



<http://researchspace.auckland.ac.nz>

ResearchSpace@Auckland

Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

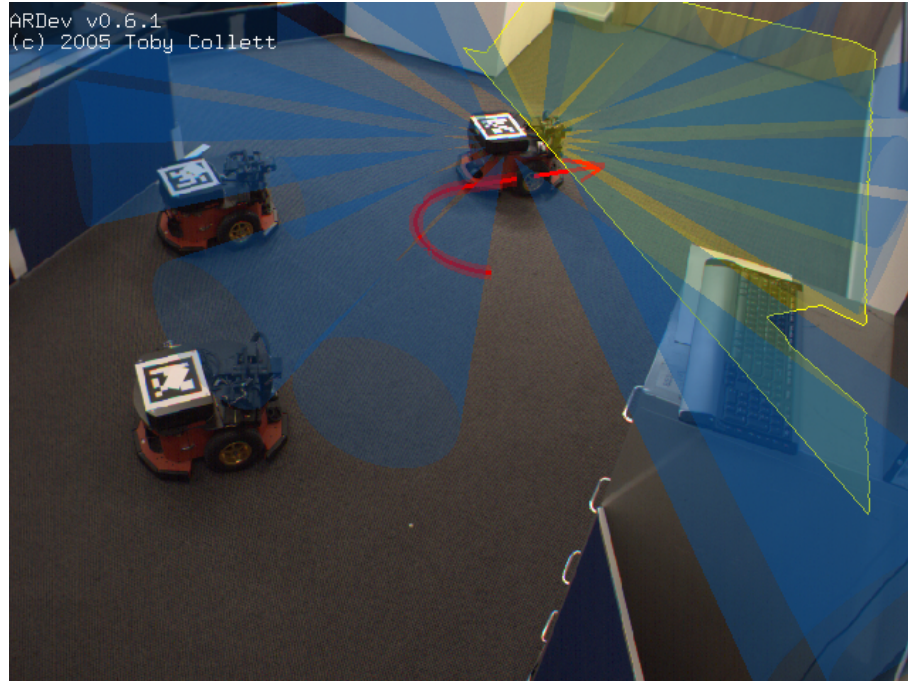
To request permissions please use the Feedback form on our webpage.

<http://researchspace.auckland.ac.nz/feedback>

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form.

*Department of Electrical and Computer Engineering
The University of Auckland
New Zealand*



Augmented Reality Visualisation for Mobile Robot Developers

Toby H. J. Collett

August 2007

Supervisor: Dr. Bruce MacDonald



A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY IN ENGINEERING,
THE UNIVERSITY OF AUCKLAND, 2007

Abstract

Developer interactions with robots during the testing and debugging phases of robot development are more complex than and distinct from general software application development. One of the primary differences is the need to understand the robot's view of the environment and the inconsistencies between this and the actual environment. Augmented reality (AR) provides an ideal way to achieve this, allowing robot program data to be displayed in context with the real world. This allows for easy comparison by the developer, highlighting the cause for any bugs in the robot behaviour. An AR debugging space is created in this work that allows the developer to have this enhanced understanding of the robot's world-view, thus improving developer efficiency.

Over the past decade robots have begun to move out of industrial assembly lines and into environments that must be shared with human users. Many of the tasks that we wish robots to perform in these environments require close interaction and collaboration with human users. The move away from the constrained environment of a production line means that the tasks required of robots are more varied, their operating environment is far more complex and unpredictable, and safety can no longer be achieved through isolation of the robot.

The result of these influences has been to change robot programming from a simple task of instructing the robot to perform a sequence of steps to an open ended challenge of specifying dynamic interactions that robot developers are still coming to terms with. Robot development is more than just design and code entry and a broader approach to improving robot development is needed. One of the founding principles of this thesis is that robot development should be approached as a human-robot interaction issue, this particularly applies to the testing and debugging phases of the development.

The nature of the robot platform, the tasks the robot is required to perform and the environments that robots work within are significantly different from those of the desktop application. Hence robot developers need a tailored tool chain that focuses on this unique combination of issues. Current robot programming research is dominated by robot APIs and frameworks, leaving support tools to be developed in an ad hoc manner by developers as features are required. This leads to disjointed tools that have minimal feature sets;

tools that generally have poor portability when applied to other robot developments. This work examines the needs of the developer in terms of a general purpose robot visualisation tool.

One of the fundamental requirements of a general purpose robot visualisation tool is that a set of stock visualisations must be available for the developer. A prerequisite to providing these is to have a set of standard interfaces to provide the visualisations for. The open source robot framework Player/Stage was used throughout this work to provide standardised access to robot hardware. As part of this research the author has contributed heavily to the Player/Stage project, particularly as one of the key developers of the 2.0 release of Player. This new release simplifies Player development and increases the ease of maintenance of Player drivers and the efficiency of the server core.

To evaluate the benefits of AR visualisation an intelligent debugging space was developed, which runs as a permanent installation in a robotic development lab providing an enhanced view of the robot's behaviour to the developer. The space is capable of automatically detecting the presence of robots and displaying visualisations of the standard interfaces of the robot, such as its sensors and effectors. The debugging space also allows the developer to create custom renderings, leveraging the developer's ability to determine the most salient items of their code and display these.

A set of representative case studies was carried out using the debugging space for testing and debugging. These studies showed that AR provides an opportunity to understand the type of errors that are encountered during debugging. Debugging is essentially a process of elimination and by understanding the type of error developers can quickly eliminate large sets of potential bug sources, focusing on the sections of code that are causing the bug and therefore substantially reducing debugging time.

The implemented system also shows that AR provides an important stepping stone between simulation environments and the real world.

This thesis contributes the novel approach of applying AR to developer interactions with robots. The use of AR has been shown to have significant benefits for the robot developer, enhancing their understanding of the robot's world-view and hence reducing debugging time. As part of the work a flexible AR visualisation tool was developed with close integration to the Player/Stage project. This tool creates an intelligent debugging space where developers can exploit the benefits of the AR visualisation with minimal overhead.

Acknowledgements

First and foremost I would like to acknowledge the support and contributions of my supervisor Dr Bruce MacDonald. His enthusiasm and ideas have helped shape this work from our early discussions on human-robot interaction through to the final stages of proofing.

I would like to thank my wife Katie for her support and encouragement which have helped me through the last four years, and for efforts in proof reading. I would also like to thank my family for their support and help with proof reading.

The Player project has been an indispensable part of this work and I would like to thank Brian Gerkey and the rest of the Player developers for creating and supporting an excellent robotics tool.

I would like to thank Grant Sargent and the rest of the University of Auckland robotics group for their ideas and for making the robotics lab a great place to work.

Finally I would like to acknowledge the financial support of the New Zealand Tertiary Education Commission through the Top Achiever Doctoral scholarship.

Contents

1	Introduction	1
1.1	Nature of Robot Development	2
1.2	Enhancing Developer-Robot Interaction	4
1.3	Augmented Reality	6
1.4	Contributions	7
1.5	Structure	7
2	Literature Review	9
2.1	Robot Programming	9
2.1.1	Robot Development Technologies	10
2.2	Human-Robot Interaction	13
2.2.1	The role of the user in Human-Robot Interaction	14
2.2.2	Requirements for Effective Interaction	15
2.2.3	Human-Robot Interface Methods	16
2.2.4	Direct Interfaces	16
2.2.5	Indirect Interfaces	20
2.2.6	Multi Modal Communication	22
2.2.7	Survey of Interaction Evaluation	23
2.3	Visualisation	24
2.4	Augmented Reality	26
2.4.1	Immersive AR - Video See-Through	26
2.4.2	Immersive AR - Optical See-Through	28
2.4.3	Immersive AR - Projected Systems	29
2.4.4	Desktop AR	29
2.4.5	Current directions in AR	30
2.4.6	The Application of AR to Robotics	30
2.5	Summary	33

3	Augmented Reality for Robot Developers	35
3.1	Model of a robot program	36
3.2	Characterising Robot Data	37
3.3	Defining the Robot Developer	40
3.4	Potential Benefits of Augmented Reality	41
4	A Conceptual Augmented Reality System	43
4.1	The AR design space	43
4.1.1	What data should be displayed?	44
4.1.2	How should the data be represented?	44
4.1.3	Where should the data be displayed?	47
4.1.4	How should the data be viewed?	47
4.1.5	Accessing the AR system and robot data	48
4.2	Paradigms of an AR interaction system	49
4.2.1	Direct Library Access	49
4.2.2	Intelligent Debugging Space	49
4.2.3	AR enabled Robot IDE	50
4.2.4	AR whiteboard	51
4.3	Evaluation Methodology	51
4.4	Summary	52
5	AR toolkit for Visualisation	53
5.1	ARDev: an AR library for robot developers	54
5.2	Toolkit API	55
5.2.1	ARDev Core	56
5.2.2	Output Object	56
5.2.3	Capture Object	58
5.2.4	Camera Object	58
5.2.5	FrameProcess Object	59
5.2.6	Position Object	59
5.2.7	Render Object	60
5.3	System Performance	60
5.4	Reference Object Implementations	61
5.4.1	Capture Objects	64
5.4.2	Camera Objects	64
5.4.3	Output Objects	65
5.4.4	Process Objects	65
5.4.5	Position Objects	67
5.4.6	Render Objects	67

5.5	Implemented Player Modules	67
5.6	Implementation of an Augmented Reality Debugging Space	68
5.6.1	The AR Configuration Manager	75
5.6.2	The AR Configuration UI	76
5.6.3	The Plug-in Driver	76
5.7	Visualisation Design	78
5.8	Functional Evaluation	81
5.9	Summary	83
6	Case Study Evaluation	87
6.1	Case Study Design	88
6.2	Test Setup	88
6.3	Case Study Tasks	89
6.3.1	Follower	89
6.3.2	Blockfinder	90
6.3.3	Pick Up the Block	92
6.4	Initial Case Study Results	92
6.4.1	Follower	93
6.4.2	Block Finder	93
6.4.3	Pick Up the Block	98
6.5	Participant Case Study Results	101
6.5.1	General Participant Notes	102
6.5.2	Errors Located With AR System	105
6.5.3	Limitations of the Participant Study	105
6.6	Discussion	106
6.7	Summary	108
7	The Player 2.0 Distributed Framework	111
7.1	Original Player Architecture	112
7.2	Motivation for Changes	113
7.3	Player 2.0 Requirements	114
7.4	Player 2.0 library division	115
7.4.1	Player core	115
7.4.2	Transport layer	116
7.4.3	New usage paradigms	116
7.5	Example plug-in driver for the new API	116
7.6	The Player Client Libraries	119
7.7	Summary	121

8	Future Work and Conclusions	123
8.1	Future Work	123
8.2	Conclusions	125
A	Case Study Source	129
A.1	Study 1: Follower	129
A.2	Study 2: Block Finder	130
A.3	Study 3: Block Picker	134
B	Example ARDEV Configuration	139

List of Figures

2.1	Screenshot of the GSV simulation and visualisation tool	12
2.2	Visualisation of robot data in a 3D simulation	13
2.3	Shared Human-Robot Perceptual Space	15
2.4	Interface Methods	16
2.5	Augmented Reality Process	27
2.6	Example Augmented Reality Output	28
2.7	AR arrows indicating object of interest with small group of robots	31
2.8	AR rendering of planned foot placements for the HRP-2 robot	31
2.9	AR Bubblegram showing the current state of the Aibo robot	32
2.10	AR rendering of the active axes on a KUKA robotic arm	33
3.1	General Robot Program Fragment	36
4.1	Eclipse IDE debug window	45
4.2	DDD visualisation of a list structure	46
5.1	Source for simple RenderObject extension for ARDev	54
5.2	ARDev Software Architecture	56
5.3	Source for simple ARDev session rendering laser data for a pioneer robot	57
5.4	Render Tree	60
5.5	Set up of the video see-through head mounted display	62
5.6	Wall mounted hardware configuration	64
5.7	Calibration application	66
5.8	HMD view of laser data	69
5.9	Pioneer odometry history	70
5.10	Sonar Sensors on Pioneer Robot	71
5.11	Shuriken Robot with IR and Sonar	71
5.12	B21r with Bumper and PTZ visualisation	72
5.13	Actuator Array and Limb interface visualisation	73
5.14	Augmented system localisation visualisation	74

5.15	Configuration Manager Separation	75
5.16	Configuration Manager Classes	76
5.17	Configuration UI - Main Window	77
5.18	Configuration UI - Environment Configuration	77
5.19	Configuration UI - Display list Configuration	78
5.20	Rendering of three lasers in a single colour	80
5.21	Rendering of three lasers with a unique colour for each robot	80
5.22	Comparison of Virtual and AR visualisation of laser scan - Virtual Data	84
5.23	Comparison of Virtual and AR visualisation of laser scan - Augmented Data	85
6.1	Pioneer 3DX Robot	89
6.2	Intelligent Debugging Space	90
6.3	Block used in block finder and block picker tasks	91
6.4	Block Finder Task Setup	92
6.5	Screen captures from the follower trial	94
6.6	Simulated block finding task with visualisation	95
6.7	Errors in simulation caused by interpolation of laser scans	96
6.8	AR visualisation used with the block finding task	97
6.9	AR visualisation of the actuator array	100
6.10	Error between actual and reported orientation of end effector	101
7.1	Example of potential Player 2.0 server connections	117
7.2	Hokuyo URG laser scanner	118
7.3	Header file for urg laser driver	119
7.4	Additional code for plug-in module	119
7.5	Source of urg laser driver	120

List of Tables

3.1	Data types used in Player interfaces	39
5.1	Key Objects Implemented in the Prototype System	63

