined as

$$\Delta b_{lf} \triangleq b_l - b, \tag{7.6}$$

$$\Delta c_{lf} \triangleq c_l - b. \tag{7.7}$$

Geometrically, we obtain

$$\Delta b_{lf} = \frac{n}{d_{foc}} a_l - b = \frac{n}{d_{foc}} a - \frac{n}{d_{foc}} d - b = -\frac{n}{d_{foc}} d, \tag{7.8}$$

$$\Delta c_{lf} = \frac{n}{d_{foc}} g_l - b = \frac{n}{d_{foc}} a + \frac{n}{d_{foc}} d - b = +\frac{n}{d_{foc}} d. \tag{7.9}$$

This result demonstrates that, to compensate for a horizontal shift in camera position to the left of distance d, the corresponding frustum's near plane boundaries along the x axis (illustrated in Figure 7.8) must be shifted to the right by $\frac{n}{d_{foc}} d$. This allows the focal plane of the frustum to remain in place. By symmetry, the same reasoning can be applied for the right frustum. More generally, given a frustum with left and right frustum near plane boundaries a positions l and r along the x axis, for a translation of value $d_{camera} \in \mathbb{R}$ along the x axis, the resulting change in left and right frustum near plane boundaries Δl and Δr to retain the same focal plane are:

$$\Delta l = l_{shifted} - l = \Delta r = r_{shifted} - r = -\frac{n}{d_{foc}} d_{camera}. \tag{7.10}$$

The general matrix form for a perspective projection frustum is

$$P_{frustum} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2f \times n}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}, \tag{7.11}$$

where t and b represent the frustum top and bottom near plane boundaries along the up axis (typically the y axis), and f is the distance from the camera to the far plane. To understand how the use of off-axis frustums impacts the visualisation of the augmented reality, the frustum shift can be back-propagated to the intrinsic matrix of the real camera. For this, the perspective projection frustum must first be expressed in terms of the camera intrinsics. Using a pinhole model for the

real camera, the perspective projection frustum for the augmented reality virtual camera can be expressed as

$$P_{AR} = P_{ortho} \times P_{camera}, \tag{7.12}$$

$$P_{ortho} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}, P_{camera} = \begin{bmatrix} \alpha & 0 & C_x & 0 \\ 0 & \beta & C_y & 0 \\ 0 & 0 & n+f & n \times f \\ 0 & 0 & -1 & 0 \end{bmatrix},$$

where α and β are respectively the focal lengths along the horizontal and vertical axis of the camera, and C_x and C_y the principal point centers along the horizontal and vertical axis respectively. The matrix P_{camera} contains the camera intrinsics, with an added third row to preserve Z-ordering between the near and far planes. The camera matrix projects the frustum-shaped camera view-space into a cube-shaped space, effectively mapping 3D points to their 2D screen coordinates plus a depth ordering (Z coordinate). The resulting space is then transformed into a standard frustum by clipping and normalizing the cube-shaped space using an orthographic projection P_{ortho}. By expanding equation 7.12 we obtain

$$P_{AR} = \begin{bmatrix} \frac{2\alpha}{r-l} & 0 & \frac{2C_x+r+l}{r-l} & 0 \\ 0 & \frac{2\beta}{t-b} & \frac{2C_y+t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2f \times n}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}. \tag{7.13}$$

Notice that the same form as in equation 7.11 appears by performing the following substitutions

$$\begin{aligned} r' &= \frac{n}{\alpha}(r + C_x) \\ l' &= \frac{n}{\alpha}(l + C_x) \\ t' &= \frac{n}{\beta}(t + C_y) \\ b' &= \frac{n}{\beta}(b + C_y). \end{aligned} \tag{7.14}$$

Using the substitutions from equation 7.14 and the results from equation 7.10, the shift in principal point center along the horizontal axis corresponding to the shift in frustum boundaries can be expressed as

$$\Delta C_x = \frac{\alpha}{n}\Delta r' = \frac{\alpha}{n}\Delta l' = -\frac{\alpha}{d_{foc}}. \tag{7.15}$$

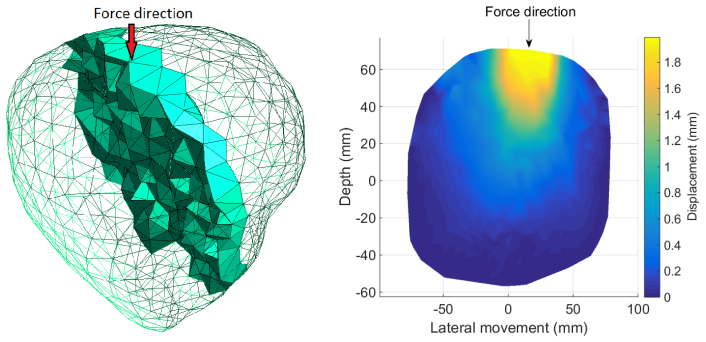
In summary, the method described above allows for the rendering of an augmented reality stereo scene while avoiding the limitations of naive stereo implementations. Given that real surgical camera systems often compensate in post-processing to avoid stereo artefacts, this method is useful for avoiding the introduction of new and inconsistent artefacts during the rendering of the virtual scene.

7.5 Force-Based Displacement

7.5.1 Deformation Modelling

While the IR texture allows the augmented reality overlay to appear embedded below the tissue surface, it does not provide any motion cues. Unless the registered microscope or skull are moved, the overlaid structure will appear static regardless of the deformation of the surface tissue. The force information from the hand-held instrument can be used to simulate this motion, by displacing the overlaid structure relative to the forces exerted.

Preliminary work with a linear displacement model suggested that the model did not have to be highly accurate in order to be compelling, due to the visual nature of the application. In order to strike a balance between simplicity and accuracy, a finite element model (FEM) simulation was conducted. A professional neurosurgical phantom (Kezlex, Tokyo, Japan) was scanned and a tetrahedral mesh of 1483 nodes obtained. A downwards force exerted from the top of the brain tissue simulated the push of a blunt dissector, and the boundary conditions were set by areas of the skull consistent with a microscopic operation. An implementation of the total Lagrangian explicit dynamics algorithm was used to model tissue deformation, employing a constant lumped mass matrix and a proportional damping matrix with Rayleigh damping coefficient $\alpha = 7.5E + 01$. The constitutive law followed that of a compressible neo-Hookean material, with Young's



(a) (b)

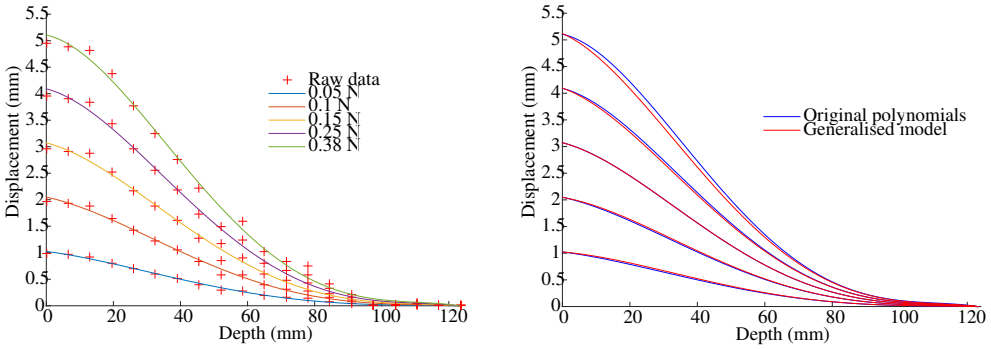
Figure 7.9: Propagation of displacements in the brain for a surface deformation of 2mm. (a) Section of the model. (b) Map of displacements in the section.

modulus $E = 3240$ Pa and Poisson's ratio $\eta = 0.499$ [193]. The results of the simulation were studied for different sections of the model, as shown in Figure 7.9. For each of five simulations chosen to cover the range of forces typically applied in neurosurgery, displacement data of the tissue at different depths was recorded. Polynomials of up to degree 6 were fitted to the data. Degree 4 polynomials were chosen as they presented the best global fitting while retaining a monotonic behaviour which matched the original data over the interval of interest, as shown in Figure 7.10(a). Results from the literature [193] are used to convert force values from the sensor to surface displacements. Finally a generalized model was developed, describing the displacement $D(x)$ at a depth x for a surface displacement D_0:

$$D(x) = \frac{D_0}{n}\left(\sum_{i=1}^{n} \frac{a_i \times x^4 + b_i \times x^3 + c_i \times x^2 + d_i \times x + e_i}{D_{0_i}}\right), \tag{7.16}$$

where n is the number simulations (5 in this work), D_{0_i} is the surface displacement for the i^{th} simulation, and a_i, b_i, c_i, d_i, and e_i the coefficients of the fitted polynomial for the i^{th} simulation. The generalized model is compared to the polynomials of the individual simulations in Figure 7.10(b).

The model described in equation 7.16 can be derived by normalizing the results from the different simulations. This is achieved by dividing all the displacement values obtained in a simulation i by the surface displacement for that simula-



(a) (b)

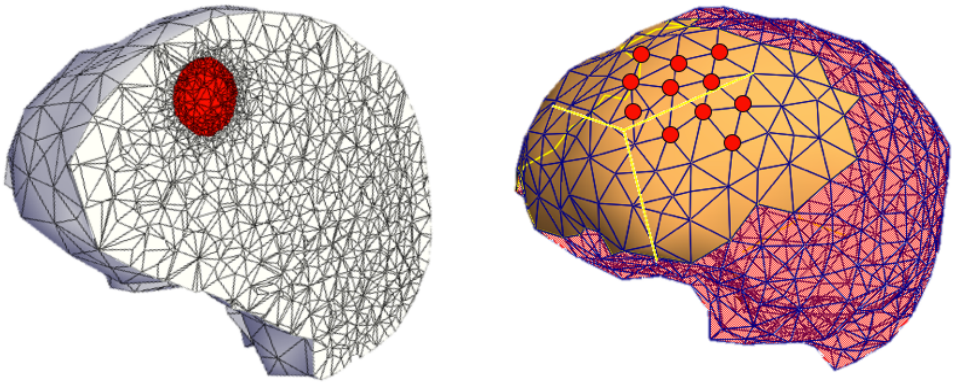
Figure 7.10: Displacements relative to depth (a) Raw data (1 out of 5 points) and polynomial fitting. (b) Comparison between the generalised model and the polynomial fitting.

tion D_{0_i} (i.e the maximum displacement). In so doing all the displacement values are normalized to a $[0; 1]$ range, and the resulting normalized polynomials of each simulation can be averaged to a polynomial P_{norm}. The displacement $D(x)$ at a depth x for a surface displacement D_0 can then be derived as follows:

$$D(x) = D_0 \times P_{norm}(x) \tag{7.17}$$

7.5.2 Alternative Methods for Deformation Modelling

The deformation model presented in section 7.5.1 presents a useful tool for introducing additional realism and depth perception. However it requires multiple time-consuming FEM simulations to be run prior to the operation. Additionally, the model only covers a limited set of applied forces and force orientations on the brain surface. While the current approach is still valid as both the starting point for the operation and the expected trajectory are determined during pre-operative planning, two other related approaches were studied in collaborative work. A neural network approach to explore the use of much more comprehensive simulations in real-time, and a vision-based approach to explore simulation-free deformation modelling of surface tissue.



(a) (b)

Figure 7.11: (a) Tetrahedral mesh of the brain (white), with the tumour (red). (b) Display of the boundary conditions for the FEM simulations. The static regions are in pink, and the dynamic regions are in orange. The red dots represent the nodes on which different forces will be applied. Picture from the co-authored publication [194].

Neural Network for Deformation Prediction

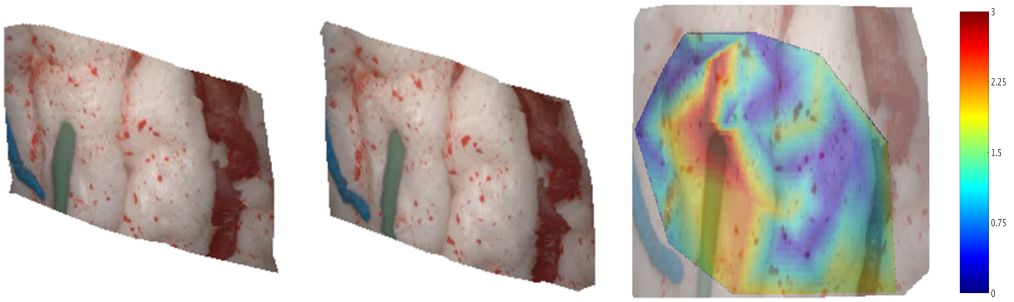
The approach presented in section 7.5.1 is limited to modelling the deformation at one point in the brain, generated by a varying force along a single direction. While the blunt dissector is tracked using infrared markers to find the direction in which the force is applied, the same model is used regardless of the direction. The virtual element is displaced along the direction of the force by an amount dictated by the model, based solely on the amplitude of the force exerted. Thus, for different tool tip positions and orientations than the ones used in the FEM simulations, the resulting displacement is potentially less accurate. For this reason, the chosen simulation conditions of the model presented in section 7.5.1 represent those most present during the relevant part of the operation, meaning small deviations from these conditions will not generate a noticeable error. The proof of concept studies presented below in section 7.6 highlight this, showing the usefulness of that model. However, in use cases where the conditions differ significantly from those used in the simulation, the augmented reality could potentially be mislead the surgeon. As such, it was also decided to investigate more detailed models.

Two main challenges arise from trying to increase the accuracy of the defor-

mation model. The first is that multiple simulations need to be run to cover a broader range of cases. The second is that the result of these numerous simulations needs be available in a real-time model. Both of these aspects are explored in [194]. A semi-automated workflow was developed to generate the tetrahedral meshes and select the areas of interest with minimal user input required. Multiple simulations are conducted in parallel, over a discretised range of force positions, orientations, and amplitudes. Figure 7.11 illustrates an example group of nodes where different forces will be applied during the simulations. A neural network is then trained to model the result of these simulations. The neural network generalizes well the highly non-linear nature of these deformations, taking advantage of all the simulations available. In addition, regression of the neural network can be performed in real-time. Results from [194] show that neural networks are able to efficiently learn and generalize from the FEM simulations, allowing a more accurate deformation model to be built.

Vision-Based Deformation Modelling

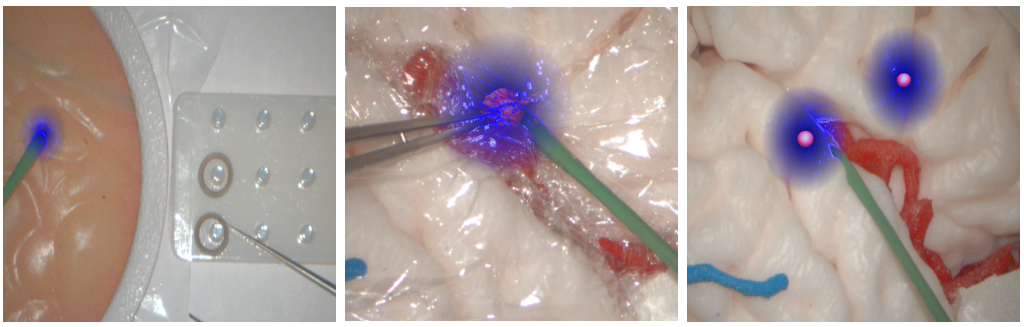
This approach, presented in [195], uses computer vision to estimate the deformation of the surgical tissue and the corresponding forces applied by the surgical dissector. The key advantage of this approach is that it does not require any potentially-lengthy FEM simulations to be run, nor any additional hardware such as force sensors or infrared trackers. It is also able to cope with live deformations of the brain induced during the operation, whereas the simulation-based approaches presented previously rely on pre-operative data. This method uses stereo reconstruction to estimate the deformation of the tissue, taking advantage of a new stereo correspondence method to estimate the tissue structure. This is illustrated in Figure 7.12. Note that this method is not able to recover deformations inside the brain; only the surface deformation is recovered. Performance evaluations of this method demonstrate submillimetre accuracy, highlighting the potential of this approach for surface deformation modelling. Combined with a separate model for the deformation of the brain tissue below the surface, potentially using similar approaches to those presented in sections 7.5.1 or 7.5.2, a deep deformation model adapting to intra-operative changes to the tissue could be developed with minimum hardware requirements.



(a) (b) (c)

Figure 7.12: (a) Estimated 3D structure before tool-tissue interaction. (b) Estimated 3D structure during tool-tissue interaction. (c) Deformation heatmap. The colorscale units are in millimetres. Figure re-used from Springer open access co-authored publication [195].

7.6 User Studies



(a) (b) (c)

Figure 7.13: Illustrations of the three user studies. (a) Retraction study. (b) Dissection study. (c) Depth study.

To quantify the usefulness of the proposed visual force feedback augmented reality scheme, three user studies were conducted. A retraction and dissection study (Figures 7.13(a) and 7.13(b)) were conducted to assess the performance of users to respect neurosurgical force constraints. A depth study (Figure 7.13(c)) was also conducted to assess the usefulness of the deformation model presented in section 7.5.1 towards depth perception. In the studies below, the live input video stream used was obtained from a Leica M525 OH4 surgical microscope. The output HD video stream was displayed on a screen with stereo capabilities.

7.6.1 Retraction Study

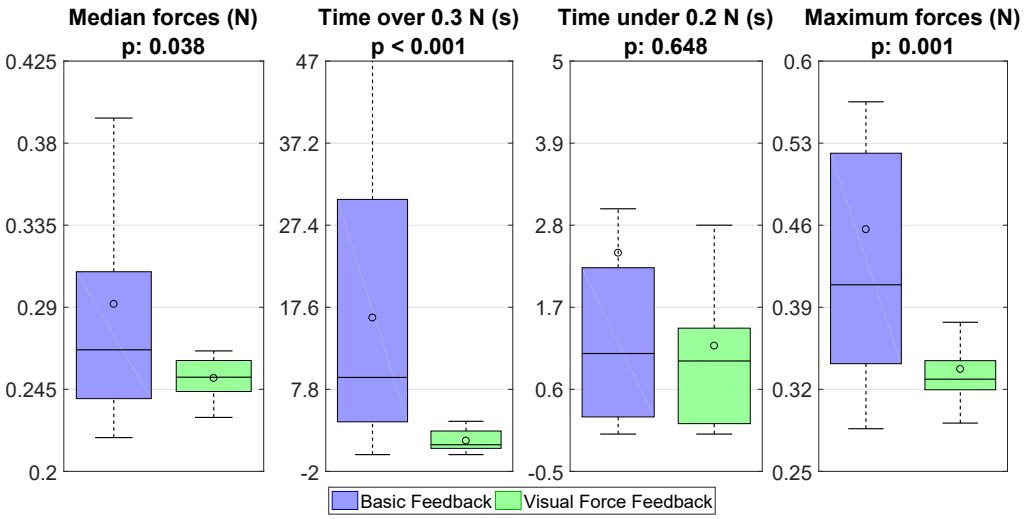


Figure 7.14: Primary outcomes of the retraction study.

Tissue retraction is among the most common manoeuvres performed in surgery. Eight participants were asked to use the blunt dissector in their non-dominant hand to provide retraction, and use the forceps in their dominant hand to simultaneously perform a peg transfer task. The forces exerted through the instrument were recorded, and a coloured display used to indicate when a minimum effective force of 0.2N was applied (red for forces under 0.2N, grey otherwise). The 0.2N threshold was chosen to simulate the force required to perform retraction. Each participant was asked to apply an increasing retraction force with the dissector,

	Mean		**Median**		**p-value**
	Default	Feedback	Default	Feedback	
Median forces (N)	0.292	0.251	0.267	0.252	0.038
Time over 0.3 N (s)	16.35	1.66	9.22	1.18	1.42×10^{-4}
Time under 0.2 N (s)	2.43	1.18	1.08	0.98	0.648
Maximum forces (N)	0.456	0.337	0.409	0.329	1.16×10^{-3}

Table 7.1: Mean, median, and statistical significance results from the visual force feedback retraction study.

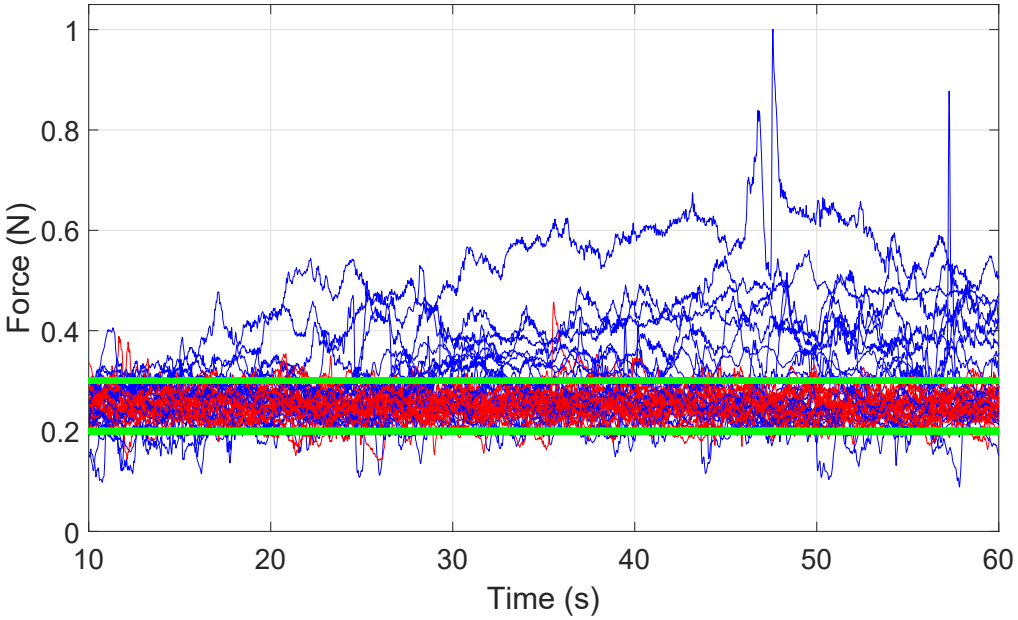


Figure 7.15: Forces exerted by the participants during the retraction study. Blue: forces exerted without visual force feedback. Red: forces exerted with visual force feedback. Green: target force threshold.

until the colour became grey. Participants were then asked to keep the indicator grey by maintaining this minimum force of 0.2N for 60 seconds. A force threshold of 0.3N was selected as indicative of excessive force. Participants performed the experiment 6 times, 3 times with visual force-feedback and 3 times without, in a randomized manner. The primary outcomes of the study were the median forces exerted, the maximum forces exerted, the time spent exerting forces under 0.2N, and the time spent exerting forces higher than 0.3N. These results are shown in Figure 7.14. Numerical values for the mean, median, and statistical significance of the primary outcomes are listed in Table 7.1. Additionally, a comparison of the forces exerted by the participants during the task with and without feedback is shown in Figure 7.15. As visible in those results, the presence of the grey/red visual indicator available in both modes meant that participants were always able to respect the 0.2N force limit. However, the lack of graduated force feedback in the control study made it challenging for participants to be aware of how much force they were exerting above the required 0.2N, resulting in significant increases in median and maximum forces, as well as time spent over 0.3N. The use of the vi-

sual force feedback resulted in a mean reduction in median forces of 14%, as well as a 19.7% reduction in maximum forces. Importantly, the median time spent over 0.3N was reduced by 87.2% with the visual force feedback. These results highlight the usefulness of the proposed visual force feedback for respecting force constraints. More specifically, the visual force feedback seems particularly useful to improve the constancy of forces applied. Figure 7.15 illustrates this further, showing that users were prone to suffer from drift and apply gradually increasing forces without realising it when they performed the task without visual force feedback.

7.6.2 Dissection Study

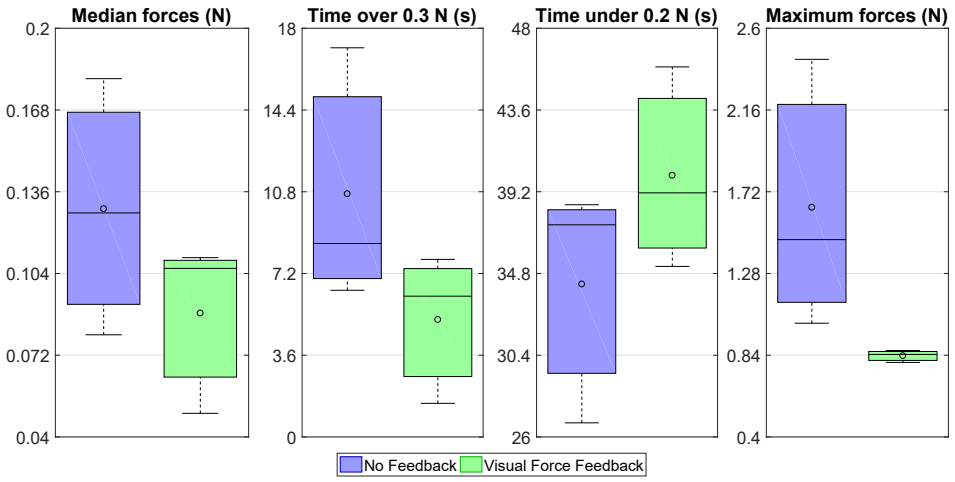


Figure 7.16: Primary outcomes of the dissection study.

Tissue dissection is also commonly performed in surgery, but represents a more complex manoeuvre than retraction. A surgical resident was recruited from a university hospital, and performed a dissection alternating at 2 min intervals between no force-feedback, and NPR visual force-feedback. The participant was asked to use the blunt dissector in their right hand to perform dissection, and use a sucker in their left hand to simultaneously provide counter-traction. A model brain was used to simulate dissection (Kezlex, Tokyo, Japan). The participant was asked to perform blunt arachnoid dissection of the Sylvian fissure on the model brain. A force threshold of 0.3N was selected as indicative of excessive force.

The study showed the use of NPR visual force-feedback resulted in a 31% mean decrease of median forces, as well as a 52% decrease in the mean overall time spent above 0.3N. The mean maximum forces exerted also decreased by 49%. Note that unlike in the retraction study, no minimum forces are required to perform the dissection task. As such, the increase in time spent under 0.2N with the visual force feedback is a positive outcome here. These results are summarized in Figure 7.16 and Table 7.2. As the study was conducted with a single surgical resident, statistical significance measures are not particularly meaningful and so not shown.

7.6.3 Depth Perception Study

The application of inverse realism overlay to microsurgery represents a unique challenge, because image injection on commercial operating microscopes is 2D. To mitigate for the loss of binocular depth cues, we developed and evaluated a modified image guidance platform that incorporates force data to displace the virtual embedded object. Six participants were asked to use the blunt dissector in their right hand to interact with a series of scenes, each containing two virtual spheres located within the brain phantom used for the FEM simulation. One sphere was 50% larger than the other, but both were placed at different simulated depths under the brain tissue, so that size alone could not be a sufficient depth indicator. Each participant was asked to contact both virtual sphere three times with the dissector, and then select which was deeper. Participants performed this task 24 times, 12 times in 2D and 12 times in 3D. In both the 2D and 3D experiments, the system would alternate between using displacement modelling or not

	Mean		**Median**	
	Default	Feedback	Default	Feedback
Median forces (N)	0.129	0.089	0.128	0.106
Time over 0.3 N (s)	10.71	5.17	8.52	6.20
Time under 0.2 N (s)	34.23	40.08	37.42	39.14
Maximum forces (N)	1.64	0.837	1.462	0.844

Table 7.2: Mean and median results from the visual force feedback dissection study.

in a randomized but equally distributed manner (i.e. for both 2D and 3D display modes, the task was performed 6 times with displacement modelling and 6 times without in total).

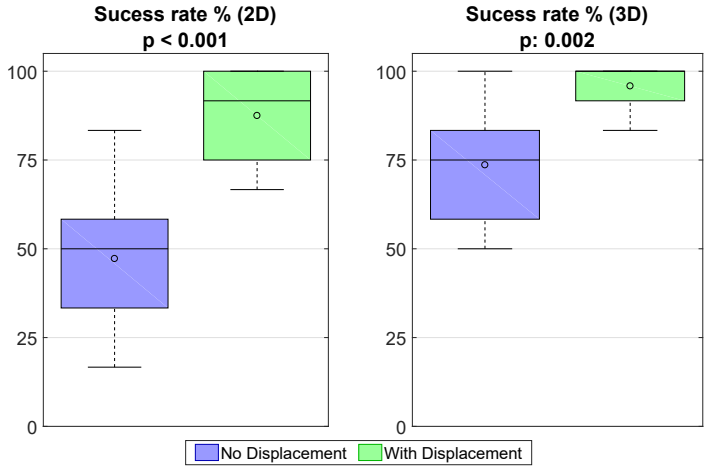


Figure 7.17: Primary outcomes of the depth study.

	Mean		Median	
	Static	Displacement	Static	Displacement
2D	47.2	87.5	50.0	91.7
3D	73.6	95.8	75.0	100

Table 7.3: Mean and median success rates from depth study.

Without displacement modelling (static display), the median success rate was 50% in 2D (random), and 75% in 3D. With displacement modelling, the median success rate rose to 91.7% in 2D, and 100% in 3D. The differences between with and without displacement modelling were statistically significant in both 2D and 3D ($p < 0.01$). This represents a 83.4% improvement in depth perception in 2D for the displacement modelling relative to the static display, and a 33.3% improvement in 3D. Guided 2D also showed statistically significant improvements over unguided 3D. These results are summarized in Figure 7.17. Numerical values for the mean and median of the success rates for the different setups are listed in Table 7.3. The statistical significance between those setups are listed in Table 7.4.

	2D default	2D displacement	3D default	3D displacement
2D default	-	1.65×10^{-4}	5.20×10^{-3}	2.87×10^{-5}
2D displacement	1.65×10^{-4}	-	0.062	0.140
3D default	5.20×10^{-3}	0.062	-	1.80×10^{-3}
3D displacement	2.87×10^{-5}	0.140	1.80×10^{-3}	-

Table 7.4: p values from Wilcoxon ranksum tests for the depth study.

7.7 Discussion and Conclusion

This chapter presents a novel visual force feedback scheme scheme for hand-held microsurgical instruments. A NPR overlay is used to provide both graduated visual force feedback and image guidance based on the forces exerted on the tissue by the surgical instrument. Furthermore, displacement modelling is used to translate the virtual overlay to provide enhanced depth perception through motion cues. User studies are conducted to evaluate both the efficacy of the NPR overlay as visual force feedback, and the improvement in depth perception from the displacement modelling.

Results from the retraction and dissection study show the suitability of the NPR overlay for visual force feedback. Participants exerted significantly lower median and maximum forces when using the visual force feedback, and were able to respect force constraints. It should be noted that this result can be mainly attributed to the presence of any sort of visual feedback, not specifically the visual force feedback using the NPR overlay. Indeed, in the retraction study where a grey/red colour display was used to indicate whether the minimum force was respected, the use of the NPR overlay did not provide statistically significant improvements for the time spent under the minimum force. However, the choice of the NPR overlay allows image guidance information to be seamlessly integrated with the force feedback, and to minimize the visual clutter on the scene by retaining salient features.

Additionally, the depth study conducted to evaluate the displacement model shows the importance of motion cues for depth perception. The results show significant improvement between the cases with and without displacement modelling, both in 2D and in 3D. The results of the experiment in 2D without dis-

placement modelling show a median success rate of 50%, which was expected as participants did not have any valid cues to judge the depth and so had to pick at random. The addition of 3D shows a statistically significant improvement for the unguided case. While the improvement in the percentage of correct answers is not statistically significant between 2D and 3D using displacement modelling, the success rate for both is significantly higher without displacement modelling. From these results it can be concluded that the force displacement is essential for estimating the relative depth of objects in a neurosurgical scene, at least equivalent to stereo alone. Furthermore the results suggest that the force displacement enhances the effect of a stereo display, due to added 3D motion cues.

The method presented in this chapter is a form of context-aware augmented reality behaviour: the augmented reality appears when the information is relevant to the surgeon, and disappears otherwise. As in chapter 6, the augmented reality display does not require explicit activation by the surgeon. This allows the surgeon to benefit from the surgical navigation at times where it is relevant but they might not think to activate it. A limitation of the proposed system is that only force and instrument tracking information to evaluate when and where to display the visual force feedback augmented reality. While this information is particularly relevant in the presented application, the system could be further improved by integrating additional information into the decision process for the visual force feedback display. Future work could develop models similarly to chapter 6, to integrate the force information and provide more fine-tuned control over the visual force feedback. Lastly, a criticism can be made regarding the displacement modelling presented in this work. The displacement modelling requires detailed simulations to provide accurate results, and those results may be invalidated as the surgery progresses and the anatomy changes. However, asides from the alternatives proposed in section 7.5.2, it should be noted that the main purpose of the displacement modelling is to provide enhanced depth perception during the dissection phase, when the virtual elements revealed are not close to the tissue surface. In this case, the accuracy of the modelling is not as important as deformations remain small in neurosurgery. Indeed, in early prototypes of the proposed system a heuristically parametrized linear displacement model was used. While the linear model was later discarded in favour of the non-linear FEM-based model presented in this work, initial testing showed that the change in model did not

significantly impact depth perception. This would suggest that the displacement model remains useful for depth perception so long as the displacement introduced is credible relative to the surface deformation of the brain tissue.

8

Conclusions and Future Research Directions

New surgical robots are continuously being developed to push the boundaries of minimally invasive surgery. These robots possess every-growing capabilities thanks to technological innovations, such as smaller form-factors, increased number of jointed articulations, enhanced sensing capabilities, or improved imaging modalities. These innovations grant robotic systems the capability to perform increasingly delicate surgical procedures with minimal patient trauma. However, in order for these robot capabilities to be beneficial in practice, the surgeon must be able to make use of them intuitively when controlling the system. Due to the complexity of surgical robot systems, dedicated interfaces are needed to allow the surgeon to control the many components that comprise the robot. However, while robot capabilities are only constrained by current technological progress, human operators have physical limits in how they can interact with a system. For this reason, the more complex the surgical robot, the greater the importance of the interface used to control that system. Without suitable human-robot interfaces to disguise the complexity of surgical robots, their advanced capabilities become effectively unusable in practice.

A common limitation of existing human-robot interfaces is their lack of adaptability to the context in which they are used. Instead, a static, pre-defined behaviour is typically used, which requires explicit surgeon input to be changed. This approach forces compromises to be made regarding system behaviour, often

with important undesirable consequences. For example, a static motion scaling factor will either restrict the user's capability to cross large workspaces, or reduce precision when performing delicate tasks. Likewise, manual on/off controls for surgical navigation information using augmented reality will either obstruct the surgical view when the augmented reality is on, or provide no guidance when it is off. In this thesis novel human-robot interfacing methods were presented, using intention recognition and scene context to adapt system behaviour implicitly.

8.1 Summary of Thesis Achievements

The first technical contributions of this thesis were presented in chapter 3, with the development of an interaction detection scheme to control the surgical camera. Unlike in existing approaches, the activation and de-activation of the system was performed implicitly by fixating the screen, without requiring active input from the user. Smooth velocity profiles generated from the gaze data ensured safe system behaviour even under noisy conditions, and allowed the user to safely take their attention away from the system and resume usage at any point. The gaze tracking was used to perform localised stereo reconstruction, allowing the system to track the user's gaze in 3D space. This was exploited to provided automatic zooming capabilities to the surgical camera by maintaining an optimal camera-to-tissue viewing distance based on the focal length of the camera. Additionally, this 3D gaze tracking was used to automatically re-position and target a laser pointer simulating a focussed energy delivery system, with millimetre precision. User studies conducted to evaluate the performance of the gaze-contingent control showed clear improvements over a manually controlled system, with a 52.92% reduction of median task completion times for the gaze-controlled system. This chapter highlighted the usefulness of gaze data to infer user intention, particularly when used in an implicit manner. The results of the user studies demonstrated the substantial time gains that can be obtained by integrating implicit interfaces naturally in the operative workflow. Furthermore, this chapter outlined how gaze data can synergise with additional scene context to provide valuable functionality, in this case automatic targeting and camera zooming.

Building on the concepts introduced in chapter 3, chapter 4 presented a novel adaptive motion scaling scheme using gaze data to infer user intention implicitly.

The system combined gaze tracking with instrument and hand-motion tracking to detect and facilitate reaching motions, by dynamically modulating the motion scaling factor. A highly dexterous bi-manual robotic system was developed, combining the dexterity of single-port instruments with the triangulation capabilities of external robotic arms. This system was used to provide a realistic platform with which to test the impact of master device workspace limitations. User studies consisting of a ring transfer task were conducted, requiring both crossing a large workspace and precise motions. Results demonstrated the effectiveness of the adaptive motion scaling, with a 50% reduction in number of times the system was clutched as well as a significant 17% reduction in median task completion times. The adaptive motion scaling presented in this chapter constitutes an effective and inherently safe solution to the workspace limitations of master devices, an issue which is present in the majority of existing master-slave surgical robots.

The implicit nature of the adaptive motion scaling presented in chapter 4 means that it is does not require any input from the user to function, and is designed to feel as transparent as possible. In chapter 4, the dynamic change in the motion scaling factor generated by the system was parametrised by a set of manually chosen parameters. These parameters dictated important aspects of the adaptive motion scaling, such as the maximum scaling possible, or how rapidly to increase or decrease the motion scaling. As these parameters directly impacted the manner in which the user's commands are performed by the robot, they played an important role on the intuitiveness of the resulting motion. To further improve the transparency of the adaptive motion scaling, the work presented in chapter 5 researched principled approaches to optimally tune those parameters, and in a more generic sense the core parameters of any human-robot interface. Two Bayesian optimisation approaches were explored: a graded approach were users had to rate the behaviour of the system, and a preference-based approach where users expressed their preference for certain system behaviours over others. User studies conducted with the same setup as in chapter 4 demonstrated a highly significant improvement in the intuitiveness of the motion resulting from the optimisation process, with over 90% of users preferring the optimised system behaviour over the manually tuned one in a blind randomised study. No statistical difference was found in terms of task completion time, showing that the increased intuitiveness did not come at the cost of performance. Additionally, the result of the graded op-

timisation showed a statistically significant 50% median reduction in the number of times the master device was clutched. These results highlight the importance of using principled approaches rather than intuition to tune system behaviour in human-robot interfaces, particularly when that behaviour directly impacts the control of the robot. Indeed, these results demonstrate that the manual tuning process was suboptimal in ways that were not initially predicted, and significantly better system behaviour was obtained when relying on a principled optimisation process.

The implicit gaze-contingent methods developed in chapters 3, 4, and 5 were exploited in chapter 6 to design a novel approach for the augmented reality display behaviour. This approach utilised scene context including gaze tracking to infer the optimal augmented reality display mode at any given moment. This context-based augmented reality display addressed an important limitation of existing augmented reality systems: the reliance on the surgeon to manually activate and de-activate the surgical navigation. Using this context-based behaviour, the system was able to constantly provide meaningful image guidance while simultaneously minimizing visual clutter. This was reinforced by the addition of a partial augmented reality display mode, showing the augmented reality overlay only around specific areas when relevant. As the augmented reality display was adapted automatically, the introduction of additional display modes could be done seamlessly, providing better-suited image guidance in specific situations. Unlike in the previous chapters, the desired behaviour was learned entirely from supervised samples using a much more flexible model. Additionally, the model's uncertainty propagation was exploited to direct the user's attention towards low certainty configurations during training, speeding up the training process. A detailed simulator of a robotic tumour contour marking task was developed, to assess the usefulness of the context-based augmented reality compared to manually toggled on/off augmented reality. Simulated changes in the brain texture were implemented, including simulated bleeding which needed to be addressed by participants. This was used to quantify the risk of inattention blindness caused by the augmented reality. Results from the user study showed a significant median reduction in bleeding time of 44.5% using the context-based augmented reality mode, as well as a 10.7% reduction in median task completion time. These results demonstrate the importance of adapting augmented reality displays to minimise visual clutter, and emphasize the ergonomic limitations involved in manually toggling the

augmented reality display.

Chapter 7 further explored methods for the seamless integration of augmented reality in surgery, this time in the context of hand-held surgical robots for microsurgery. A method combining visual force feedback and image guidance was developed, using the force-sensing capabilities of a hand-held blunt dissector. By modulating an augmented reality overlay with the detected forces, this approach allowed the surgeon to visualise when excessive forces were applied, while simultaneously revealing any nearby critical anatomical structures at risk of being damaged by the instrument. Additionally, the force information was also used to displace the virtual elements shown by the augmented reality using a model generated from finite element simulations. The use of a non-photorealistic rendering retaining surface features for the augmented reality, the motion cues generated by the displacement model, and an efficient stereo rendering pipeline all contributed to generate a compelling visual force feedback granting see-through vision. User studies were carried out to assess the ability of the modulated augmented reality overlay to relay force information, and evaluate the importance of the displacement modelling for depth perception. Results from a retraction and a dissection study showed that participants were able to respect specific minimum force constraints while exerting significantly lower median and maximum forces thanks to the visual force feedback. Results from the study assessing depth perception showed that the use of displacement modelling resulted in a median 83.4% improvement in depth perception in 2D, and a 33.3% improvement in 3D. These results demonstrate the synergy between visual force feedback and image guidance, and highlight how the two can be used together in a single interface.

8.2 Discussion and Future Research Directions

The work presented in this thesis constantly makes use of the same underlying strategy: information gathered from the ongoing procedure is used as a substitute for explicit user input. As demonstrated in this thesis, implicit interfaces can provide substantial advantages to robotic systems by adapting the system behaviour depending on the information available. These interfaces effectively behave like a very fast surgical assistant entirely dedicated to manually adjusting the robot behaviour during the operation to suit the surgeon's needs, or choosing what sur-

gical navigation information to show and when. As such, these implicit interfaces constitute a form of low-level automation, where modelled decision processes automatically adjust system behaviour using real-time information.

The first and most obvious means of improving the implicit interfaces presented in this thesis is by exploiting more information. As with any automated decision process, using additional relevant information would allow for the design of more detailed and effective interface behaviour. This can be particularly beneficial for approaches concerned with detecting user intention. The introduction of force-sensing capabilities to the robotic instruments used in chapters 4, 5, and 6 could be used to improve the quality of the intention-driven interfaces presented in those chapters. For the adaptive motion scaling, the detection of forces applied on the instruments could provide a way to rapidly reduce the motion scaling to prevent collisions, for example. For the smart augmented reality mode presented in chapter 6, force information could be used to detect contact between the cutting tool and the brain tissue, allowing the user to define more specific use-cases for the different augmented reality modes.

Depth recovery of the surgical scene could also be used to refine the behaviour of the adaptive motion scaling, by considering the distances between the current instrument position and the 3D target point instead of their 2D projections in the camera plane. Accurate depth recovery could also greatly improve the displacement modelling presented in chapter 7, as illustrated in [195]. Continuing with considerations related to visual information, a segmented and labelled representation of the surgical video feed could be extremely useful to improve implicit interfaces, allowing the definition of behaviour specific to certain anatomical structures. Likewise, the detection of high-level surgical phases could provide another useful source of information to adapt system behaviour. However, while these additional sources of information can be highly beneficial, they should not be included at the cost of system reliability. As human-robot interfaces are a means for the surgeon to interact with the robotic system, it is typically preferable to design simpler but reliable interfaces rather than ambitious but seemingly inconsistent ones.

These concepts of reliability and intuitiveness are crucial to the functionality of human-robot interfaces. They are even more critical to implicit interfaces, that act without user input and modify system behaviour automatically. If the user is

not capable of understanding why the interface modifies the system behaviour the way it does, the supposedly implicit interface becomes extremely explicit in a very negative manner. The issue is in part emphasized by the fact that the inner workings of these interfaces are hidden from the user. While poorly designed interfaces with simple and obvious behaviour will be tedious and cumbersome, users will be able to adapt to them and eventually make use of the system. A simple example of this is the frustum effect present in laparoscopic surgery, requiring all surgeon trainees to learn to invert their motions. Although highly counter-intuitive at first, surgeons eventually learn to adapt to it so much so that they sometimes then feel uncomfortable using a robotic system that compensates for this inversion. Implicit interfaces merging information from multiple sources, on the other hand, can be much more challenging to adapt to if poorly designed.

Asides from the sources of information available, the key factors dictating the intuitiveness of implicit interfaces are the model structure employed, and the manner in which those models are tuned. In this thesis, chapter 5 is entirely dedicated to studying the impact of interface parameters on system performance. As shown in that chapter, relying on intuition and manual optimisation to tune a complex interface is significantly suboptimal, and can result in unintuitive and potentially dangerous system behaviour. These results highlight the importance of using principled approaches to optimise human-robot interfaces, something mainly ignored in the literature. The approaches presented in this thesis, however, could benefit from the use of more flexible models. In chapters 4 and 5 only the parameters of the adaptive motion scaling model were optimised, the model structure itself was manually chosen to reflect the intuited design considerations. While the optimisation process significantly improved the system behaviour, the choice of the model structure could have itself been suboptimal. In chapter 6, the constraints on the model structure were specified through kernel and mean functions, allowing much greater model flexibility. However dynamic effects were not considered, as well as non-stationarity considerations. These could both be incorporated in future work to improve the interface behaviour.

A limitation of these optimisation and training steps is the time requirement imposed on users of the system. Both training and optimisation processes require users to rate or label the performance of the system. This step can be time consuming to perform in a user-specific manner, particularly if the input feature space

is large. A potential solution to this problem consists in pre-training models with expert users that can then be further refined by individual users. Partially mentioned in chapter 6, this solution could use models that can be updated on the fly such as Gaussian processes. The core functionality of the interface would be loosely encoded through pre-training performed by an expert user, allowing users to subsequently refine it and specify their individual preferences. This approach would allow surgeons to start using the system right away without risking unsafe behaviour, and shape the system to their preferred behaviour over time.

A final potential extension for the work presented in this thesis is the concept of descriptive control. As discussed previously, implicit interfaces represent a low level of autonomy in surgery, where user intention and system information is used to automatically adapt the robot behaviour. With sufficient scene information and appropriate models, it would be possible to go beyond simply adapting the system behaviour to a more optimal one, and instead allow the surgeon to directly describe the desired system behaviour. The surgeon could use such descriptive control interfaces to dictate surgical subtasks to the robot, executed through a form of supervised autonomy. While this may at first seem significantly beyond current technological capabilities for reliable use, it would in fact not necessitate much more than is already available to be possible in certain applications. For example, using a haptic interface, scene depth recovery, and force sensing, it would be feasible to create an interface allowing a surgeon to command a robotic forceps to grasp, retract, and hold steady a specific anatomical surface. By moving one step away from the purely one-to-one nature of current master-slave interfaces, these descriptive control interfaces would allow advanced robotic capabilities to be used much more simply, such as the use of systems with three or more surgical instruments working together. Such descriptive control interfaces could represent the next step towards automation in general surgical applications.

Appendix A

Theoretical Formulation of the Online Gaze Calibration

A.1 Application of Thin Plate Spline to Gaze Mapping

Given a vector $x \in \mathbb{R}^m$, a Radial Basis Function (RBF) is any real-valued function Φ whose value depends only on the distance from a centre vector $\boldsymbol{c} \in \mathbb{R}^m$, such that $\Phi(\boldsymbol{x}, \boldsymbol{c}) = \Phi(\|\boldsymbol{x} - \boldsymbol{c}\|)$, where $\|.\|$ denotes the Euclidean norm. Sums of RBFs are commonly used to approximate functions. They take the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} w_i \Phi(\|\boldsymbol{x} - \boldsymbol{c}_i\|), \tag{A.1}$$

where the approximating function $f(\boldsymbol{x})$ is represented as a sum of n RBFs, each associated with a different centre $\boldsymbol{c}_i$, and weighted by an appropriate coefficient w_i. This function defines a spatial mapping that maps any location $\boldsymbol{x}$ in space to a new location $f(\boldsymbol{x})$. In order to use the TPS [196] as the basis function, the TPS takes the form $\Phi(\boldsymbol{r}) = \boldsymbol{r}^2 ln(\boldsymbol{r})$, where $\boldsymbol{r} = \|\boldsymbol{x} - \boldsymbol{c}_i\|$.

To guarantee the unique existence of interpolants with a small variation to Equation A.1, lower order polynomials are added to $f(\boldsymbol{x})$ along with additional

conditions. Specifically, we obtain the following formulation for the TPS:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} w_i \Phi(\|\boldsymbol{x} - \boldsymbol{c}_i\|) + a + \boldsymbol{b}^T \boldsymbol{x}, \tag{A.2}$$

where $\boldsymbol{x} \in \mathbb{R}^m$, a is a constant, and $\boldsymbol{b} \in \mathbb{R}^m$. Equation A.2 gives rise to a unique interpolating function $f(\boldsymbol{x})$ using the TPS basis function $\Phi(\boldsymbol{r}) = \boldsymbol{r}^2 ln(\boldsymbol{r})$, with $\Phi(\mathbf{0}) \triangleq 0$. The additional component $a + \boldsymbol{b}^T \boldsymbol{x}$ is introduced to allow an affine offset to the zero-centred basis function. The uniqueness can be guaranteed provided that the centre vectors $\boldsymbol{c}_i$ of the basis functions are not collinear, and the following conditions are fulfilled:

$$\sum_{i=1}^{n} w_i = 0 \quad and \quad \sum_{i=1}^{n} w_i \boldsymbol{c}_i = (0, \ldots, 0)^T, \tag{A.3}$$

where $(0, \ldots, 0)^T$ denotes the zero vector in m dimensions. Under these conditions, the scalars w_i, a, and the vector $\boldsymbol{b}$ can be uniquely solved for.

To use the TPS basis function as the gaze calibration mapping, the unmapped Point of Regard (PoR) coordinate centroids, $\boldsymbol{P} = (x, y)$ are defined as the input, and hence $m = 2$. The mapping to the screen coordinates $f(\boldsymbol{P})$ is represented as:

$$f(\boldsymbol{P}) = \sum_{i=1}^{n} w_i \Phi(\|\boldsymbol{P} - \boldsymbol{c}_i\|) + a + \boldsymbol{b}^T \boldsymbol{P}, \tag{A.4}$$

where $\boldsymbol{c_i} = (\bar{x}_i, \bar{y}_i)$ is the i^{th} PoR feature vector obtained during calibration, and $\boldsymbol{b} = [a_x, a_y]$. The conditions for the scalars w_i, a, and the vector $\boldsymbol{b}$ to be solved uniquely become:

$$\sum_{i=1}^{n} w_i = 0 \quad and \quad \sum_{i=1}^{n} w_i \bar{x}_i = 0 \quad and \quad \sum_{i=1}^{n} w_i \bar{y}_i = 0. \tag{A.5}$$

The mapping in Equation A.4 specifies an approximation function $f : \mathbb{R}^2 \to \mathbb{R}$ in order to describe an eye tracking calibration mapping in $\mathbb{R}^2 \to \mathbb{R}^2$. As such, two TPS functions are used; one for the x-axis of the screen $f^x(\boldsymbol{P})$ and one for the y-axis of the screen $f^y(\boldsymbol{P})$. These mapping functions share their n PoR feature

vectors, $\boldsymbol{c_i} = (\bar{x}_i, \bar{y}_i)$ obtained during calibration to give:

$$M(\boldsymbol{P}) = \begin{bmatrix} f^x(\boldsymbol{P}) \\ f^y(\boldsymbol{P}) \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} w_i^x \\ w_i^y \end{bmatrix} \Phi(\|\boldsymbol{P} - \boldsymbol{c}_i\|) + \begin{bmatrix} a^x \\ a^y \end{bmatrix} + \begin{bmatrix} \boldsymbol{b}^x \\ \boldsymbol{b}^y \end{bmatrix} \boldsymbol{P}. \tag{A.6}$$

Here, w_i^x, a^x, and $\boldsymbol{b}^x$ represent the scalar coefficients and vector for the x-axis, and w_i^y, a^y, and $\boldsymbol{b}^y$ represent the scalar coefficients and vector for the y-axis. The resultant mapping function $M(\boldsymbol{P})$ now describes the mapping of the PoR coordinates $\boldsymbol{P}$ in the eye image plane to the screen pixel coordinate plane. The overall TPS calibration mapping is divided into two parts; the sum of the RBF weighted by the TPS coefficients w_i^x and w_i^y, which are bounded and asymptotically flat, and an affine part described by the last three terms. Importantly, three coefficients are required to express an affine transform, which means a minimum of three calibration points will be needed to compute the TPS mapping.

A.2 Thin Plate Spline Mapping Parameter Estimation

Once n calibration points have been extrapolated, with $n \geq 3$, the relevant mapping parameters including w_i^x, w_i^y, a^x, a^y, $\boldsymbol{b}^x$, and $\boldsymbol{b}^y$ can be obtained by solving the following linear system:

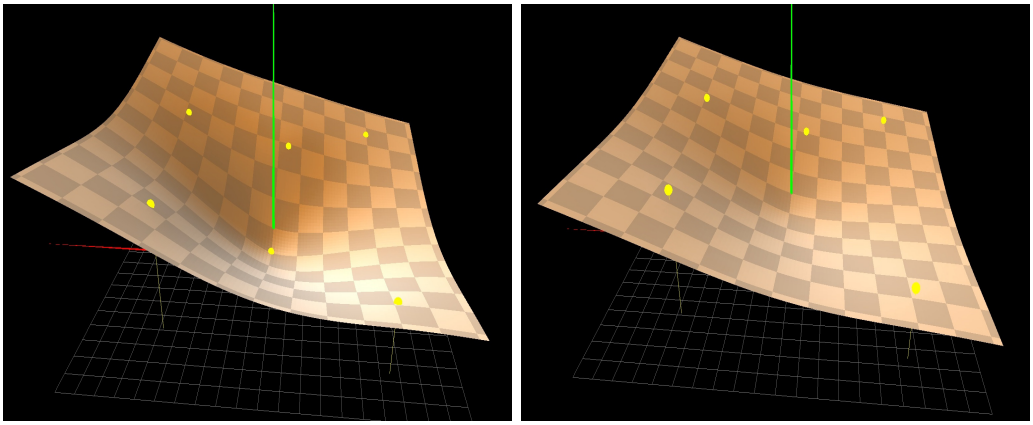
$$\begin{bmatrix} \boldsymbol{S}^T \\ \boldsymbol{0}^{3,2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{0}^{3,3} \end{bmatrix} \begin{bmatrix} \boldsymbol{t}^x & \boldsymbol{t}^y \end{bmatrix}, \tag{A.7}$$

where $\boldsymbol{S}$ is composed of the n calibration points' respective screen coordinates:

$$\boldsymbol{S} = \begin{bmatrix} S^x \\ S^y \end{bmatrix} = \begin{bmatrix} S_1^x & \dots & S_n^x \\ S_1^y & \dots & S_n^y \end{bmatrix}, \tag{A.8}$$

as illustrated in Figure 3.10. $\boldsymbol{0}^{3,3}$ and $\boldsymbol{0}^{3,2}$ denote the 3×3 and 3×2 zero matrices respectively. The calibration points' respective n PoR feature coordinates represented by $\boldsymbol{c_i} = (\bar{x}_i, \bar{y}_i)$ form the $n \times 3$ matrix $\boldsymbol{C}$, whose i_{th} row is $(1, \bar{x}_i, \bar{y}_i)$. Let $\boldsymbol{L}$ be a $n \times n$ matrix defined as:

$$\boldsymbol{L} = \Phi(\|\boldsymbol{c}_j - \boldsymbol{c}_i\|) + \lambda \cdot \alpha^2 \cdot I^{n,n}, \tag{A.9}$$



(a) (b)

Figure A.1: Illustrations of a six-point thin plate spline modelling visualised using TPSDemo[1]. (a) Regularization coefficient $\lambda = 0$. (b) Regularization coefficient $\lambda = 0.575$.

where $I^{n,n}$ is the $n \times n$ identity matrix, λ is the regularization coefficient, and:

$$\alpha = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\boldsymbol{c}_j - \boldsymbol{c}_i\|. \tag{A.10}$$

The regularization coefficient λ controls how strictly the model respects the control points. For $\lambda = 0$ the TPS interpolation will intersect each control point exactly, at the cost of having a more rapidly varying model between control points. For $\lambda \to \infty$, the TPS model approaches a least squares fitted plane. This is illustrated in Figure A.1.

$\boldsymbol{t}^x$ and $\boldsymbol{t}^y$ are arranged as:

$$\begin{aligned} \boldsymbol{t}^x &= \begin{bmatrix} w_1^x & \dots & w_n^x & a^x & \boldsymbol{b}^x \end{bmatrix}^T \\ \boldsymbol{t}^y &= \begin{bmatrix} w_1^y & \dots & w_n^y & a^y & \boldsymbol{b}^y \end{bmatrix}^T. \end{aligned} \tag{A.11}$$

Both $\boldsymbol{t}^x$ and $\boldsymbol{t}^y$ are $(n+3) \times 1$ vectors. By further denoting:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{L} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{0}^{3,3} \end{bmatrix}, \tag{A.12}$$

[1]Program written by Jarno Elonen available at `https://elonen.iki.fi/code/tpsdemo`

We can invert the system in Equation A.7 to solve for the scalar and vector parameters $\boldsymbol{t}^x$ and $\boldsymbol{t}^y$ as follows:

$$\begin{bmatrix} \boldsymbol{t}^x & \boldsymbol{t}^y \end{bmatrix} = \boldsymbol{K}^{-1} \begin{bmatrix} \boldsymbol{S}^T \\ \boldsymbol{0}^{3,2} \end{bmatrix}. \tag{A.13}$$

Appendix B

Quantification of Calibration Drift

B.1 Experimental Setup

Ten participants were asked to perform a laparoscopic ring transfer task using standard laparoscopic instruments. The same laparoscopic phantom that was used for the paper's studies was also used for this study. The experiment was divided in 7 steps:

1. An initial offline calibration of the eye tracker.
2. 1 minute of laparoscopic ring transfers
3. A second offline calibration
4. 1 minute of laparoscopic ring transfers
5. A third offline calibration
6. 1 minute of laparoscopic ring transfers, and the participant was asked to turn around and simulate talking to an assistant for 10 seconds.
7. A final offline calibration.

The participants were not asked to stay particularly still throughout the different steps, to represent natural behaviour during surgery. Reaching the specified area inside the phantom to perform the ring transfer was made sufficiently challenging to be able to simulate the motions that might arise from a real procedure.

B.2 Results

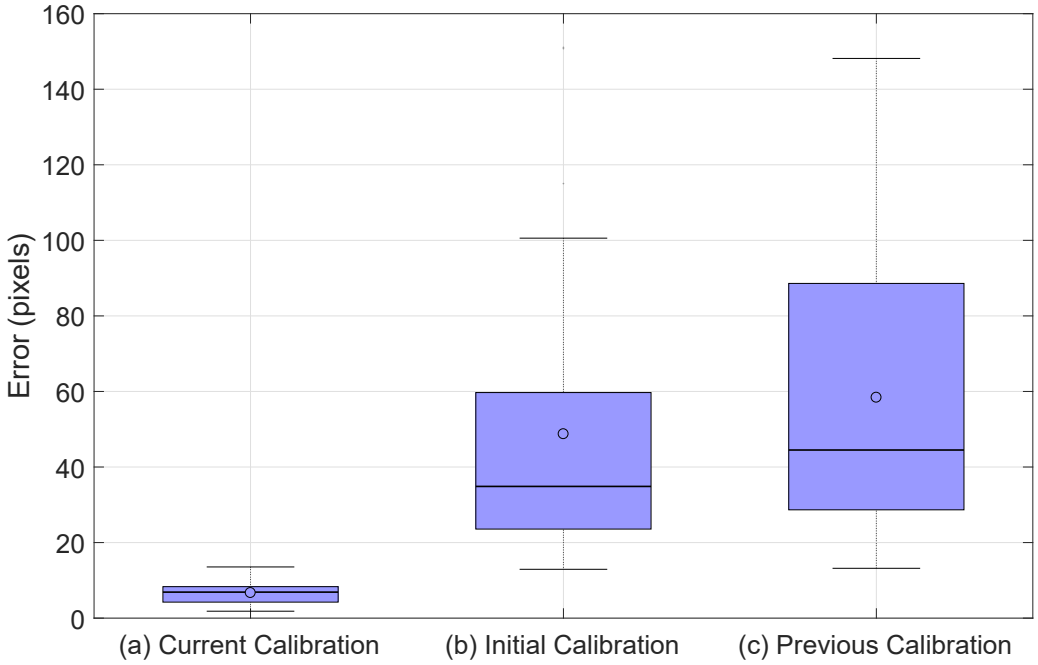


Figure B.1: Pixel error obtained using a sequence of gaze points and, from left to right: the latest TPS model, i.e the one generated from that sequence of points; the TPS model generated by the initial calibration procedure; and the TPS model generated by the previous calibration procedure.

A calibration procedure yields two elements: a sequence of detected gaze points corresponding to the calibration dots displayed on the screen, and a thin plate spline model computed from these points. For a sequence of gaze points issued from a given calibration, Figure B.1. shows the error computed using:

a) The thin plate spline model generated from that calibration (the calibration self error). Note that the regularization parameter of the thin plate spline model was chosen to insure a smoothly varying model without overfitting.

b) The thin plate spline model generated from the initial calibration

c) The thin plate spline model generated from the previous calibration

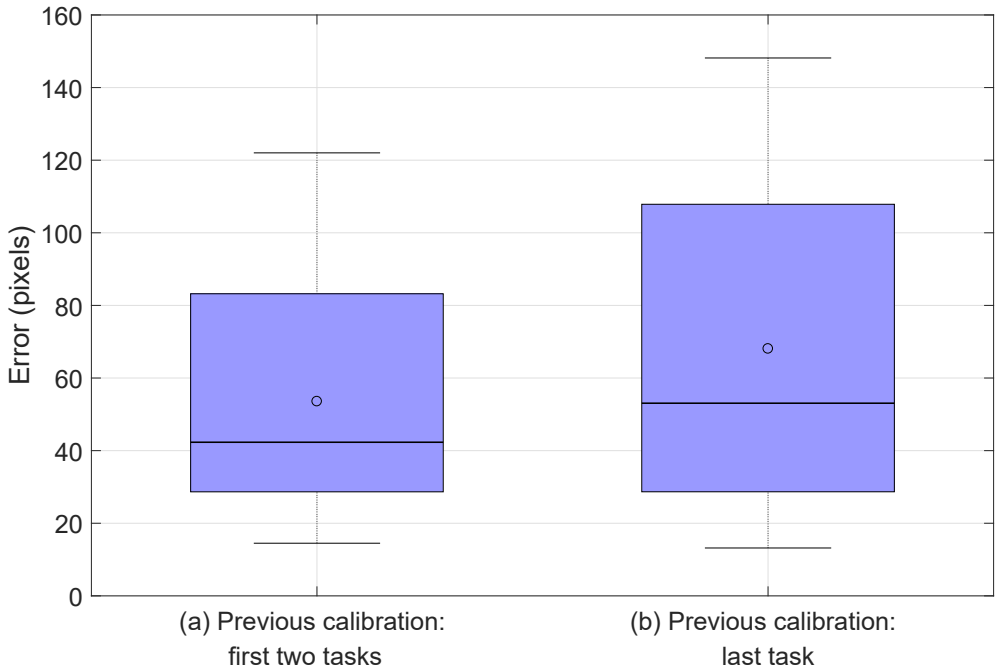


Figure B.2: Pixel error obtained using a sequence of gaze points and the TPS model generated by the previous calibration procedure for, from left to right: the first two tasks; and the last task.

Figure B.2 shows the impact of having the user turn around and talk after having performed the calibration. This done by splitting the data from Figure B.1(c) into two categories:

a) The error obtained using the thin plate spline models generated before the user was asked to turn around. These are the errors generated by the first and second calibration models, using the gaze points recorded from the second and third calibrations respectively.

b) The error obtained using the thin plate spline model generated by the third calibration, after the turning motion was performed. The errors are generated with the gaze points recorded from the final calibration.

The results shown in Figure B.1 highlight the substantial difference between a newly-calibrated system and one where the user has been free to move while performing a challenging task. A newly calibrated system has a non-zero error

due to the low-bending model used to prevent overfitting ($\lambda > 0$, see A). The median error for a newly calibrated system is 6.9 pixels, as opposed to a 34.9 pixel error when using the initial calibration with new gaze data after performing the tasks. Projecting this error on the 24 inch 1920x1200 pixel used for the study, this amounts to a 1.8 mm mean error for a newly calibrated system, as opposed to a 9.4 mm error after use. This five times increase in the detection error is easily noticeable in practice, as the mean error after use is nearly a centimeter.

Although not statistically significant relative to the initial calibration error, the median calibration error between successive calibrations is slightly higher, at 44.5 pixels (p-value of 0.2549). This is not unexpected as motion generated during the task will typically move the user away from their starting position for that task, but these movements would average out over the duration of the entire experiment (i.e the user is not continuously physically drifting in one direction). Effectively, this result means that the initial position a user had at the beginning of the experiment is closer on average to any given position after a task than the position the user had before starting that particular task.

Finally, Figure B.2 shows that asking the user to turn around and simulate a quick conversation does not make a statistically significant difference with regards to the calibration drift (p-value of 0.5824). While the mean, median, and standard deviation of the errors all increase, this result shows that most of the drift already appears during naturally occurring motions.

Appendix C

Bayesian Learning

This appendix introduces Gaussian Processes in the context of Bayesian linear regression, showing that they can appear from an extended Bayesian linear regression model with an infinite number of basis functions. The standard linear model with Gaussian noise can be written as

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{x}^T \boldsymbol{w} \\ y &= f(\boldsymbol{x}) + \varepsilon, \end{aligned} \tag{C.1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is a d-dimensional real-valued input vector from the dataset D, and y is the corresponding observation. The noise ε is independent, identically distributed (*iid*) with zero mean and variance σ_n^2. In the case of linear regression, the parameter vector $\boldsymbol{w}$ is a d-dimensional vector of weights.

Let X be the aggregation of n input vectors, such that $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ is a $d \times n$ matrix, and $\boldsymbol{y} = [y_1, \ldots, y_n]^T$ be the corresponding observations. Due to the *iid* assumption, the likelihood of the model can be written as

$$p(\boldsymbol{y} \mid X, \boldsymbol{w}) = \prod_{i=1}^{n} p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}) = \prod_{i=1}^{n} \mathcal{N}(\boldsymbol{x}_i^T \boldsymbol{w}, \sigma_n^2) = \mathcal{N}(\boldsymbol{y} \mid X^T \boldsymbol{w}, \sigma_n^2 I). \tag{C.2}$$

For added clarity, the notation $\mathcal{N}(v|\mu, \Sigma)$ is used to refer to a Gaussian distribution of a variable v, of mean μ, and covariance Σ. To generate a Bayesian formulation of the linear model, a prior belief must be placed on the parameters $\boldsymbol{w}$. A zero

mean Gaussian prior with covariance matrix Σ_p is chosen:

$$\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p). \tag{C.3}$$

Note that in this section, both σ_n and Σ_p are considered constant. Given the prior distribution and likelihood, the posterior distribution can now be expressed using Bayes' rule:

$$p(\boldsymbol{w} \mid \boldsymbol{y}, X) = \frac{p(\boldsymbol{y} \mid X, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{y} \mid X)}. \tag{C.4}$$

The marginal likelihood $p(\boldsymbol{y} \mid X)$ is a normalizing constant, and so independent of the parameters $\boldsymbol{w}$. As such, the posterior distribution can be computed up to proportionality without having to compute the marginal likelihood. Discarding constant terms, the posterior distribution can be written as

$$p(\boldsymbol{w} \mid \boldsymbol{y}, X) \propto \mathcal{N}(\boldsymbol{y} \mid X^T\boldsymbol{w}, \sigma_n^2) \times \mathcal{N}(\boldsymbol{w} \mid \mathbf{0}, \Sigma_p). \tag{C.5}$$

In the following, we expand equation C.5, crossing out terms independent of $\boldsymbol{w}$ while keeping proportionality:

$$\begin{aligned}
p(\boldsymbol{w} \mid \boldsymbol{y}, X) &\propto \exp\left(-\frac{1}{2\sigma_n^2}(\boldsymbol{y} - X^T\boldsymbol{w})^T(\boldsymbol{y} - X^T\boldsymbol{w})\right)\exp\left(-\frac{1}{2}\boldsymbol{w}^T\Sigma_p^{-1}\boldsymbol{w}\right) \\
&\propto \exp\left(-\frac{1}{2\sigma_n^2}\left[\cancel{\boldsymbol{y}^T\boldsymbol{y}} - \boldsymbol{y}^TX^T\boldsymbol{w} - \boldsymbol{w}^TX\boldsymbol{y} + \boldsymbol{w}^TXX^T\boldsymbol{w}\right] - \frac{1}{2}\boldsymbol{w}^T\Sigma_p^{-1}\boldsymbol{w}\right) \\
&\propto \exp\left(-\frac{1}{2}\boldsymbol{w}^T\left(\frac{1}{\sigma_n^2}XX^T + \Sigma_p^{-1}\right)\boldsymbol{w} + \frac{1}{2\sigma_n^2}\boldsymbol{y}^TX^T\boldsymbol{w} + \frac{1}{2\sigma_n^2}\boldsymbol{w}^TX\boldsymbol{y}\right) \\
&\propto \exp\left(-\frac{1}{2}\left[\boldsymbol{w}^TA\boldsymbol{w} - \boldsymbol{b}^T\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{b}\right]\right),
\end{aligned} \tag{C.6}$$

where $A = \sigma_n^{-2}XX^T + \Sigma_p^{-1}$ and $\boldsymbol{b} = \sigma_n^{-2}X\boldsymbol{y}$. Note that since X only has real entries, then XX^T is symmetric positive-semidefinite. Additionally, Σ_p is a symmetric positive definite covariance matrix. As the sum of a positive semidefinite and a positive definite matrix, A is symmetric and invertible, and so A^{-1} exists and is symmetric. Lastly, note that the since neither A nor $\boldsymbol{b}$ are dependent on $\boldsymbol{w}$, the term $\boldsymbol{b}^TA^{-1}\boldsymbol{b}$ is not as well. As such, the term $\boldsymbol{b}^TA^{-1}\boldsymbol{b}$ can be injected into equation C.6 while maintaining proportionality (referred to as *completing the*

square). Finally, we define $\bar{\boldsymbol{w}} = A^{-1}\boldsymbol{b}$.

$$\begin{aligned} p(\boldsymbol{w} \mid \boldsymbol{y}, X) &\propto \exp\left(-\frac{1}{2}\left[\boldsymbol{w}^T A\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{w} + \boldsymbol{b}^T A^{-1}\boldsymbol{b}\right]\right) \\ &\propto \exp\left(-\frac{1}{2}\left[\boldsymbol{w}^T A\boldsymbol{w} - \boldsymbol{w}^T AA^{-1}\boldsymbol{b} - \boldsymbol{b}^T A^{-1}A\boldsymbol{w} + \boldsymbol{b}^T A^{-1}AA^{-1}\boldsymbol{b}\right]\right) \\ &\propto \exp\left(-\frac{1}{2}\left[\boldsymbol{w}^T A\boldsymbol{w} - \boldsymbol{w}^T A\bar{\boldsymbol{w}} - \bar{\boldsymbol{w}}^T A\boldsymbol{w} + \bar{\boldsymbol{w}}^T A\bar{\boldsymbol{w}}\right]\right) \\ &\propto \exp\left(-\frac{1}{2}\left[(\boldsymbol{w} - \bar{\boldsymbol{w}})^T A(\boldsymbol{w} - \bar{\boldsymbol{w}})\right]\right). \end{aligned} \tag{C.7}$$

The result of equation C.7 is of the form of a Gaussian distribution with mean $\bar{\boldsymbol{w}}$ and covariance A^{-1}, demonstrating that the posterior is Gaussian distributed

$$p(\boldsymbol{w} \mid \boldsymbol{y}, X) \sim \mathcal{N}(\sigma_n^{-2}(\sigma_n^{-2}XX^T + \Sigma_p^{-1})^{-1}X\boldsymbol{y}, (\sigma_n^{-2}XX^T + \Sigma_p^{-1})^{-1}). \tag{C.8}$$

The tractability of the posterior is an important practical property of such models, which translates directly to Gaussian Processes. The marginal likelihood is also tractable as the convolution of conditional distributions:

$$\begin{aligned} p(\boldsymbol{y} \mid X) &= \int p(\boldsymbol{y} \mid X, \boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w} \\ &= \int \mathcal{N}(\boldsymbol{y} \mid X^T\boldsymbol{w}, \sigma_n^2 I)\mathcal{N}(\boldsymbol{w} \mid \boldsymbol{0}, \Sigma_p)d\boldsymbol{w} \\ &= \mathcal{N}(\boldsymbol{0}, X^T\Sigma_p X + \sigma_n^2 I). \end{aligned} \tag{C.9}$$

Finally, the predictive posterior distribution $p(y_*|\boldsymbol{x}_*, \boldsymbol{y}, X, \boldsymbol{w})$ for a new test vector $\boldsymbol{x}_*$ is also Gaussian distributed, which can again be shown using the properties of conditional Gaussian distributions:

$$\begin{aligned} p(y_* \mid \boldsymbol{x}_*, \boldsymbol{y}, X, \boldsymbol{w}) &= \int p(y_* \mid \boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{w})p(\boldsymbol{w} \mid X, \boldsymbol{y})d\boldsymbol{w} \\ &= \int \mathcal{N}(y_* \mid \boldsymbol{x}_*^T\boldsymbol{w}, \sigma_n^2)\mathcal{N}(\boldsymbol{w} \mid \bar{\boldsymbol{w}}, A^{-1})d\boldsymbol{w} \\ &= \mathcal{N}(\frac{1}{\sigma_n^2}\boldsymbol{x}_*^T A^{-1}X\boldsymbol{y}, \boldsymbol{x}_*^T A^{-1}\boldsymbol{x}_* + \sigma_n^2). \end{aligned} \tag{C.10}$$

Likewise, the predictive latent value f_* has a very similar distribution

$$f_* \mid \boldsymbol{x}_*, \boldsymbol{y}, X, \boldsymbol{w} \sim \mathcal{N}(\frac{1}{\sigma_n^2}\boldsymbol{x}_*^T A^{-1} X \boldsymbol{y}, \boldsymbol{x}_*^T A^{-1} \boldsymbol{x}_*). \tag{C.11}$$

Note that the prediction is made as a weighted average over all possible parameter values. This benefit stems from the Bayesian approach, where other approaches typically need to evaluate parameter sets separately.

The above formulation is strictly limited to a linear model. But what if the desired model is non-linear? One way to introduce non-linearity in equation C.1 is to use a mapping function $\phi(\boldsymbol{x})$ to a higher dimensionality:

$$f(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w}. \tag{C.12}$$

For example, a feature space $\phi(x) = (1, x, x^2)^T$ for $d = 1$ would introduce a second order non-linearity. Using the mapping from equation C.12, the predictive posterior latent distribution from equation C.11 then becomes:

$$f_* \mid \boldsymbol{x}_*, \boldsymbol{y}, X, \boldsymbol{w} \sim \mathcal{N}(\frac{1}{\sigma_n^2}\phi(\boldsymbol{x}_*)^T A^{-1} \Phi \boldsymbol{y}, \phi(\boldsymbol{x}_*)^T A^{-1} \phi(\boldsymbol{x}_*)), \tag{C.13}$$

where $\Phi = [\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_n)]$ corresponds to the high-dimensionality mapping of X. However, in order to get arbitrarily complex functions, an infinity feature space would be needed. The above equations can be manipulated such that the feature space only appears in the form $\phi(\boldsymbol{x})\Sigma_p\phi(\boldsymbol{x'})$, where $\boldsymbol{x}$ and $\boldsymbol{x'}$ are either in the training set or the test set [166]. The "kernel trick" consists in noticing that this inner product can be equally computed through a kernel function k such that

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \phi(\boldsymbol{x})\Sigma_p\phi(\boldsymbol{x'}). \tag{C.14}$$

This formulation allows the definition of similarity between groups of points in an intuitive manner, as opposed to defining a feature space. Additionally, this also allows for the definition of arbitrarily complex feature vectors, without having to compute correspondingly complex feature vectors. As will be seen in the next section, different kernels can have powerful behavioural properties. For example, the *squared exponential* kernel can be shown to be equivalent to an infinite dimensional feature map.

With a prior distribution over the model parameters, the linear model presented

in this section is already a simple Gaussian process. However that Gaussian Process is degenerate, as it is constrained to be linear. This rigidity is the reason the entire model can be solved for with a fixed number of samples, even when using any finite feature mapping. The next section will formally define Gaussian Processes, and illustrate how their flexibility makes them ideal candidates for data-efficient optimization and learning.

Appendix D

Intellectual Property Re-use Permissions

D.1 IEEE Permissions

All images and textual content from IEEE publications re-used in this thesis have been referenced according to the IEEE guidelines listed on the IEEE website. In particular, for the re-use of substantial material from the author's own publications the following has been obeyed:
The IEEE does not require individuals working on a thesis to obtain a formal reuse license. If you are using the entire IEEE copyright owned article, the following IEEE copyright/ credit notice should be placed prominently in the references: ©[year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication].

D.2 Elsevier Permissions

The following notice on the Elsevier website allows re-use of the author's own publications:
Can I include/use my article in my thesis/dissertation? Yes. Authors can include their articles in full or in part in a thesis or dissertation for non-commercial purposes.

The following licenses were obtained for non co-authored Elsevier publications:

Title: Google Glass in pediatric surgery: An exploratory study
License number: 4431541445151

D.3 Springer Permissions

The following licenses were obtained from Springer for re-use:

Title: Visual Force Feedback for Hand-Held Microsurgical Instruments
License number: 4422141409714

Title: Augmented reality in the surgery of cerebral arteriovenous malformations: technique assessment and considerations
License number: 4431530941203

Title: An Integrated Visualization System for Surgical Planning and Guidance Using Image Fusion and Interventional Imaging
License number: 4431531300976

Title: pq-space Based Non-Photorealistic Rendering for Augmented Reality
License number: 4431860675426

Title: On-patient see-through augmented reality based on visual SLAM
License number: 4431531134428

Title: Gaze-Contingent Motor Channelling and Haptic Constraints for Minimally Invasive Robotic Surgery
License number: 4433011278961

Title: Image guidance and neuromonitoring in neurosurgery
License number: 4431530262312

Title: An effective visualisation and registration system for image-guided robotic partial nephrectomy
License number: 4431531033871

Title: Robust ultrasound probe tracking: initial clinical experiences during robot-assisted partial nephrectomy
License number: 4431530828953

Title: A Smart Haptic Hand-Held Device for Neurosurgical Microdissection
License number: 4437651190334

D.4 Permissions from Other Publishers

The following licenses were obtained from the mentioned publishers for re-use:

Publisher: Karger Publishers
Title: Easy-to-Use Augmented Reality Neuronavigation Using a Wireless Tablet PC
License number: 4431540672671

Publisher: Oxford University Press
Title: DEX-RAY: augmented reality neurosurgical navigation with a handheld video probe
License number: 4431550430594

Publisher: Annual Review of Biomedical Engineering
Title: Image-Guided Interventions: Technology Review and Clinical Applications
License number: 4431391213031

Publisher: Society for Translational Medicine
Title: Real-time 3D image reconstruction guidance in liver resection surgery
License number: 4431541310003

Publisher: John Wiley and Sons
Title: Image overlay navigation by markerless surface registration in gastrointestinal, hepatobiliary and pancreatic surgery
License number: 4431531489650

Bibliography

[1] B. Davies, "A review of robotics in surgery," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 214, no. 1, pp. 129–140, 2000.

[2] R. H. Taylor and D. Stoianovici, "Medical Robotics in Computer-Integrated Surgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 765–781, 2003.

[3] P. Pott, H. P. Scharf, and M. Schwarz, "Today's state of the art in surgical robotics," *Computer Aided Surgery*, vol. 10, no. 2, pp. 101–132, 2005.

[4] R. A. Beasley, "Medical Robots: Current Systems and Research Directions," *Journal of Robotics*, 2012.

[5] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel, V. J. Santos, and R. H. Taylor, "Medical robotics-Regulatory, ethical, and legal considerations for increasing levels of autonomy," *Science Robotics*, vol. 2, no. 4, 2017.

[6] G. R. Sutherland, P. B. McBeth, and D. F. Louw, "NeuroArm: An MR compatible robot for microsurgery," *International Congress Series*, vol. 1256, pp. 504–508, 2003.

[7] G. R. Sutherland, S. Wolfsberger, S. Lama, and K. Zarei-Nia, "The evolution of neuroArm," *Neurosurgery*, vol. 72, pp. 27–32, 2013.

[8] G. Guthart and J. Salisbury, "The Intuitive telesurgery system: overview and application," in *IEEE International Conference on Robotics and Automation.*, vol. 1, pp. 618–621, 2000.

[9] M. J. H. Lum, D. C. W. Friedman, G. Sankaranarayanan, H. King, K. Fodero, R. Leuschke, B. Hannaford, J. Rosen, and M. N. Sinanan, "The RAVEN: Design and validation of a telesurgery system," *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1183–1197, 2009.

[10] B. Hannaford, J. Rosen, D. W. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. N. Kosari, and L. White, "Raven-II: An open platform for surgical robotics research," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 954–959, 2013.

[11] D. Stephan, H. Sälzer, and F. Willeke, "First Experiences with the New Senhance® Telerobotic System in Visceral Surgery," *Visceral Medicine*, vol. 34, no. 1, pp. 31–36, 2018.

[12] S. Gueli Alletti, C. Rossitto, S. Cianci, E. Perrone, S. Pizzacalla, G. Monterossi, G. Vizzielli, S. Gidaro, and G. Scambia, "The Senhance™ surgical robotic system ("Senhance") for total hysterectomy in obese patients: a pilot study," *Journal of Robotic Surgery*, vol. 12, no. 2, pp. 229–234, 2018.

[13] H. Reichenspurner, R. J. Damiano, M. Mack, D. H. Boehm, H. Gulbins, C. Detter, B. Meiser, R. Ellgass, and B. Reichart, "Use of the voice-controlled and computer-assisted surgical system ZEUS for endoscopic coronary artery bypass grafting," *The Journal of Thoracic and Cardiovascular Surgery*, vol. 118, no. 1, pp. 11–16, 1999.

[14] "Monarch, Auris Health," *https://www.aurishealth.com/monarch-platform*, 2018.

[15] A. Dello Russo, G. Fassini, M. Casella, F. Bologna, O. Al-Nono, D. Colombo, V. Biagioli, P. Santangeli, L. Di Biase, M. Zucchetti, B. Majocchi, V. Marino, J. J. Gallinghouse, A. Natale, and C. Tondo, "Simultaneous assessment of contact pressure and local electrical coupling index using robotic navigation," *Journal of Interventional Cardiac Electrophysiology*, vol. 40, no. 1, pp. 23–31, 2014.

[16] T. Maier, G. Strauss, M. Hofer, T. Kraus, A. Runge, R. Stenzel, J. Gumprecht, T. Berger, A. Dietz, and T. C. Lueth, "A new micromanip-

ulator system for middle ear surgery," *IEEE International Conference on Robotics and Automation*, pp. 1568–1573, 2010.

[17] C. Neupert, S. Matich, N. Scherping, M. Kupnik, R. Werthschutzky, and C. Hatzfeld, "Pseudo-Haptic Feedback in Teleoperation," *IEEE Transactions on Haptics*, vol. 9, no. 3, pp. 397–408, 2016.

[18] P. Wisanuvej, G. Gras, K. Leibrandt, P. Giataganas, C. A. Seneci, J. Liu, and G.-Z. Yang, "Master manipulator designed for highly articulated robotic instruments in single access surgery," in *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 209–214, 2017.

[19] F. Carpi and C. Pappone, "Stereotaxis Niobe® magnetic navigation system for endocardial catheter ablation and gastrointestinal capsule endoscopy," *Expert Review of Medical Devices*, vol. 6, no. 5, pp. 487–498, 2009.

[20] R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, E. Dejuan, L. Kavoussi, R. Taylor, P. Jensen, L. Whitcomb, and A. Barnes, "A Steady-Hand Robotic System for Microsurgical Augmentation," *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1201–1210, 1999.

[21] A. Uneri, M. A. Balicki, J. Handa, P. Gehlbach, and R. H. Taylor, "New Steady-Hand Eye Robot with Micro-Force Sensing for Vitreoretinal Surgery," *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 814–819, 2010.

[22] M. D. De Smet, T. C. Meenink, T. Janssens, V. Vanheukelom, G. J. Naus, M. J. Beelen, C. Meers, B. Jonckx, and J. M. Stassen, "Robotic assisted cannulation of occluded retinal veins," *PLoS ONE*, vol. 11, no. 9, pp. 1–16, 2016.

[23] M. Jakopec, F. Rodriguez y Baena, S. J. Harris, P. Gomes, J. Cobb, and B. L. Davies, "The Hands-On Orthopaedic Robot "Acrobot": Early Clinical Trials of Total Knee Replacement Surgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 902–911, 2003.

[24] J. Pransky, "ROBODOC – surgical robot success story," *Industrial Robot: An International Journal*, vol. 24, no. 3, pp. 231–233, 1997.

[25] W. L. Bargar, A. Bauer, and M. Borner, "Primary and revision total hip replacement using the Robodoc system.," *Clinical orthopaedics and related research*, no. 354, pp. 82–91, 1998.

[26] J. R. Adler Jr, S. D. Chang, M. J. Murphy, J. Doty, P. Geis, and S. L. Hancock, "The Cyberknife: a frameless robotic system for radiosurgery," *Stereotactic and functional neurosurgery*, vol. 69, pp. 124–128, 1997.

[27] W. Kilby, J. R. Dooley, G. Kuduvalli, S. Sayeh, and C. R. Maurer, "The CyberKnife® robotic radiosurgery system in 2010," *Technology in Cancer Research and Treatment*, vol. 9, no. 5, pp. 433–452, 2010.

[28] L. Zhang, N. Zhou, and S. Wang, "Direct manipulation of tool-likemasters for controlling a master–slave surgical robotic system," *The international journal of medical robotics amd computer assisted surgery : MRCAS*, vol. 10, no. 4, pp. 427–437, 2014.

[29] P. Berthet-Rayne, G. Gras, K. Leibrandt, P. Wisanuvej, A. Schmitz, C. A. Seneci, and G. Z. Yang, "The i2Snake Robotic Platform for Endoscopic Surgery," *Annals of Biomedical Engineering*, pp. 1–13, 2018.

[30] P. Berthet-Rayne, K. Leibrandt, G. Gras, P. Fraisse, A. Crosnier, and G.-Z. Yang, "Inverse Kinematics Control Methods for Redundant Snake-Like Robot Teleoperation during Minimally Invasive Surgery," *IEEE Robotics and Automation Letters*, 2018.

[31] Z. M. Thant, S. C. Low, S. W. Tang, L. Phee, K. Y. Ho, and S. C. Chung, "Ergonomic master controller for flexible endoscopic gastrointestinal robot manipulator," *International Conference on Biomedical and Pharmaceutical Engineering*, pp. 575–579, 2006.

[32] T. Travaglini, P. Swaney, H. D. Weaver, and R. Webster III, "Initial Experiments with the Leap Motion as a User Interface in Robotic Endonasal Surgery," *Robotics and Mechatronics*, pp. 171–179, 2016.

[33] R. Wen, W. L. Tay, B. P. Nguyen, C. B. Chng, and C. K. Chui, "Hand gesture guided robot-assisted surgery based on a direct augmented reality interface," *Computer Methods and Programs in Biomedicine*, vol. 116, no. 2, pp. 68–80, 2014.

[34] M. G. Jacob, Y. T. Li, and J. P. Wachs, "Surgical instrument handling and retrieval in the operating room with a multimodal robotic assistant," *IEEE International Conference on Robotics and Automation*, pp. 2140–2145, 2013.

[35] T. Kawai, M. Fukunishi, A. Nishikawa, Y. Nishizawa, and T. Nakamura, "Hands-free interface for surgical procedures based on foot movement patterns," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 345–348, 2014.

[36] S. Voros, G. P. Haber, J. F. Menudet, J. A. Long, and P. Cinquin, "ViKY robotic scope holder: Initial clinical experience and preliminary results using instrument tracking," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 6, pp. 879–886, 2010.

[37] S. Zhai, *Human Performance in Six Degree of Freedom Input Control*. PhD thesis, 1996.

[38] L. Dominjon, A. Lecuyer, G. Andrade-barroso, J.-m. Burkhardt, and S. Richir, "The "Bubble" Technique : Interacting with Large Virtual Environments Using Haptic Devices with Limited Workspace," *Eurohaptics Conference, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2005.

[39] G. G. Casiez, D. Vogel, Q. Pan, and C. Chaillou, "RubberEdge: reducing clutching by combining position and rate control with elastic feedback," *ACM symposium on User interface software and technology*, pp. 129–138, 2007.

[40] T. Yamada, Y. Hayamizu, Y. Yamamoto, Y. Yomogida, A. Izadi-Najafabadi, D. N. Futaba, and K. Hata, "A stretchable carbon nanotube strain sensor for human-motion detection," *Nature Nanotechnology*, vol. 6, no. 5, pp. 296–301, 2011.

[41] C. K. Chui, J. S. Ong, Z. Y. Lian, Z. Wang, J. Teo, J. Zhang, C. H. Yan, S. H. Ong, S. C. Wang, H. K. Wong, C. L. Teo, and S. H. Teoh, "Haptics in computer-mediated simulation: Training in vertebroplasty surgery," *Simulation and Gaming*, vol. 37, no. 4, pp. 438–451, 2006.

[42] J. A. Sanchez-Margallo and J. B. Pagador, "Ergonomic Assessment of Hand Movements in Laparoscopic Surgery Using the CyberGlove," *Computational Biomechanics for Medicine*, pp. 121–128, 2010.

[43] "HaptX VR," *https://haptx.com/*, 2018.

[44] "Manus VR," *https://manus-vr.odoo.com/*, 2018.

[45] D. Whitney, E. Rosen, D. Ullman, E. Phillips, and S. Tellex, "ROS Reality : A Virtual Reality Framework Using Consumer-Grade Hardware for ROS Reality : A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots,"

[46] H. G. Hoffman, W. J. Meyer, M. Ramirez, L. Roberts, E. J. Seibel, B. Atzori, S. R. Sharar, and D. R. Patterson, "Feasibility of Articulated Arm Mounted Oculus Rift Virtual Reality Goggles for Adjunctive Pain Control During Occupational Therapy in Pediatric Burn Patients," *Cyberpsychology, Behavior, and Social Networking*, vol. 17, no. 6, pp. 397–401, 2014.

[47] K. Leibrandt, G. Gras, P. Wisanuvej, C. Seneci, and J. Shang, "A Flexible Robot for Working in Tight Spaces," *Hamlyn Symposium Robotic Challenge*, 2015.

[48] M. Aiple and A. Schiele, "Pushing the limits of the CyberGrasp for haptic rendering," *IEEE International Conference on Robotics and Automation*, pp. 3541–3546, 2013.

[49] A. Alamri, R. Iglesias, M. Eid, A. El Saddik, S. Shirmohammadi, and E. Lemaire, "Haptic exercises for measuring improvement of post-stroke rehabilitation patients," *IEEE International Workshop on Medical Measurement and Applications*, 2007.

[50] Y. Kim, P. C. Kim, R. Selle, A. Shademan, and A. Krieger, "Experimental evaluation of contact-less hand tracking systems for tele-operation of surgical tasks," *IEEE International Conference on Robotics and Automation*, pp. 3502–3509, 2014.

[51] B. M. Kraft, C. Jager, K. Kraft, B. J. Leibl, and R. Bittner, "The AESOP robot system in laparoscopic surgery: Increased risk or advantage for surgeon and patient?," *Surgical Endoscopy and Other Interventional Techniques*, vol. 18, no. 8, pp. 1216–1223, 2004.

[52] B. Mitchell, J. Koo, I. Iordachita, P. Kazanzides, A. Kapoor, J. Handa, G. Hager, and R. Taylor, "Development and application of a new steady-hand nanipulator for retinal surgery," *IEEE International Conference on Robotics and Automation*, pp. 623–629, 2007.

[53] P. Kazanzides, G. Fichtinger, G. D. Hager, A. M. Okamura, L. L. Whitcomb, and R. H. Taylor, "Surgical and interventional robotics-core concepts, technology, and design," *IEEE Robotics and Automation Magazine*, vol. 15, no. 2, pp. 122–130, 2008.

[54] B. Davies, K. Fan, R. Hibberd, M. Jakopec, and S. Harris, "ACROBOT - using robots and surgeons synergistically in knee surgery," *8th International Conference on Advanced Robotics. Proceedings. ICAR'97*, pp. 173 – 178, 1997.

[55] G.-Z. Yang, L. Dempere-Marco, X. P. Hu, and A. Rowe, "Visual search: Psychophysical models and practical applications," *Image and Vision Computing*, vol. 20, no. 4, pp. 291–305, 2002.

[56] S. H. Fairclough, *Advances in Physiological Computing*. Springer, humancomp ed., 2014.

[57] D. A. Robinson, "A Method of Measuring Eye Movemnent Using a Scieral Search Coil in a Magnetic Field," *IEEE Transactions on Bio-medical Electronics*, vol. 10, no. 4, pp. 137–145, 1963.

[58] G. P. Mylonas, A. Darzi, and G.-Z. Yang, "Gaze-contingent control for minimally invasive robotic surgery," *Computer Aided Surgery*, vol. 11, no. 5, pp. 256–266, 2006.

[59] I. Tong, O. Mohareri, S. Tatasurya, C. Hennessey, and S. Salcudean, "A retrofit eye gaze tracker for the da Vinci and its integration in task execution using the da Vinci Research Kit," *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 2043–2050, 2015.

[60] S. Li and X. Zhang, "Implicit Intention Communication in Human-Robot Interaction Through Visual Behavior Studies," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 437–448, 2017.

[61] M.-Y. Wang, A. A. Kogkas, A. Darzi, and G. P. Mylonas, "Free-View, 3D Gaze-Guided, Assistive Robotic System for Activities of Daily Living," *IEEE/RJS International Conference on Intelligent Robots and Systems*, 2018.

[62] A. Tapus, M. Mataric, and B. Scassellati, "Socially assistive robotics: The grand challenges in helping humans through social interaction," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 35–42, 2007.

[63] M. a. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends® in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[64] S. Najarian, M. Fallahnezhad, and E. Afshari, "Advances in medical robotic systems with specific applications in surgery-a review," *Journal of Medical Engineering and Technology*, vol. 35, no. 1, pp. 19–33, 2011.

[65] N. Hogan and H. Krebs, "MIT-MANUS: a workstation for manual therapy and training. I," *International Workshop on Robot and Human Communication*, pp. 161–165, 1992.

[66] M. Bouri and Y. Stauffer, "The WalkTrainer: a robotic system for walking rehabilitation," *International Conference on Robotics and Biomimetics, Kunming, China*, pp. 1616–1621, 2006.

[67] G. Gras, V. Vitiello, and G.-z. Yang, "Cooperative Control of a Compliant Manipulator for Robotic-Assisted Physiotherapy," *IEEE International Conference on Robotics & Automation*, pp. 339–346, 2014.

[68] S. Li, X. Zhang, and J. D. Webb, "3D-Gaze-Based Robotic Grasping Through Mimicking Human Visuomotor Function for People with Motion Impairments," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 12, pp. 2824–2835, 2017.

[69] A. A. Kogkas, A. Darzi, and G. P. Mylonas, "Gaze-contingent perceptually enabled interactions in the operating theatre," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 7, pp. 1131–1140, 2017.

[70] M. Unger, D. Black, N. M. Fischer, T. Neumuth, and B. Glaser, "Design and evaluation of an eye tracking support system for the scrub nurse," *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2018.

[71] A. A. Kogkas, A. Darzi, and G. P. Mylonas, "Gaze-Driven Human-Robot Interaction in the Operating Theatre," *6th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery (CRAS)*, p. 103, 2016.

[72] B. Zheng, X. Jiang, G. Tien, A. Meneghetti, O. N. M. Panton, and M. S. Atkins, "Workload assessment of surgeons: Correlation between NASA TLX and blinks," *Surgical Endoscopy*, vol. 26, no. 10, pp. 2746–2750, 2012.

[73] C. C. Nduka, P. A. Super, J. Monson, and A. W. Darzi, "Cause and prevention of electrosurgical injuries in laparoscopy," *Journal of the American College of Surgeons*, vol. 179, no. 2, pp. 161–170, 1994.

[74] S. S. Kommu, P. Rimington, C. Anderson, and A. Rane, "Initial experience with the EndoAssist camera-holding robot in laparoscopic urological surgery," *Journal of Robotic Surgery*, vol. 1, no. 2, pp. 133–137, 2007.

[75] K. Sriskandarajah, S. Gillen, A. D. Marco, M. Sodegren, D. R. C. James, and J. Clark, "Robotic tele-manipulating devices for laparoscopy improve

surgical performance in simulated porcine laparoscopic cholecystectomies on the ELITE simulator.," *Proceedings of the Hamlyn Symposium on Medical Robotics*, pp. 36–37, 2012.

[76] D. P. Noonan, G. P. Mylonas, J. Shang, C. J. Payne, A. Darzi, and G. Z. Yang, "Gaze contingent control for an articulated mechatronic laparoscope," *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, BioRob*, pp. 759–764, 2010.

[77] Y. Cao, Q. Liu, Y. Kobayashi, K. Kawamura, S. Sugano, and M. G. Fujie, "Development of an endoscopic manipulator control system with intention recognition based on pupil movement," *IEEE International Conference on Robotics and Biomimetics*, pp. 1637–1642, 2015.

[78] Y. Cao, S. Miura, Q. Liu, Y. Kobayashi, K. Kawamura, S. Sugano, and M. G. Fujie, "An image processing method for changing endoscope direction based on pupil movement," *ROBOMECH Journal*, vol. 3, no. 1, 2016.

[79] Y. Cao, S. Miura, Y. Kobayashi, K. Kawamura, S. Sugano, and M. G. Fujie, "Pupil Variation Applied to the Control of an Endoscope," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 531–538, 2016.

[80] D. D. Salvucci and J. H. Goldberg, "Identifying Fixations and Saccades in Eye-Tracking Protocols," *Proceedings of the Eye Tracking Research and Applications Symposium*, pp. 71–78, 2000.

[81] P. Blignaut, "Fixation identification: The optimum threshold for a dispersion algorithm.," *Attention, perception & psychophysics*, vol. 71, no. 4, pp. 881–895, 2009.

[82] V. Krassanakis, V. Filippakopoulou, and B. Nakos, "EyeMMV toolbox: An eye movement post-analysis tool based on a two-step spatial dispersion threshold for fixation identification," *Journal of Eye Movement Research*, vol. 7(1), no. 1, pp. 1–10, 2014.

[83] S. Li, X. Zhang, F. J. Kim, R. Donalisio da Silva, D. Gustafson, and W. R. Molina, "Attention-Aware Robotic Laparoscope Based on Fuzzy Interpre-

tation of Eye-Gaze Patterns," *Journal of Medical Devices*, vol. 9, no. 4, 2015.

[84] S. Li, J. Zhang, L. Xue, F. J. Kim, and X. Zhang, "Attention-aware robotic laparoscope for human-robot cooperative surgery," *IEEE International Conference on Robotics and Biomimetics*, pp. 792–797, 2013.

[85] C. Staub, S. Can, B. Jensen, A. Knoll, and S. Kohlbecher, "Human-computer interfaces for interaction with surgical tools in robotic surgery," *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 81–86, 2012.

[86] D. P. Noonan, G. P. Mylonas, A. Darzi, and Guang-Zhong Yang, "Gaze contingent articulated robot control for robot assisted minimally invasive surgery," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1186–1191, 2008.

[87] G. P. Mylonas, K.-W. Kwok, A. Darzi, and G.-Z. Yang, "Gaze-contingent motor channelling and haptic constraints for minimally invasive robotic surgery," 2008.

[88] Z. Li, I. Tong, L. Metcalf, C. Hennessey, and S. E. Salcudean, "Free Head Movement Eye Gaze Contingent Ultrasound Interfaces for the da Vinci Surgical System," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2137–2143, 2018.

[89] K. Fujii, A. Salerno, K. Sriskandarajah, K. W. Kwok, K. Shetty, and G. Z. Yang, "Gaze contingent cartesian control of a robotic arm for laparoscopic surgery," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 3582–3589, 2013.

[90] R. J. K. Jacob, "The use of eye movements in human-computer interaction techniques: what you look at is what you get," *ACM Transactions on Information Systems*, vol. 9, no. 2, pp. 152–169, 1991.

[91] S. Zhai, C. Morimoto, and S. Ihde, "Manual and gaze input cascaded (MAGIC) pointing," *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 246–253, 1999.

[92] G. Z. Yang, G. P. Mylonas, K. W. Kwok, and A. Chung, "Perceptual docking for robotic control," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5128 LNCS, pp. 21–30, 2008.

[93] K. Fujii, *Gaze contingent robotic control in minimally invasive surgery*. PhD thesis, 2014.

[94] K. Fujii, G. Gras, A. Salerno, and G.-Z. Yang, "Gaze gesture based human robot interaction for laparoscopic surgery," *Medical Image Analysis*, vol. 44, 2018.

[95] H. Drewes, A. De Luca, and A. Schmidt, "Eye-gaze interaction for mobile phones," *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, vol. 07, pp. 364–371, 2007.

[96] E. Mollenbach, J. P. Hansen, M. Lillholm, and A. G. Gale, "Single stroke gaze gestures," *Extended abstracts on Human factors in computing systems*, pp. 4555–4560, 2009.

[97] D. Rozado, F. B. Rodriguez, and P. Varona, "Low cost remote gaze gesture recognition in real time," *Applied Soft Computing Journal*, vol. 12, no. 8, pp. 2072–2084, 2012.

[98] J. O. Wobbrock, J. Rubinstein, M. W. Sawyer, and A. T. Duchowski, "Longitudinal evaluation of discrete consecutive gaze gestures for text entry," *Symposium on Eye tracking research & applications*, pp. 11–18, 2008.

[99] D. Rozado, T. Moreno, J. San Agustin, F. B. Rodriguez, and P. Varona, "Controlling a smartphone using gaze gestures as the input mechanism," *Human-Computer Interaction*, vol. 30, no. 1, pp. 34–63, 2013.

[100] H. Istance, A. Hyrskykari, L. Immonen, S. Mansikkamaa, and S. Vickers, "Designing gaze gestures for gaming," *Symposium on Eye-Tracking Research & Applications*, pp. 323–330, 2010.

[101] K. Cleary and T. M. Peters, "Image-guided interventions: Technology and applications," *Annual Review of Biomedical Engineering*, vol. 12, pp. 119–142, 2010.

[102] J. Gong, G. Mohr, and J. L. Vezina, "Endoscopic pituitary surgery with and without image guidance: an experimental comparison," *Surgical Neurology*, vol. 67, no. 6, pp. 572–578, 2007.

[103] W. H. Ng, K. Mukhida, and J. T. Rutka, "Image guidance and neuromonitoring in neurosurgery," *Child's Nervous System*, vol. 26, no. 4, pp. 491–502, 2010.

[104] T. Peters, B. Davey, P. Munger, R. Comeau, A. Evans, and A. Olivier, "Three-dimensional multimodal image-guidance for neurosurgery," *IEEE Transactions on Medical Imaging*, vol. 15, no. 2, pp. 121–128, 1996.

[105] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," *ACM Siggraph Computer Graphics*, vol. 22, no. 4, pp. 65–74, 1988.

[106] G. Kindlmann and J. W. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," *IEEE Symposium on Volume Visualization*, pp. 79–86, 1998.

[107] J. Kniss, S. Member, and I. C. Society, "Multidimensional transfer functions for interactive volume rendering," *Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 270 – 285, 2002.

[108] L. Soler, S. Nicolau, P. Pessaux, D. Mutter, and J. Marescaux, "Real-time 3D image reconstruction guidance in liver resection surgery.," *Hepatobiliary surgery and nutrition*, vol. 3, no. 2, pp. 73–81, 2014.

[109] D. T. Gering, A. Nabavi, R. Kikinis, W. E. L. Grimson, N. Hata, F. A. Jolesz, and W. M. Wells, "An integrated visualization system for surgical planning and guidance using image fusion and an open MR," *We present a software package which uniquely integrates sev- eral facets of image-guided medicine into a single portable, extendable environment. It provides capabilities for automatic registration, semi- automatic segmentation, 3D surface model generatio*, pp. 808–819, 1999.

[110] S. De Buck, F. Maes, J. Ector, J. Bogaert, S. Dymarkowski, H. Heidbüchel, and P. Suetens, "An augmented reality system for patient-specific guidance of cardiac catheter ablation procedures," *IEEE Transactions on Medical Imaging*, vol. 24, no. 11, pp. 1512–1524, 2005.

[111] S. Bernhardt, S. A. Nicolau, L. Soler, and C. Doignon, "The status of augmented reality in laparoscopic surgery as of 2016," *Medical Image Analysis*, vol. 37, pp. 66–90, 2017.

[112] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "A survey of augmented reality," *Computer Graphics and Applications*, vol. 6, pp. 34–47, 2001.

[113] W. Deng, F. Li, M. Wang, and Z. Song, "Easy-to-use augmented reality neuronavigation using a wireless tablet PC," *Stereotactic and Functional Neurosurgery*, vol. 92, no. 1, pp. 17–24, 2014.

[114] N. Mahmoud, O. G. Grasa, S. A. Nicolau, C. Doignon, L. Soler, J. Marescaux, and J. M. Montiel, "On-patient see-through augmented reality based on visual SLAM," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 1, pp. 1–11, 2017.

[115] C. Bichlmeier, S. M. Heining, M. Feuerstein, and N. Navab, "The virtual mirror: a new interaction paradigm for augmented reality environments.," *IEEE transactions on medical imaging*, vol. 28, no. 9, pp. 1498–1510, 2009.

[116] S. Vogt, A. Khamene, and F. Sauer, "Reality augmentation for medical procedures: System architecture, single camera marker tracking, and system evaluation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 179–190, 2006.

[117] M. Aschke, C. Wirtz, J. Raczkowsky, H. Worn, and S. Kunze, "Augmented reality in operating microscopes for neurosurgical interventions," *First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings.*, pp. 652–655, 2003.

[118] I. Cabrilo, P. Bijlenga, and K. Schaller, "Augmented reality in the surgery of cerebral arteriovenous malformations: technique assessment and considerations," *Acta Neurochirurgica*, vol. 156, no. July, pp. 1769–1774, 2014.

[119] I. Cabrilo, P. Bijlenga, and K. Schaller, "Augmented reality in the surgery of cerebral aneurysms: A technical report," *Operative Neurosurgery*, vol. 10, no. 2, pp. 252–261, 2014.

[120] M. Sugimoto, H. Yasuda, K. Koda, M. Suzuki, M. Yamazaki, T. Tezuka, C. Kosugi, R. Higuchi, Y. Watayo, Y. Yagawa, S. Uemura, H. Tsuchiya, and T. Azuma, "Image overlay navigation by markerless surface registration in gastrointestinal, hepatobiliary and pancreatic surgery," *Journal of Hepato-Biliary-Pancreatic Sciences*, vol. 17, no. 5, pp. 629–636, 2010.

[121] R. Wen, L. Yang, C.-K. Chui, K.-B. Lim, and S. Chang, "Intraoperative visual guidance and control interface for augmented reality robotic surgery," *IEEE International Conference on Control and Automation (ICCA)*, pp. 947–952, 2010.

[122] A. Osorio, J.-A. Galan, J. Nauroy, S. Dahdouh, J.-J. Lobato, P. Donars, and I. Navarro, "Real time planning, guidance and validation of surgical acts using 3D segmentations, augmented reality projections and surgical tools video tracking," *Medical Imaging 2010: Visualization, Image-Guided Procedures, and Modeling*, 2010.

[123] O. J. Muensterer, M. Lacher, C. Zoeller, M. Bronstein, and J. Kübler, "Google Glass in pediatric surgery: An exploratory study," *International Journal of Surgery*, vol. 12, no. 4, pp. 281–289, 2014.

[124] P. Pratt, M. Ives, G. Lawton, J. Simmons, N. Radev, L. Spyropoulou, and D. Amiras, "Through the HoloLens looking glass: augmented reality for extremity reconstruction surgery using 3D vascular models with perforating vessels," *European Radiology Experimental*, vol. 2, no. 1, p. 2, 2018.

[125] I. Kuhlemann, M. Kleemann, P. Jauer, A. Schweikard, and F. Ernst, "Towards X-ray free endovascular interventions – using HoloLens for on-line holographic visualisation," *Healthcare Technology Letters*, vol. 4, no. 5, pp. 184–187, 2017.

[126] N. Cui, P. Kharel, and V. Gruev, "Augmented reality with Microsoft HoloLens holograms for near infrared fluorescence based image guided surgery," *Molecular-Guided Surgery: Molecules, Devices, and Applications III*, 2017.

[127] D. G. Armstrong, T. M. Rankin, N. A. Giovinco, J. L. Mills, and Y. Matsuoka, "A heads-up display for diabetic limb salvage surgery: A view through the google looking glass," *Journal of Diabetes Science and Technology*, vol. 8, no. 5, pp. 951–956, 2014.

[128] P. Pratt, A. Jaeger, A. Hughes-Hallett, E. Mayer, J. Vale, A. Darzi, T. Peters, and G.-Z. Yang, "Robust ultrasound probe tracking: initial clinical experiences during robot-assisted partial nephrectomy," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 12, pp. 1905–1913, 2015.

[129] R. a. Kockro, Y. T. Tsai, I. Ng, P. Hwang, C. Zhu, K. Agusanto, L. X. Hong, and L. Serra, "DEX-Ray: Augmented reality neurosurgical navigation with a handheld video probe," *Neurosurgery*, vol. 65, no. 4, pp. 795–807, 2009.

[130] L. Johnson, P. Edwards, L. Griffin, and D. Hawkes, "Depth perception of stereo overlays in image-guided surgery," *Medical Imaging 2004: Image Perception, Observer Performance, and Technology Assessment*, 2004.

[131] H. J. Marcus, P. Pratt, A. Hughes-Hallett, T. P. Cundy, A. P. Marcus, G.-z. Yang, and D. Nandi, "Comparative Effectiveness and Safety of Image Guidance Systems in Neurosurgery: A Preclinical Randomized Study," *Journal of Neurosurgery*, 2015.

[132] T. Sielhorst, C. Bichlmeier, S. M. Heining, and N. Navab, "Depth perception–a major issue in medical AR: evaluation study by twenty surgeons.," *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 9, pp. 364–372, 2006.

[133] C. Bichlmeier, T. Sielhorst, S. M. Heining, and N. Navab, "Improving depth perception in medical AR a virtual vision panel to the inside of the patient," *Bildverarbeitung fur die Medizin*, pp. 217–221, 2007.

[134] J. E. Cutting and P. M. Vishton, "Perceiving Layout and Knowing Distances : The Integration, Relative Potency, and Contextual Use of Different Information about Depth," *Perception of Space and Motion*, pp. 69–117, 1995.

[135] C. Bichlmeier, F. Wimmer, S. M. Heining, and N. Navab, "Contextual anatomic mimesis: Hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality," *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*, pp. 129–138, 2007.

[136] O. Kutter, A. Aichert, C. Bilchmeier, J. Traub, S. M. Heining, B. Ockert, E. Euler, and N. Navab, "Real-time Volume Rendering for High Quality Visualization in Augmented Reality," *International Workshop on Augmented environments for Medical Imaging including Augmented Reality in Computer-aided Surgery*, 2008.

[137] M. Lerotic, A. J. Chung, G. Mylonas, and G.-Z. Yang, "Pq-space based non-photorealistic rendering for augmented reality.," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 10, pp. 102–109, 2007.

[138] P. Pratt, E. Mayer, J. Vale, D. Cohen, E. Edwards, A. Darzi, and G. Z. Yang, "An effective visualisation and registration system for image-guided robotic partial nephrectomy," *Journal of Robotic Surgery*, vol. 6, no. 1, pp. 23–31, 2012.

[139] B. J. Dixon, M. J. Daly, H. Chan, A. D. Vescan, I. J. Witterick, and J. C. Irish, "Surgeons blinded by enhanced navigation: The effect of augmented reality on attention," *Surgical Endoscopy*, vol. 27, no. 2, pp. 454–461, 2013.

[140] N. Navab, C. Hennersperger, B. Frisch, and B. Fürst, "Personalized, relevance-based Multimodal Robotic Imaging and augmented reality for Computer Assisted Interventions," *Medical Image Analysis*, vol. 33, pp. 64–71, 2016.

[141] D. Katic, A. L. Wekerle, J. Gortler, P. Spengler, S. Bodenstedt, S. Rohl, S. Suwelack, H. G. Kenngott, M. Wagner, B. P. Muller-Stich, R. Dillmann, and S. Speidel, "Context-aware Augmented Reality in laparoscopic surgery," *Computerized Medical Imaging and Graphics*, vol. 37, no. 2, pp. 174–182, 2013.

[142] Colon Cancer Laparoscopic or Open Resection Study Group, "Laparoscopic surgery versus open surgery for colon cancer: Short-term outcomes of a randomised trial," *Lancet Oncology*, vol. 6, no. 7, pp. 477–484, 2005.

[143] B. Zheng, E. Rieder, M. A. Cassera, D. V. Martinec, G. Lee, O. N. M. Panton, A. Park, and L. L. Swanström, "Quantifying mental workloads of surgeons performing natural orifice transluminal endoscopic surgery (NOTES) procedures," *Surgical Endoscopy*, vol. 26, no. 5, pp. 1352–1358, 2012.

[144] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," *European Conference on Computer Vision*, pp. 756–771, 2014.

[145] R. Y. Tsai and R. K. Lenz, "Real time versatile robotics hand/eye calibration using 3D machine vision," *IEEE International Conference on Robotics and Automation*, pp. 554–561, 1988.

[146] Y. C. Shiu and S. Ahmad, "Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form AX = XB," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.

[147] K. Daniilidis, "Hand-Eye Calibration Using Dual Quaternions," *The International Journal of Robotics Research*, vol. 18, pp. 286–298, mar 1999.

[148] A. Malti and J. P. Barreto, "Robust hand-eye calibration for computer aided medical endoscopy," in *IEEE International Conference on Robotics and Automation*, pp. 5543–5549, Ieee, may 2010.

[149] G. Bradski and A. Kaehler, "OpenCV," *Journal of software tools*, 2000.

[150] C. Fu, P. Zhang, J. Jiang, K. Yang, and Z. Lv, "A Bayesian approach for sleep and wake classification based on dynamic time warping method," *Multimedia Tools and Applications*, vol. 76, no. 17, pp. 17765–17784, 2017.

[151] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum Robots for Medical Applications: A Survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[152] V. Vitiello, S.-L. Lee, T. P. Cundy, and G.-Z. Yang, "Emerging robotic platforms for minimally invasive surgery," *IEEE reviews in biomedical engineering*, vol. 6, pp. 111–126, 2013.

[153] A. Bajo, R. E. Goldman, L. Wang, D. Fowler, and N. Simaan, "Integration and preliminary evaluation of an Insertable Robotic Effectors Platform for Single Port Access Surgery," *IEEE International Conference on Robotics and Automation*, pp. 3381–3387, 2012.

[154] H. J. Marcus, A. Hughes-Hallett, T. P. Cundy, G. Z. Yang, A. Darzi, and D. Nandi, "da Vinci robot-assisted keyhole neurosurgery: a cadaver study on feasibility and safety," *Neurosurgical Review*, vol. 38, no. 2, pp. 367–371, 2015.

[155] H. Han, J. Lee, and J. Park, "A continuous task transition algorithm for operational space control framework," *Ubiquitous Robots and Ambient Intelligence*, pp. 148–152, 2012.

[156] F. Chaumette and Marchand Éric, "A redundancy-based iterative approach for avoiding joint limits: Application to visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 719–730, 2001.

[157] A. Kapoor, N. Simaan, and R. Taylor, "Suturing in Confined Spaces: Constrained Motion Control of a Hybrid 8-DOF Robot," *International Conference on Advanced Robotics*, pp. 452–459, 2005.

[158] J. Lee, J. Kim, K. K. Lee, S. Hyung, Y. J. Kim, W. Kwon, K. Roh, and J. Y. Choi, "Modeling and control of robotic surgical platform for single-port

access surgery," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3489–3495, 2014.

[159] K. Leibrandt, P. Wisanuvej, G. Gras, J. Shang, C. A. Seneci, P. Giataganas, V. Vitiello, A. Darzi, and G.-Z. Yang, "Effective manipulation in confined spaces of highly articulated robotic instruments for single access surgery," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1704–1711, 2017.

[160] J. Shang, K. Leibrandt, P. Giataganas, V. Vitiello, C. A. Seneci, P. Wisanuvej, J. Liu, G. Gras, J. Clark, A. Darzi, and G.-Z. Yang, "A single-port robotic system for transanal micro-surgery – design and validation," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1510–1517, 2017.

[161] C. A. Seneci, J. Shang, A. Darzi, and G.-Z. Yang, "Rapid manufacturing with selective laser melting for robotic surgical tools: Design and process considerations," *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 824–830, 2015.

[162] M. J. Tsai and Y. H. Chiou, "Manipulability of manipulators," *Mechanism and Machine Theory*, vol. 25, no. 5, pp. 575–585, 1990.

[163] M. Ye, L. Zhang, S. Giannarou, and G. Z. Yang, "Real-time 3D tracking of articulated tools for robotic surgery," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 386–394, 2016.

[164] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," *International Conference on Pattern Recognition*, pp. 2756–2759, 2010.

[165] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the Human Out of the Loop : A Review of Bayesian Optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[166] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[167] E. Brochu, V. M. Cora, N. D. Freitas, and N. de Freitas, "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to

Active User Modeling and Hierarchical Reinforcement Learning," pp. 1–5, 2010.

[168] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "An experimental comparison of Bayesian optimization for bipedal locomotion," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1951–1958, 2014.

[169] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 512, no. 7553, pp. 503–507, 2015.

[170] J. Vogel and N. De Freitas, "Target-directed attention: Sequential decision-making for gaze planning," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2372–2379, 2008.

[171] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot," *Autonomous Robots*, vol. 27, no. 2, pp. 93–103, 2009.

[172] T. Matsubara, Y. Funaki, M. Ding, T. Ogasawara, and K. Sugimoto, "Data-efficient human training of a care motion controller for human transfer assistant robots using Bayesian optimization," *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 606–611, 2016.

[173] W. Chu and Z. Ghahramani, "Preference Learning with Gaussian Processes," *Proceedings of the International Conference on Machine Learning, Bonn, Germany*, pp. 137–144, 2005.

[174] E. Brochu, M. W. Hoffman, and N. de Freitas, "Portfolio Allocation for Bayesian Optimization," *arXiv:1009.5419*, 2010.

[175] G. Gras, K. Leibrandt, P. Wisanuvej, P. Giataganas, C. A. Seneci, M. Ye, J. Shang, and G.-Z. Yang, "Implicit gaze-assisted adaptive motion scaling for highly articulated instrument manipulation," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4233–4239, 2017.

[176] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[177] J. M. Gablonsky and C. T. Kelley, "A Locally-Biased form of the DIRECT Algorithm," *Journal of Global Optimization*, vol. 21, no. 1, pp. 27–37, 2001.

[178] M. J. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," *Cambridge NA Report NA2009/06*, pp. 26–46, 2009.

[179] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization.," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[180] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients.," *The computer journal*, vol. 7, no. 2, pp. 149–154, 1964.

[181] S. G. Johnson, "The NLopt nonlinear-optimization package.," 2014.

[182] A. D. Bull, "Convergence Rates of Efficient Global Optimization Algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2879–2904, 2011.

[183] R. Benassi, J. Bect, and E. Vazquez, "Robust Gaussian process-based global optimization using a fully Bayesian expected improvement criterion," *International Conference on Learning and Intelligent Optimization*, pp. 176–190, 2011.

[184] C. J. Payne and G. Z. Yang, "Hand-held medical robots," *Annals of Biomedical Engineering*, vol. 42, no. 8, pp. 1594–1605, 2014.

[185] G. Stetten, B. Wu, R. Klatzky, J. Galeotti, M. Siegel, R. Lee, F. Mah, A. Eller, J. Schuman, and R. Hollis, "Hand-held force magnifier for surgical instruments," *IPCAI*, pp. 90–100, 2011.

[186] R. Lee, B. Wu, R. Klatzky, V. Shivaprabhu, J. Galeotti, S. Horvath, M. Siegel, J. S. Schuman, R. Hollis, and G. Stetten, "Hand-held force

magnifier for surgical instruments: Evolution toward a clinical device," *Augmented Environments for Computer-Assisted Interventions*, vol. 7815, pp. 77–89, 2013.

[187] C. J. Payne, H. J. Marcus, and G.-Z. Yang, "A Smart Haptic Hand-Held Device for Neurosurgical Microdissection," *Annals of Biomedical Engineering*, 2015.

[188] C. J. Payne, G. Gras, M. Hughes, D. Nathwani, and G.-Z. Yang, "A hand-held flexible mechatronic device for arthroscopy," in *IEEE/RJS International Conference on Intelligent Robots and Systems*, pp. 817–823, 2015.

[189] E. Vander Poorten, L. Esteveny, A. Gijbels, B. Rosa, L. Schoevaerdts, K. Willekens, P. Stalmans, D. Reynaerts, G. Naus, M. Beelen, N. D. Jonge, G. Kronreif, Y. Douven, N. Dorrestijn, L. V. Gool, G. Szekely, E. Lankenau, M. Krug, P. Koch, G. Huttmann, A. Gianni, M. Cereda, and G. Staurenghi, "Use Case for European Robotics in Ophthalmologic Micro-Surgery," *Proceedings of the 5th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, pp. 1–32, 2015.

[190] W. T. Latt, R. C. Newton, M. Visentini-Scarzanella, C. J. Payne, D. P. Noonan, J. Shang, and G. Z. Yang, "A hand-held instrument to maintain steady tissue contact during probe-based confocal laser endomicroscopy," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2694–2703, 2011.

[191] T. Akinbiyi, C. E. Reiley, S. Saha, D. Burschka, C. J. Hasser, D. D. Yuh, and A. M. Okamura, "Dynamic augmented reality for sensory substitution in robot-assisted surgical systems," *Annual International Conference of the IEEE, Engineering in Medicine and Biology*, pp. 567–570, 2006.

[192] M. Kitagawa, D. Dokko, A. M. Okamura, and D. D. Yuh, "Effect of sensory substitution on suture-manipulation forces for robotic surgical systems," *Journal of Thoracic and Cardiovascular Surgery*, vol. 129, no. 1, pp. 151–158, 2005.

[193] K. Miller, K. Chinzei, G. Orssengo, and P. Bednarz, "Mechanical properties of brain tissue in-vivo: Experiment and computer simulation," *Journal of Biomechanics*, vol. 33, pp. 1369–1376, 2000.

[194] M. Tonutti, G. Gras, and G.-Z. Yang, "A machine learning approach for real-time modelling of tissue deformation in image-guided neurosurgery," *Artificial Intelligence in Medicine*, vol. 80, pp. 39–47, 2017.

[195] S. Giannarou, M. Ye, G. Gras, K. Leibrandt, H. Marcus, and G.-Z. Yang, "Vision-based deformation recovery for intraoperative force estimation of tool-tissue interaction for neurosurgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 6, 2016.

[196] Fred L Bookstein, "Principle Warps: Thin-Plate Splines and the Decomposition of Deformations," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 6, 1989.