Conferences in Research and Practice in Information Technology

Volume 150

User Interfaces 2014

Australian Computer Science Communications, Volume 36, Number 5

User Interfaces 2014

Proceedings of the
Fifteenth Australasian User Interface Conference
(AUIC 2014), Auckland, New Zealand,
Auckland, New Zealand, 20 - 23 January 2014

Burkhard C. Wünsche and Stefan Marks, Eds.

Volume 150 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.

Published in association with the ACM Digital Library.

User Interfaces 2014. Proceedings of the Fifteenth Australasian User Interface Conference (AUIC 2014), Auckland, New Zealand, 20 - 23 January 2014

Conferences in Research and Practice in Information Technology, Volume 150.

Editors:

Burkhard C. Wünsche
Department of Computer Science
The University of Auckland
Private Bag 92019
Auckland 1142
New Zealand
Email: burkhard@cs.auckland.ac.nz

Stefan Marks
School of Computing and Mathematical Sciences
AUT University
Private Bag 92006
Auckland 1142
New Zealand
Email: stefan.marks@aut.ac.nz

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
Simeon J. Simoff, University of Western Sydney, NSW
Email: crpit@scem.uws.edu.au

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 150.
ISSN 1445-1336.
ISBN 978-1-921770-33-3.

Document engineering, January 2014 by CRPIT
On-line proceedings, January 2014 by the University of Western Sydney
Electronic media production, January 2014 by the AUT University

The *Conferences in Research and Practice in Information Technology* series disseminates the results of peer-reviewed research in all areas of Information Technology. Further details can be found at http://crpit.com/.

Table of Contents

Contributed Papers

Contributed Posters

Preface

It is our great pleasure to welcome you to the 15th Australasian User Interface Conference (AUIC), held at AUT University in Auckland, New Zealand, January 20th to 23rd 2014. AUIC is one of 13 co-located conferences that make up the annual Australasian Computer Science Week (ACSW).

AUIC provides an opportunity for researchers in the areas of User Interfaces, HCI, CSCW, and pervasive computing to present and discuss their latest research, to meet with colleagues and other computer scientists, and to strengthen the community and explore new projects, technologies and collaborations.

This year we have received a diverse range of submission from all over the world. Out of 28 submitted papers, 9 papers were selected for full paper presentations and 5 were selected for posters. The breadth and quality of the papers reflect the dynamic and innovative research in the field and we are excited to see the international support. Accepted papers were rigorously reviewed by the community to ensure high quality publications.

We offer our sincere thanks to the people who made this years conference possible: the authors and participants, the program committee members and reviewers, the ACSW organizers and the publisher CRPIT (Conference in Research and Practice in Information Technology).

Burkhard C. Wünsche
The University of Auckland

Stefan Marks
AUT University

AUIC 2014 Programme Chairs
January 2014

Programme Committee

Chairs

Burkhard C. Wünsche, The University of Auckland, New Zealand
Stefan Marks, AUT University, New Zealand

Members

Robert Amor, The University of Auckland, New Zealand
Mark Apperley, The University of Waikato, New Zealand
Nilufar Baghaei, UNITEC, New Zealand
Christoph Bartneck, HITLab NZ, New Zealand
Mark Billinghurst, HITLab NZ, New Zealand
Rachel Blagojevic, The University of Auckland, New Zealand
Wolfgang Broll, TU Ilmenau, Germany
Rafael Calvo, University of Sydney, Australia
David Chen, Griffith University, Australia
Andy Cockburn, University of Canterbury, New Zealand
Sally Jo Cunningham, The University of Waikato, New Zealand
Geraldine Fitzpatrick, TU Vienna, Austria
Tom Gedeon, Australian National University, Australia
Judith Good, University of Sussex, United Kingdom
John Grundy, Swinburne University of Technology, Australia
Jun Hu, Eindhoven University of Technology, Netherlands
Christof Lutteroth, The University of Auckland, New Zealand
Paul Lyons, Massey University, New Zealand
Bruce MacDonald, The University of Auckland, New Zealand
Sylvain Malacria, University of Canterbury, New Zealand
Stuart Marshall, Victoria University of Wellington, New Zealand
Masood Masoodian, The University of Waikato, New Zealand
Mathieu Nancel, University of Canterbury, New Zealand
Marcin Nowina-Krowicki, Defence Science and Technology Organisation (DSTO), Australia
Helen Purchase, University of Glasgow, United Kingdom
Andreas Rausch, TU Clausthal, Germany
Holger Regenbrecht, University of Otago, New Zealand
Hokyoung Ryu, Hanyang University, Republic of Korea
Haifeng Shen, Flinders University, Australia
Ross T. Smith, University of South Australia, Australia
Maki Sugimoto, Keio University, Japan
Stewart von Itzstein, University of South Australia, Australia
Gerald Weber, The University of Auckland, New Zealand
Robert Wellington, AUT University, New Zealand

Organising Committee

Chairs

Tony Clear and Russel Pears

Venue

Tony Clear

Communications

Russel Pears, Hui Ling Tan, Melanie Curry-Irons and Ryan Butler

Finance

Alison Clear and Eva Ihaia

Sponsorship

Stephen Thorpe

Operations

Adam Winship and Eva Ihaia

Programme, proceedings and booklet

Alison Clear

Catering and registration

AUT Hospitality Services

Welcome from the Organising Committee

On behalf of the Organising Committee, it is our pleasure to welcome you to Auckland and to the 2014 Australasian Computer Science Week (ACSW 2014). Auckland is New Zealand's largest urban area with a population of nearly one and a half million people. As the centre of commerce and industry, Auckland is the most vibrant, bustling and multicultural city in New Zealand. With the largest Polynesian population in the world, this cultural influence is reflected in many different aspects of city life. ACSW 2014 will be hosted at the City Campus of Auckland University of Technology (AUT), which is situated just up from the Town Hall and the Auckland central business district. ACSW is the premier event for Computer Science researchers in Australasia. ACSW2014 consists of conferences covering a wide range of topics in Computer Science and related areas, including:

- Australasian Computer Science Conference (ACSC) (Chaired by Bruce Thomas and Dave Parry)
- Australasian Computing Education Conference (ACE) (Chaired by Jacqueline Whalley and Daryl D'Souza)
- Australasian Information Security Conference (AISC) (Chaired by Udaya Parampalli and Ian Welch)
- Australasian User Interface Conference (AUIC) (Chaired by Burkhard C. Wünsche and Stefan Marks)
- Australasian Symposium on Parallel and Distributed Computing (AusPDC) (Chaired by Bahman Javadi and Saurabh Kumar Garg)
- Australasian Workshop on Health Informatics and Knowledge Management (HIKM) (Chaired by James Warren)
- Asia-Pacific Conference on Conceptual Modelling (APCCM) (Chaired by Georg Grossmann and Motoshi Saeki)
- Australasian Web Conference (AWC2013) (Chaired by Andrew Trotman and Michael Sheng)

This year reflects an increased emphasis for ACSW on community building. Complementing these published technical volumes therefore, ACSW also hosts two doctoral consortia and a number of associated workshops, including those for the Heads and Professors of Computer Science, plus for the first time the 'Australasian Women in Computing Celebration'. Naturally in additional to the technical program, there are a range of events, which aim to provide the opportunity for interactions among our participants. A welcome reception will be held in the atrium of the award winning newly built Sir Paul Reeves Building, which has integrated the city campus as a hub for student activity and provides a wonderful showcase for this year's ACSW. The conference banquet will be held on campus in one of the reception rooms in this impressive complex.

Organising a multi-conference event such as ACSW is a challenging process even with many hands helping to distribute the workload, and actively cooperating to bring the events to fruition. This year has been no exception. We would like to share with you our gratitude towards all members of the organising committee for their combined efforts and dedication to the success of ACSW2014. We also thank all conference co-chairs and reviewers, for putting together the conference programs which are the heart of ACSW, and to the organisers of the symposia, workshops, poster sessions and accompanying conferences. Special thanks to Alex Potanin, as the steering committee chair who shared valuable experiences in organising ACSW and to John Grundy as chair of CoRE for his support for the innovations we have introduced this year. We'd also like to thank Hospitality Services from AUT, for their dedication and their efforts in conference registration, venue, catering and event organisation. This year we have secured generous support from several sponsors to help defray the costs of the event and we thank them for their welcome contributions. Last, but not least, we would like to thank all speakers, participants and attendees, and we look forward to several days of stimulating presentations, debates, friendly interactions and thoughtful discussions.

We hope your stay here will be both rewarding and memorable, and encourage you to take the time while in New Zealand to see some more of our beautiful country.

Tony Clear
Russel Pears
School of Computer & Mathematical Sciences

ACSW2014 General Co-Chairs
January, 2014

CORE - Computing Research & Education – OLD VERSION

CORE welcomes all delegates to ACSW2013 in Adelaide. CORE, the peak body representing academic computer science in Australia and New Zealand, is responsible for the annual ACSW series of meetings, which are a unique opportunity for our community to network and to discuss research and topics of mutual interest. The original component conferences - ACSC, ADC, and CATS, which formed the basis of ACSW in the mid 1990s - now share this week with eight other events - ACE, AISC, AUIC, AusPDC, HIKM, ACDC, APCCM and AWC which build on the diversity of the Australasian computing community.

In 2013, we have again chosen to feature a small number of keynote speakers from across the discipline: Riccardo Bellazzi (HIKM), and Divyakant Agrawal (ADC), Maki Sugimoto (AUIC), and Wen Gao. I thank them for their contributions to ACSW2013. I also thank invited speakers in some of the individual conferences, and the CORE award winner Michael Sheng (CORE Chris Wallace Award). The efforts of the conference chairs and their program committees have led to strong programs in all the conferences, thanks very much for all your efforts. Thanks are particularly due to Ivan Lee and his colleagues for organising what promises to be a strong event.

The past year has been turbulent for our disciplines. ERA2012 included conferences as we had pushed for, but as a peer review discipline. This turned out to be good for our disciplines, with many more Universities being assessed and an overall improvement in the visibility of research in our disciplines. The next step must be to improve our relative success rates in ARC grant schemes, the most likely hypothesis for our low rates of success is how harshly we assess each others' proposals, a phenomenon which demonstrably occurs in the US NFS. As a US Head of Dept explained to me, "in CS we circle the wagons and shoot within".

Beyond research issues, in 2013 CORE will also need to focus on education issues, including in Schools. The likelihood that the future will have less computers is small, yet where are the numbers of students we need? In the US there has been massive growth in undergraduate CS numbers of 25 to 40% in many places, which we should aim to replicate. ACSW will feature a joint CORE, ACDICT, NICTA and ACS discussion on ICT Skills, which will inform our future directions.

CORE's existence is due to the support of the member departments in Australia and New Zealand, and I thank them for their ongoing contributions, in commitment and in financial support. Finally, I am grateful to all those who gave their time to CORE in 2012; in particular, I thank Alex Potanin, Alan Fekete, Aditya Ghose, Justin Zobel, John Grundy, and those of you who contribute to the discussions on the CORE mailing lists. There are three main lists: csprofs, cshods and members. You are all eligible for the members list if your department is a member. Please do sign up via http://lists.core.edu.au/mailman/listinfo - we try to keep the volume low but relevance high in the mailing lists.

I am standing down as President at this ACSW. I have enjoyed the role, and am pleased to have had some positive impact on ERA2012 during my time. Thank you all for the opportunity to represent you for the last 3 years.

Tom Gedeon

President, CORE
January, 2013

ACSW Conferences and the Australian Computer Science Communications

The Australasian Computer Science Week of conferences has been running in some form continuously since 1978. This makes it one of the longest running conferences in computer science. The proceedings of the week have been published as the *Australian Computer Science Communications* since 1979 (with the 1978 proceedings often referred to as *Volume 0*). Thus the sequence number of the Australasian Computer Science Conference is always one greater than the volume of the Communications. Below is a list of the conferences, their locations and hosts.

2015. Volume 37. Host and Venue - University of Technology, Sydney, NSW.

2014. Volume 36. Host and Venue - AUT University, Auckland, New Zealand.

2013. Volume 35. Host and Venue - University of South Australia, Adelaide, SA.
2012. Volume 34. Host and Venue - RMIT University, Melbourne, VIC.
2011. Volume 33. Host and Venue - Curtin University of Technology, Perth, WA.
2010. Volume 32. Host and Venue - Queensland University of Technology, Brisbane, QLD.
2009. Volume 31. Host and Venue - Victoria University, Wellington, New Zealand.
2008. Volume 30. Host and Venue - University of Wollongong, NSW.
2007. Volume 29. Host and Venue - University of Ballarat, VIC. First running of HDKM.
2006. Volume 28. Host and Venue - University of Tasmania, TAS.
2005. Volume 27. Host - University of Newcastle, NSW. APBC held separately from 2005.
2004. Volume 26. Host and Venue - University of Otago, Dunedin, New Zealand. First running of APCCM.
2003. Volume 25. Hosts - Flinders University, University of Adelaide and University of South Australia. Venue - Adelaide Convention Centre, Adelaide, SA. First running of APBC. Incorporation of ACE. ACSAC held separately from 2003.
2002. Volume 24. Host and Venue - Monash University, Melbourne, VIC.
2001. Volume 23. Hosts - Bond University and Griffith University (Gold Coast). Venue - Gold Coast, QLD.
2000. Volume 22. Hosts - Australian National University and University of Canberra. Venue - ANU, Canberra, ACT. First running of AUIC.
1999. Volume 21. Host and Venue - University of Auckland, New Zealand.
1998. Volume 20. Hosts - University of Western Australia, Murdoch University, Edith Cowan University and Curtin University. Venue - Perth, WA.
1997. Volume 19. Hosts - Macquarie University and University of Technology, Sydney. Venue - Sydney, NSW. ADC held with DASFAA (rather than ACSW) in 1997.
1996. Volume 18. Host - University of Melbourne and RMIT University. Venue - Melbourne, Australia. CATS joins ACSW.
1995. Volume 17. Hosts - Flinders University, University of Adelaide and University of South Australia. Venue - Glenelg, SA.
1994. Volume 16. Host and Venue - University of Canterbury, Christchurch, New Zealand. CATS run for the first time separately in Sydney.
1993. Volume 15. Hosts - Griffith University and Queensland University of Technology. Venue - Nathan, QLD.
1992. Volume 14. Host and Venue - University of Tasmania, TAS. (ADC held separately at La Trobe University).
1991. Volume 13. Host and Venue - University of New South Wales, NSW.
1990. Volume 12. Host and Venue - Monash University, Melbourne, VIC. Joined by Database and Information Systems Conference which in 1992 became ADC (which stayed with ACSW) and ACIS (which now operates independently).
1989. Volume 11. Host and Venue - University of Wollongong, NSW.
1988. Volume 10. Host and Venue - University of Queensland, QLD.
1987. Volume 9. Host and Venue - Deakin University, VIC.
1986. Volume 8. Host and Venue - Australian National University, Canberra, ACT.
1985. Volume 7. Hosts - University of Melbourne and Monash University. Venue - Melbourne, VIC.
1984. Volume 6. Host and Venue - University of Adelaide, SA.
1983. Volume 5. Host and Venue - University of Sydney, NSW.
1982. Volume 4. Host and Venue - University of Western Australia, WA.
1981. Volume 3. Host and Venue - University of Queensland, QLD.
1980. Volume 2. Host and Venue - Australian National University, Canberra, ACT.
1979. Volume 1. Host and Venue - University of Tasmania, TAS.
1978. Volume 0. Host and Venue - University of New South Wales, NSW.

Conference Acronyms

ACDC	Australasian Computing Doctoral Consortium
ACE	Australasian Computer Education Conference
ACSC	Australasian Computer Science Conference
ACSW	Australasian Computer Science Week
ADC	Australasian Database Conference
AISC	Australasian Information Security Conference
APCCM	Asia-Pacific Conference on Conceptual Modelling
AUIC	Australasian User Interface Conference
AusPDC	Australasian Symposium on Parallel and Distributed Computing (replaces AusGrid)
AWC	Australasian Web Conference
CATS	Computing: Australasian Theory Symposium
HIKM	Australasian Workshop on Health Informatics and Knowledge Management

Note that various name changes have occurred, which have been indicated in the Conference Acronyms sections in respective CRPIT volumes.

ACSW and AUIC 2014 Sponsors

We wish to thank the following sponsors for their contribution towards this conference.

Host Sponsor

Auckland University of Technology,
www.aut.ac.nz

Platinum Sponsor

DATACOM,
www.datacom.com.au

Gold Sponsor

The University of Auckland,
www.auckland.ac.nz

Silver Sponsors

COLAB - AUT Design+Creative Technologies,
colab.aut.ac.nz

Institute of IT Professionals, New Zealand,
www.iitp.org.nz

CORE - Computing Research and Education,
www.core.edu.au

Bronze Sponsor

SERL - AUT Software Engineering Research Laboratory,
www.serl.aut.ac.nz

Australian Computer Society,
www.acs.org.au

Publication Sponsor

University of Western Sydney,
www.uws.edu.au

Contributed Papers

Math Tutor: An Interactive Android-Based Numeracy Application for Primary Education

Zainab Masood **Rashina Hoda**
SEPTA Research, Electrical and Computer Engineering
The University of Auckland
Private Bag 92019, Auckland, New Zealand
zmas690@aucklanduni.ac.nz r.hoda@auckland.ac.nz

Abstract

With growing exposure of children to handheld and mobile devices, there is an increasing interest in exploring the use of mobile technology for educational purposes. In particular, touch-based devices seem to promise great potential in this domain. In this paper, we present Math Tutor – an Android-based application designed to help children learn and practice early numeracy addition and subtraction (take away) as well as help teachers monitor and review children's progress, with support for English and Māori languages. We describe the design and development process, features of the application, and the results of a usability evaluation. This project takes a step towards creating interactive platforms required for educating the upcoming generation of digital natives.

Keywords: Interactive software applications, primary education, numeracy, child-computer interaction, Android, Māori

1 Introduction

In the last few years, mobile technology has changed faster than ever in ways that affect our lives in every aspect. Children, in particular, who are born or growing up in this 'brave NUI world' (Wigdor and Wixon 2011) are considered digital natives (Palfrey et. al 2008). It's not unusual to find children manipulating mobile devices even before they've mastered their alphabets and numbers. Touch-based interactive devices being easy to use by children as well as the teachers are considered important and handy tools for teaching core concepts such as numeracy in early and primary years learning.

There have been variety of effective traditional ways in which children are taught numeracy in classrooms, but research suggests that children are more engaged and grasp concepts quickly when they enjoy the subject matter being taught. At the same time, children are more attracted to colourful and interactive applications – aspects weakly supported by traditional white-board based teaching approaches.

The aim of our project was to conduct research and development into this upcoming area of inquiry.

Our Android-based application - *Math Tutor* - is targeted for classroom teaching and been designed specifically to be used both by students and teachers. A part of our application is designed to help students learn and practice single digit addition and subtraction using numbers and images. The application's GUI and colours have been carefully chosen as to attract children and make learning easy and fun. The children are encouraged to learn by using the simple technique of earning medals when they complete a particular level. The second part of the application has been designed for the teachers and includes functionalities such as language selection, exercise mode selection, etc. (described later in detail). The functionality for teachers to login and perform administrative activities can be easily found in web-based applications but is currently hard to find in tablet-based applications. There have been other touch-based applications and games designed to teach children early single digit addition and take away, but unfortunately we find very few applications that associate numbers with (equal number of) objects visually and have multi-lingual options. Our society being a multicultural society consists of people belonging to different parts of the world who speak more than one language. As such, many children take some time learn to speak and understand English. So the teachers may find it difficult to teach addition and take away with applications that just have English as a language option. For this reason the second language chosen for this application is Māori, which is one of the three official languages for the people living in New Zealand and is a core part of early and primary school education. This prototype can be expanded to include other languages with relative ease.

The intended users of this application are children between five and six years of age and the teachers who will be assisting and using the application to manage students. A student can practice addition and take away for up to two levels i.e. for numbers from 1 to 5 and between 1 to 10. The application uses two techniques used for addition and taking away, first using simple single digits e.g. 3+2=5 and second using the visual concept of adding/taking away things with counting equal number of objects like fruits. Children can use the objects (e.g. fruit images) to count and find the answers. Upon successful completion, children will earn different coloured star medals as rewards that will encourage them to practice more and improve their skills.

The application tracks student responses. Results, including number of correct and incorrect responses with percentage and medals are available for teachers for further analysis and evaluation.

2 Related Works

Unique capabilities of computer applications for providing learning practice include: the combination of visual displays, animated graphics and speech; the ability to provide feedback and keep a variety of records; and the opportunity to explore a situation interactively (Clements 2002). A study by Starkey et. al (2004) found that children can develop mathematical competence with developmentally appropriate mathematics learning software. Another longitudinal study examining the role that technology plays in mathematical skill achievement for kindergartners and 1st graders shows that students' access to and use of technology influences their future academic achievement in school (Espinosa et. al 2006).

Year one (at school) marks a transition to a more academically-oriented approach to learning as children may now be in a full program after half-day long kindergarten. In New Zealand, numeracy instruction, along with other subjects, becomes more structured in school as opposed to a more flexible framework offered in kindergarten by the *Te Whā riki* (curriculum) (2007). Lessons are more structured, and there are new facts to master. After acquiring the basic skills of number recognition, counting, addition and subtraction are naturally taught next, as students combine sets of objects and count the results.

It has been argued that carefully designed sequences of worked-out examples (without lectures or other direct instruction) can prove to be more effective in teaching several mathematical skills (Price et. al 2013). The role of drills has been emphasized in achieving the fluency to memorize basic mathematical facts and improve these skills. The use of this technique not only seen to help in achieving fluency in mathematical knowledge but also expected to aid in higher mathematics skills as per research (Zhu et. al 1987). However, children can find these math drills as repetitive, monotonous and boring exercises.

Trends suggest that pre-school and elementary-age children may soon be using smart devices seamlessly first at home and then perhaps in the classroom of 2015 as a normal part of growing up in a digital age (Chiong and Shuler 2010). Therefore, we need to focus on designing useful and productive applications which not only educate children but are also fun to use. Norris et. al compared the use of tablets with netbooks and laptops and concluded that features like portability, availability, instant-on devices, and longer battery time makes tablets a preferred medium over others for learning (Norris et. al 2011).

One of the most popular smart phone technologies today is Android. Android has witnessed a growth of over 250% with 300 million android devices activated, adding 850,000 new device activations every day (Rubin 2012). All those new Android users also have access to 450,000 applications on the Android market, an increase of 300% from last year (Rubin 2012). And this growth rate continues to grow along with the android applications available for users of all age. Android users have access to numerous applications on the Android market through Google Play (https://play.google.com). These include numerous learning applications for primary years.

Some such educational android applications which are available on Google Play to help children improve numeracy skills include: Kids Math, Math Mole, Math flash cards, Kids Math Book, First Grade Math Challenge and 1st Grade Math.

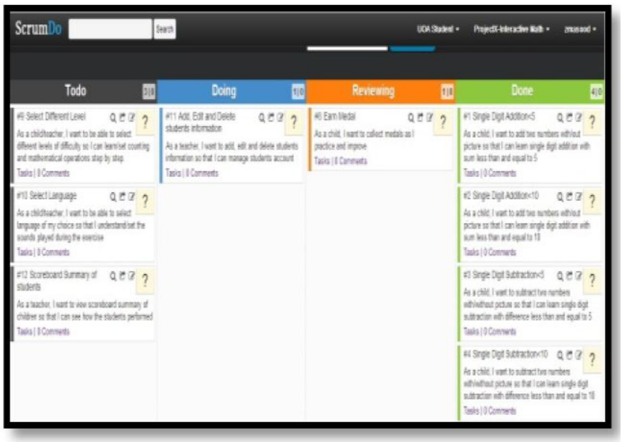


Figure 1. ScrumDo Project Management Board

While possessing different functionalities in parts, most of these applications are focused on individual users (parents and children at home) and do not offer comprehensive solutions for classroom-based teaching, such as teacher access and administration.

3 Project Design

3.1 Software Development Method

We used Agile software development as our software development methodology for this project (Cockburn and Highsmith 2001). Agile methods are based on an iterative and incremental style of development where each iteration involves set of tasks to be accomplished during the iteration. Agile software development methods are known to be flexible to change and are based on changing needs and preferences. In particular, we employed Scrum – one of the most popular Agile methods in the world today (Schwaber and Beedle 2002).

For this project, a roadmap was created to plan the activities and the timelines based on the user stories created using ScrumDo as the tool (http://www.scrumdo.com). Functionalities represented as user stories were created to help visualise the system and streamline the requirements of the application. Then a set of objectives were defined for the user stories during weekly scrum/stand-up meetings scheduled with the project supervisor (second author) to share progress and plan activities of the week. The meeting focused on what was done during last week, what was the plan for the coming week, and what was getting in way i.e. the impediments. The scrum board was updated accordingly with status of user stories using ScrumDo and iterations were updated as well.

Based on this, tasks for next iteration were assigned and were expected to be delivered before the next meeting. For any queries during the week, the coordination was done through emails. It helped to ensure that the project was on the right track with status of final deliverables.

3.2 Software Tools

3.2.1 User Interface Design Tools

For designing the user interface of the application, a custom set of images were created and some free images were chosen and modified including icons for the application, buttons and background. Adobe Photoshop and Adobe Illustrator were used as image customisation and processing tools.

3.2.2 The Android Framework

Figure 2. Samsung Galaxy Tablet (front/back) and Emulator (left)

The Android framework was chosen as the platform for the development of the application primarily due to its popularity and relative affordability of tablet devices supporting Android (as compared to iPad for example.) Eclipse was used as the Integrated Development Environment (IDE) with Java as the programming language. Android 14 was used as the version of Software Development Kit (SDK) for compatibility with the android device selected for this project: Samsung Galaxy 7 inches tablet and Android OS 4.1 (Jelly Bean).

3.2.3 Testing and Validation Tools

Unit tests were implemented to verify the correctness of the methods that were developed using the LogCat and the associated Log class provided by Android. Also, a usability evaluation was conducted to verify the usability of the android application with users, which is described later along with results and refinements resulting from the evaluations. Due to time limitations of this project, evaluations were conducted using university students rather than ideal users – children and teachers – which required lengthy ethics approval processes. This aspect is further discussed in the limitations section later.

3.3 Software Design Decisions

Certain critical software design decisions were made to enhance the application's usability based relevant research. Research was conducted to select the best design strategies to apply during the design and development of the application for young children. As proved by research, generally the more a software interface adapts to children's cognitive levels, the more it will be accepted and adored by children (Xiaodong 2010). Similarly, using bright, vivid colours that stimulate the senses, creating a happy, playful mood using cartoon characters (Lazaris, 2012), making icons look "clickable" by using three-dimensional imagery, adding navigational elements that are large and easy to find (Hafit et. al 2012) are the best practices which should be used for designing applications for children.

Also interface with much word information results in the attention detraction for children and turbulence of vision procedure, so the principal interface i.e. the exercises screens were designed to be simple. However, other screens are designed to include colours and imagery in such a way that the children enjoy and love using the application.

For children, it is the physical design (in this case, the user interfaces) that appeals to them; and this factor is perhaps as important as the actual functionality provided by the application. Children tend to choose the application that features characters they can relate to and those that appeal to them aesthetically. Also, while designing the prototypes the strategies from the research conducted were incorporated. The designed interfaces catering to students and teachers are presented in Fig.3 and Fig.4 respectively.

Figure 3. Design Prototypes for Student Functionalities

The following sections describe the application key features, the application design, and application implementation with the details of the iterations followed from the beginning towards the end of project.

4 Math Tutor: Key Features

The actual implementation of the application focused on the major features: Child Learner Exercises (Addition and Take Away), Reward System, Teacher Administration, Language Setting, and Exercise Mode Settings.

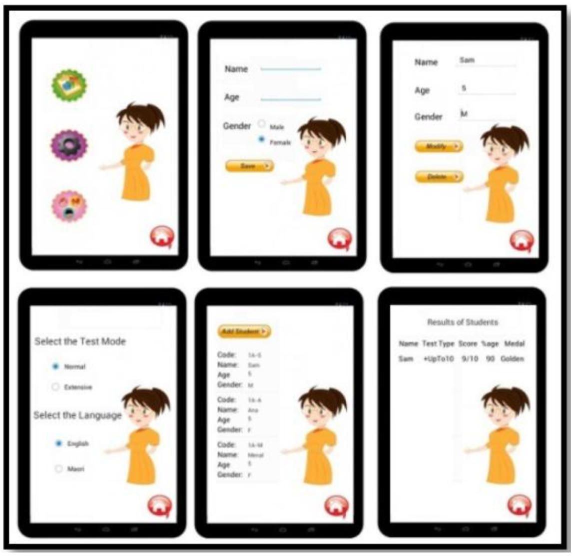


Figure 4. Design Prototypes of Teacher Functionalities

4.1.1 Child Learner Exercises (Add/Takeaway)

When the child enters the student menu screen (Fig. 3) and selects the exercise type as *Add up to 5 or 10* or *Take Away up to 5 or 10*, then a set of questions are dynamically generated randomly with the fruit images and corresponding numbers. The child has to answer each question by clicking one of the answer options displayed. If the answer is correct, a sound is played to help the child associate the answer with the sound as correct; and another sound for incorrect response. Once all the questions are displayed the child is prompted with either moving to the scoreboard or to retry the exercise.

4.1.2 Reward System

When the child selects to move to the scoreboard the results of the exercise can be viewed with the earned medal based on the percentage achieved.

Four different medals are rewarded to students as shown in Fig.5. If the percentage achieved is greater than 80% a golden medal is earned by the student, if the percentage is greater than 65% a silver medal is earned by the student. Similarly if the percentage is greater than 50% then silver medal is earned by the student and red medal if the percentage is less than 50%.

4.1.3 Teacher Administration

The functionality of this part includes choosing the language for the audio outputs (English or Māori) and selecting the exercise mode from normal to exhaustive. The functionality also includes adding, deleting and modifying student's details. Teachers can view every student's scores during different practice exercises and compare them with the previous results.

4.1.4 Language Settings

A teacher can use this screen to select between the two languages English and Māori. English is used as the default language. However if the user wants to change the language, radio button can be used for switching the mode of the language.

Figure 5. Scorecard (top) and Reward medals (bottom)

4.1.5 Exercise Mode Settings

A teacher can use this screen to select the questions mode as normal or exhaustive. This is set as 10 for the normal mode, i.e. the number of questions for exercise is 10.

5 Application Design: LoFi Prototyping

To start the project first the user stories were created. The application was designed keeping in mind the age of the end users which are children from age's 5-6. The design was kept fairly simple and easy to understand by the children. The design was also made colourful and attractive so that the children find it enjoyable. The navigation was purposely kept simple and not over-crowded with too many options as children can get confused if they are given a lot of options to move forward, backward and to the main menu. The colours for the overall design were carefully chosen and kept consistent throughout the application.

We created low-fidelity paper prototypes to get an understanding of the application flow and design (Kangas and Kinnunen 2005, Virzi et. al 1996). These prototypes were transformed to digital black and white mock-ups using *Lucidchart* (https://www.lucidchart.com) as a wireframe tool (Fig. 6)

Figure 6. Paper (left) and Wireframe (right) Prototypes

We focused on improving the application prototypes by adding colours and pictures. Since the end-users of the application are children so care was taken to design with children's interest in mind.

6 Application Development

Following Scrum's iterative and incremental, the development was executed in a series of iterations. With each of these iterations, the design and development were assessed and refined, while moving closer to completion. In the following sub-sections, we describe the activities and key design aspects achieved in the iteration providing a sense of the evolving application.

6.1 Iteration 1

The development of Android applications involves some installation pre-requisites like the installation Eclipse SDK, Android SDK and Android Development Tools (ADT) to get a running environment. For developers, Android SDK provides a rich set of tools, including debugger, libraries, handset emulator, documentation, sample code, and tutorials. Android applications can be easily developed using Eclipse (Android's official development platform) with the help of a plug-in called Android Development Tools (ADT).

The development phase consists majorly of designing and developing activities. A concise definition of activity as stated in the official help for eclipse states "An activity is a logical grouping of functionality that is centred around a certain kind of task." Hence an activity is a group of sub-classes designed and developed to create the interface and add functionality to them.

The first development iteration involved implementing the user stories of the students. The addition of numbers up to 5 was programmed initially by generating different questions randomly using the built in random number function and against every questions four options were randomly generated correspondingly. For this a java class was created for random generation of numbers and method of the class was called in the addition activity.

This activity generated questions with two random numbers such that there sum is less than or equal to five. Then four options were generated and displayed on buttons with one as the sum of the two numbers and the other three options as random numbers other than the result and less than and equal to 5. This was called recursively till the number of questions is equal to mode selected. To ensure that the options are not displayed always on same position, the position is randomly relocated by generating different sequence of locations (Fig. 7)

6.2 Iteration 2

The second iteration involved addition of numbers up to 10. For this the addition class was modified to handle the random numbers generation dynamically based on the selected exercise type. Also the take away class was implemented following the same pattern. This iteration also involved the modification of `AddActivity` and `SubtractActivity` to make it more generalized for all exercises. For this purpose the referring activity passes the exercise type and `Add /SubtractActivity` then first extracts the information from the Bundle class to get the intent variables and check the maximum value of `maxnumber`. The `maxnumber` decides which type of exercise has to be started. For example if `max=5` the 5 digit addition is started.

```
Step1:Generate RandomSequence for displaying options randomly on buttons
    int optionSeq=ran.generateRandomOption()
Step2:Convert random sequence into string for conditional comparison
    StringselectedoptionSeqStr =Integer.toString(optionSeq)
Step3:Display options randomly on buttons
    if selectedoptionSeqStr equals firstsequence{
            firstbuttontext= resultStr
            secondbuttontext= option1Str
            thirdbuttontext = option3Str}// Display in this sequence
        else if selectedoptionSeqStr equals secondsequence
            Display text on buttons in different sequence
        else if selectedoptionSeqStr equals thirdsequence
            Display text on buttons in different sequence
```

Figure 7 Pseudo-code for Options Relocation

6.3 Iteration 3

In the third iteration, the objects (e.g. fruits) were displayed alongside the numbers to help student to calculate the correct answer. The results were calculated and the scores were created and displayed on the scorecard with the medal earned. During this iteration the orientation was fixed so that while using the tablet if the orientation is changed then the application is not affected. For Android, this is neither an automatic functional and needs to be implemented. Sounds (audio files) were added against the numbers in the English language.

6.4 Iteration 4

In this iteration, the user stories pertaining to the teacher-user were implemented. User Management classes were created to add and delete the students. Then the settings class was added as a separate activity to configure the language and the exercise mode and Māori audio files were added for supporting two languages to the application.

This iteration also consisted of designing the database and the schema with the students and score table in the database. The Score table is used to save results after each exercise. The iteration used SQLite database as the base class for working with a SQLite database in Android which provided methods to open, query, update and close the database. In addition it provides the execSQL() method, which allows to execute an SQL statement directly. The object ContentValues allows to define key/values. The "key" represents the table column identifier and the "value" represents the content for the table record in this column. ContentValues was used for inserts and updates of database entries. This was implemented as a dataAdapter class. Also a java class was added to pass the values from the business logic to the database.

The major functions of the dbAdapter included creating the student, fetching the students in the student section, updating and deleting the selected students, saving the scores of the student and fetching the scores against the student. This iteration also consisted of using the List View layout to display list of all student in the student's layout.

6.5 Iteration 5

In the last iteration both the modules were integrated with implementation of all the user stories and the scores were stored in the database against the student. In the end the results class was added so that teacher could view the results of the students saved in the database using the dataAdaptor class and displaying using ListView layout in the teacher section.

Every iteration was followed by the functional testing of the features added using the positive and negative test data to ensure the correctness of the application and in the end after the integration of the modules, the overall application was verified and the discrepancies were fixed and retested accordingly.

The upcoming section will provide a complete analysis of the evaluation results performed after the completion of this application.

7 Usability Evaluation and Results

An evaluation was conducted to see how effective and useful the different features are and how suitable the user interface design is to the users. The evaluations took place in The University of Auckland, Engineering Building, Masters Room.

7.1 Participants

The participants for this evaluation were seven university students with ages between 20-29 years. The application was installed on the Android emulator device and all the participants were asked to run and test the application on the same settings of emulator as of Samsung Galaxy 7 inches Tablet.

7.2 Evaluation Method

The evaluation of the application was conducted using the Android emulator configured on personal laptop. Each user was first given written instructions to follow after providing the demographic details. This document listed a small description of the tasks to be performed with the steps of the execution procedure both for student and teacher's role.

The main tasks as a student included completing the addition and takeaway exercises for all the four exercises with normal mode setting, two exercises with language as English and two with Māori as language. The tasks to be completed while evaluating the teacher module were to add/update and delete student information and to view the score of the students, doing the configurations like setting the language for the exercise.

The participants were first asked to fill in the demographic section of the questionnaire. Then after the execution of the application the usability and usefulness sections were filled by the evaluator. This questionnaire focused on the effectiveness and usability of the application. The multiple choices consisted of statements in a Likert scale having values between 1 and 5, where strongly disagree equals 1 and strongly agree equals 5.

7.3 Results

We now present the complete analysis and results obtained after performing the evaluation.

7.3.1 Demographics

The results indicate that all participants had intermediate to expert level experience in using touch based devices. A majority (85.5%) of participants have observed children using touch-based devices. Nearly all the participants agreed that these applications can add fun to learning mathematics. All of the participants agreed that they have used numerical practice sheets in their primary years learning.

7.3.2 Usefulness of Application

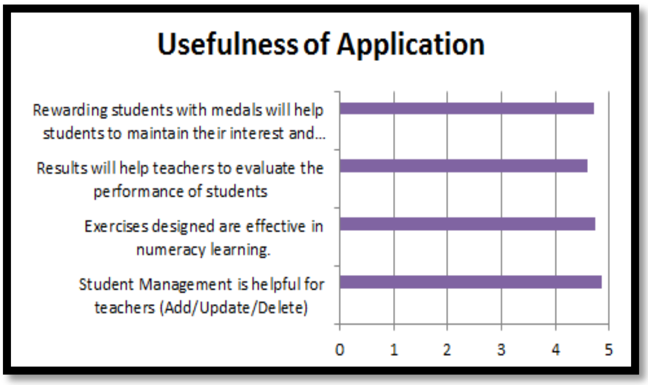


Figure 8 Mean values for Usefulness Questions

The results obtained during the evaluation procedure were used to calculate likert scale values. A median for each option was calculated using the weighted mean formula for likert scale. The results are plotted in the graphs below.

The likert scale value of 4.86 for the student update/delete value shows that almost all the participants believed that the student management module for teacher will be helpful for maintaining students. The value 4.6 in the graph below indicates that participants agree that displayed results will help teacher to monitor student's progress in a classroom. Participants agreed that the exercises designed are effective in numeracy learning with value of 4.75 and rewarding students with medals gets a 4.71 Likert scale. The results show that the participants agree that the application can actually prove to be a handy and effective tool in classroom teaching.

7.3.3 Usability of Application

The results of the usability evaluation are presented in the graph below which indicates that changing language was easy for users (4.86). All other areas including consistency in flow of application, ease in application usage and clarity of sound reached values approximately between 4.7 and 4.8 which supports that these aspects of application were approved by the evaluators.

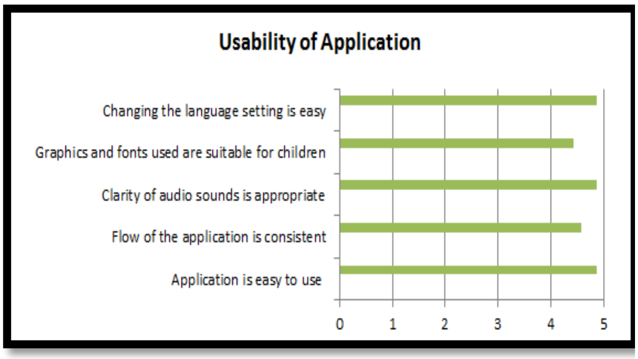


Figure 9 Mean values for Usability Questions

7.4 Suggested Refinements

The results and comments after the evaluation suggest some improvements in the application in the future.

- In the teachers section if the student names were displayed in blocks and detail of each student was displayed separately would make it easier for the teacher to view each student's details.
- In teachers sections some charts and graphs would help teacher to analyse student progress more effectively.
- If the teachers have the option to email/share the progress of a student with parents or guardians will be a helpful feature.
- More languages options such as Mandarin, Spanish, German or French can be introduced for the users belonging to different regions of world.
- At the end of each correct answer different and more interesting sounds like clapping or music could be added to make children happy and excited.

The next section will provide a general discussion and concluding remarks for the project.

8 Discussion

The application was designed and developed to help children between 5-6 years of age learn basic addition/takeaway using touch-tablet devices and help teachers manage students and monitor student progress. The results of a preliminary usability evaluation indicate that the application can be potentially effective in teaching children about basic numeracy concepts (addition and takeaway up to 5 and 10). The teacher section also promises to be used a useful classroom tool that can be used to add/update and delete students in their class list. They can view and compare the results from previous exercises and decide which students need help and more practice.

The application could not be evaluated with the actual users i.e. children aging between 5 to 6 and teachers practically in a classroom setting due to limitations of time as the university's ethics policy application process takes some months. All the participants involved in the evaluation of the application agreed that the application with some minor improvements could be a very good source for teaching a 5-6 year old children addition and take away. The evaluations if conducted using the ideal user-groups i.e. teachers and children, could be used to draw firmer conclusions about the role of this application in primary numeracy learning.

9 Conclusion

The changing trends in technology and classroom teaching suggest that touch-based mobile devices and attractive/engaging applications can be essential parts of future classrooms. The use of such devices amongst children is increasing by leaps and bounds. However, research and development of educational software that is both engaging and appropriate for classroom usage is still in its infancy. Our Math Tutor application can serve to be a good foundation for more complicated and sophisticated learning tools in future. Extension to two and three digits addition and taking away can be added in this application to be used by children ages 6 and above. The teacher section can include graphical analysis to monitor student performance with option to share student's progress report with parent's guardians as suggested by some participants. The application can also be extended to support more languages. Overall, our project is a step towards creating interactive educational software for use by digital natives in classrooms of the near future.

10 References

Ministry of Education. The New Zealand curriculum. Wellington, Learning Media, (2007) http://www.educate.ece.govt.nz/learning/curriculumAndLearning/TeWhariki.aspx

Wigdor, D. and Wixon, D. (2011): *Brave NUI World: designing natural user interfaces for touch and gesture*. Elsevier.

Palfrey, J. G., and Urs Gasser. (2008): *Born digital: Understanding the first generation of digital natives*. Basic Books.

Clements, D. H. (2002): Computers in Early Childhood Mathematics. *In Contemporary Issues in Early Childhood*, **3**(2): 160-181.

Starkey, P., Klein, A., and Wakeley, P. (2004): Enhancing young children's mathematical knowledge through a pre-kindergarten mathematics intervention. *In Early Childhood Research Quarterly*, **19**: 99–120.

Espinosa, L. M., Laffey, J. M., Whittaker, T., and Shcng, Y. (2006): Technology in the home and the achievement of young children: in Findings from the early childhood longitudinal study. *In Early Education & Development*, **17**: 421-441.

Price, G. R., Mazzocco, M. and Ansari, D. (2013): Why Mental Arithmetic Counts: Brain Activation during Single Digit Arithmetic Predicts High School Math Scores, *Journal of Neuroscience*, **33**(1):156-163.

Zhu, X. and Simon, Herbert A. (1987): Learning Mathematics from Examples and by Doing. *In Cognition and Instruction*, **4**(3): 137-166.

Chiong, C. and Shuler, C. (2010): Learning: Is there an app for that? URL: http://www.joanganzcooneycenter.org/wp-content/uploads/2010/10/learningapps_final_110410.pdf

Norris, C., Hossain, A. and Soloway, E. (2011): Using Smartphones as Essential Tools for Learning. *In Educational Technology*, **51**(3): 18-25.

Rubin, A. (2012): Android@Mobile World Congress: It's all about the ecosystem. [Online]. Available: http://googlemobile.blogspot.co.nz/2012/02/androidmobile-world-congress-its-all.html

Cockburn, A., Highsmith. J. (2001): Agile software development: the people factor. *Computer* **34**(11):131–133

Schwaber, K. and Beedle, M. (2002): *Agile Software Development with SCRUM*. Prentice-Hall.

Xiaodong P. (2010): Research of iPhone Application VI Design Based on Children Cognition Feature. *In Computer-Aided Industrial Design & Conceptual Design (CAIDCD), IEEE 11th International Conference*.

Hafit, H., Abdul Razak, F. (2012): Drawing with Fun: What We Can Learn From the Children [Online]. Lazaris, L. (2012): Designing Websites for Kids: Trends and Best Practices. http://www.smashingmagazine.com/2009/11/27/designing-websites-for-kids-trends-and-best-practices. Last accessed on 23rd August, 2013.

Available: http://www.designingforchildren.net/papers/h-hafit-designingforchildren.pdf

Kangas, E. and Kinnunen, T. (2005): Applying User-Centered Design to Mobile Application Development. *Communications of the ACM*, **48**(7), ACM.

Virzi, R. A. Sokolov, J. L., Karis, D. (1996): Usability Problem Identification Using Both Low and High-Fidelity Prototypes. *In Proceedings of Human Factors in Computing Systems Conference CHI'96*, Vancouver, BC, Canada.

Assessing the impact of a Clinical Audiology Simulator on first year students

Alexandre Heitz[1] **Andreas Dünser**[1] **Christoph Bartneck**[1] **Jonathan Grady**[2] **Catherine Moran**[2]

[1] HIT Lab NZ, University of canterbury, Christchurch
Tel.: +64 (0)3 364 2349 Fax: +64 (0)3 364 2095
Email: alexandre.heitz@canterbury.ac.nz
[2] Department of Communication Disorders, University of Canterbury, Christchurch
Tel.: +64 (0)3 364 2431 Fax: +64 (0)3 364 2760
Email: sue.bradley@canterbury.ac.nz

Abstract

Virtual Patients (VPs) have been successful in health education to promote and foster communication. Additionally, computer simulations offer the advantage of being standardized, repeatable, and do not require as much resources as role-play simulations. The research presented in this paper offers to explore the impact of a Clinical Audiology Simulator (CAS) using virtual patients technology on first year audiology students of the University of Canterbury. We look at the CAS's effects on students' perceived level of learning, confidence, and ability to conduct adequate Pure tone audiometry as well as Clinical history taking procedures. These studies showed positive results on students' perceived level of learning, and history taking skills when using the CAS as an additional training tool. We present the findings and lessons learned from these studies as well as our plans for future experiments and software implementations. VPs have the potential to offer audiology trainees more opportunities to practice and access to a wider range of pathologies as they would with their course's traditional practical sessions.

Keywords: virtual patient, computer simulation, audiology, clinical education

1 Introduction

In healthcare education, actors trained to interact with students in role-play settings are often used as Simulated Patients (SPs). However, access to SPs for healthcare education is limited due to their cost and availability. SP training is expensive. Despite the 130 medical schools in the U.S. only 5 provide simulated patients assessment. Additionally, SPs work for a low incentive, which can be around 10 USD per hour. Considering this, SPs use in clinical scenarios to produce valid interactions for clinicians and students training can be limited (Rizzo et al. 2010).

Another training method in healthcare is Virtual Patients (VPs). VPs have the potential to improve clinical practice of healthcare trainees.

They are computer generated patients that allow realistic training in controlled settings. One reality of healthcare practical education is that students can spend days doing clinical work to only see repetitive cases, when patients come in with the same pathologies. VP simulations can be used to provide a varied and standardized training among students (Collins & Harden 1998, Johnsen et al. 2005, Triola et al. 2006), assuring every one of them has the chance to train for low frequency events too, not just the cases that are the most common. However, at the same time students can choose to interact with the exact same VP again if, for instance, a debriefing made them realize they missed critical pieces of information. In addition, VPs also enable students to practice within safe boundaries and prepare them to interact with real patients where mistakes would be costly, as well as develop clinical reasoning skills(Round et al. 2009).

Previous research shows that VPs can effectively be implemented as an alternative to SPs in various types of scenarios to represent illnesses ranging from a standard medical exam, to eye or breast exams (Deladisma et al. 2009, Kotranza et al. 2009, Rizzo et al. 2009), cases for paramedic students (Conradi et al. 2009), continuity of care for pharmacy students (Fuhrman Jr et al. 2001), even geriatrics (Tan et al. 2010), surgery cases (Vash et al. 2007), or other psychiatric disorders such as disorderly conduct, post traumatic stress syndrome (PTSD), or other phobias (Gorrindo & Groves 2009, Kenny et al. 2008, Rizzo et al. 2010, Triola et al. 2006). This technology can also enable students to practice from their own homes (Stanton 2008). VPs have also been successfully used with patients with inadequate health literacy to explain medical documents (Bickmore et al. 2009, 2010), or to promote healthy behaviours such as adherence to medication and exercise (Bickmore & Picard 2005, Hayes-roth & Saker 2003, Ruttkay & Welbergen 2008).

On the other hand, VPs' main limitation, when compared to SPs, is frequently their vocabulary. The process of talking with a patient is often script based. Developing an interaction script for one VP is a time consuming process and often needs to be strengthened following pilot studies to make it usable. Work has been done on implementing ways to create more robust scripts efficiently for VP interactions but it is still an area that needs to be improved (Halan et al. 2010, Rizzo et al. 2010, Rossen et al. 2010). Consequently, there is little research on implementing a simulation platform using a multitude of VPs for history taking training to support students' practical experience. History taking is the procedure of interviewing a patient and recording past events or circumstances that could be relevant to the patients' Health state.

For the purpose of our research, we implemented a computer simulation platform aimed for trainee audiologists (more information on this sample of students in the studies' sections). We propose to investigate how using this VP based simulator as an additional tool to traditional means of learning can affect students' performance in their course, and during role-play situations using SPs. We will also investigate how the VPs simulator we implemented affects students' perceived level of learning and confidence when used in addition to regular teaching.

This paper is going to present two studies. The first is a pilot study that took place during the first part of the course year and focuses on the pure tone audiometry procedure. The second study investigates clinical history taking and spreads over the second part of the year.

2 Related work

VPs have been used in a wide range of specialities to train clinical reasoning and patients' assessment in different settings. As part of a collaboration between London's universities, the Second Life virtual environment has been used as a way to generate VPs to train paramedics and allow them to explore more open ended questions in their decision making process. The researchers concluded that VPs in a virtual environment can offer realism impossible to reach in a classroom environment, allowing them to experience the consequences of the different choices they make (Conradi et al. 2009). At the University of Munich in Germany, a VP based systems allowed students to train on gynaecological sonography. The conclusions showed that training with virtual patients seems comparable to live practice with the advantage of a standardized consistent output which is not the case of real patients (Heer et al. 2004). The research from the following groups is considered the most relevant for our own, as the technologies they use are of similar level to ours, and consequently our terminology matches as well.

The University of Florida's Virtual Experiences Research Group provided our research team with the basis of our simulation platform. The group's early studies concerned the possibility of creating a system with patient-doctor interactions, focused on assessing whether current technology enabled to simulate such experiences, with sufficient immersion and fidelity. The Virtual Experiences Research group focused on implementing medical history interviews for different disciplines as well as investigating different ways to improve user-VP exchanges. Studies exploring different ways to affect interactions have also been conducted, using life size VPs thanks to projectors for instance, investigating natural interactions such as hand gesture (both real hand through infra red cameras, or virtual hands), tablet PC and audio (Ferdig et al. 2007, Kotranza et al. 2009, Johnsen et al. 2005). Real size displays, using projectors were also found to allow users to display more empathy than head mounted displays (Johnsen & Lok 2008). Consequences of the use of synthesized speech and natural recorded speech for the VPs answer have also been investigated and compared. It was concluded that if the intent was to teach what questions to ask, then both methods were as effective. However, if the purpose was to teach how to ask questions, the realism of prerecorded speech could make up for its low flexibility (Dickerson et al. 2006).

Rather than assessing its continuous use by medical trainees, their evaluations often focused on comparing the quality of the VP to an SP using the Maastrich assessment of Simulated Patients questionnaire, student posture, tone of voice and speech content for different exams such as a breast, eye exams or pharmaceutical exams (Deladisma et al. 2009, Kotranza et al. 2009, Rizzo et al. 2009), and studies investigating how racial and social disparities affect participants relation with VP. In the different conditions explored, research showed significant results that participants were able to act with VPs, to a similar or satisfactory level compared with simulated patients (Ferdig et al. 2007, Johnsen et al. 2005).

Another research group that researches VPs is the institute for creative technologies of the University of South California. This group primarily investigates the use of VPs with military technicians and clinicians to treat soldiers, which can be dangerous both mentally and physically. VPs are considered for PTSD patients' suicide tendencies, or even traumatic brain injuries. Significant results were found and participants were able to detect trauma, its duration, or presenting its origin. However, symptoms requiring a deep level of exchanges with the VPs to identify them were almost not detected due to the state of the scripts implemented, and current performances of speech recognition (Kenny & Parsons 2010, Kenny et al. 2008). VPs were also used as an online guide to promote access to psychological healthcare information and assist as well as encourage military personal and their family to seek care if necessary.

Looking at this researches, the main downside of the VPs-user interactions appears to be that their interaction are scripted, thus a necessary time consuming step is to implement a script with sets of clues or triggers for each of the VP's answers. This is the main limitation VPs currently suffer from and that can prevent them from being as realistic as SPs. In addition, an underdeveloped script can be frustrating for participants (Johnsen et al. 2005). Development of an interaction script for one VP is a time consuming process and often needs to be strengthen following pilot studies to make it usable. Work has been done on implementing ways to create more robust scripts efficiently for VP interactions but it is still an area that needs to be improved (Halan et al. 2010, Rizzo et al. 2010, Rossen et al. 2010). As a consequence, VP simulations used in health education are commonly focusing on a single case or VP (Bickmore & Picard 2005, Deladisma et al. 2009, Fuhrman Jr et al. 2001, Gorrindo & Groves 2009, Hayes-roth & Saker 2003, Heer et al. 2004, Kenny et al. 2008, Kotranza et al. 2009, Rizzo et al. 2010, 2009, Ruttkay & Welbergen 2008, Stanton 2008, Tan et al. 2010, Triola et al. 2006, Vash et al. 2007).

Our research is focusing on the field of Audiology. In audiology, simulations have been used with success to teach clinical skills and simulators allowing practice of procedural skills such as pure tone audiometry are available on the market (e.g. Otis Audiology simulator, Parrot Software's Audiology Clinic) for Universities to use to supplement traditional course work. There is, however, a lack of simulators incorporating such procedural skills with clinical history taking training possibilities and immersing students in realistic experiences from meeting a patient up to, and including pathology assessments. Our research aims to explore the use of a computer simulation based on a varied range of VPs, with audiology trainees as our primary target audience. We aim to answer whether this system, used as an additional training tool during the curriculum, can positively impact students' clinical skills, perceived level of learning, and confidence. We reviewed students' grades as part of the Clinical Audiology Observation and Practice course to assess the impact of the pure tone audiometry component of the simulator. In addition, students were also assessed with role-play situation using SPs to evaluate their ability to conduct clinical interview and pure tone audiometry procedures following exposure to our computer simulation.

3 The Clinical Audiology Simulator

3.1 System architecture

The computer simulation used in this study is referred to as the Clinical Audiology Simulator (CAS). The CAS is used to practice procedural skills, clinical history taking and puretone audiometry, as well as decision making. This takes the form of the standard audiology range of tests including history taking, pure tone audiometry, otoscopy, and pathology diagnosis for Virtual Patients (VPs). The CAS is based on a simulation platform originally implemented by the University of Florida's Virtual Experiences Research Group (VERG 2005). The initial platform had been implemented for research study purposes and was adapted to run as a standalone application. The CAS has been implemented in C#/C++ using Visual studio 2010. The application makes use of the open source 3D library Ogre for graphics, to render the VPs as well as the room where the consultations take place within an embedded window (see Fig. 1).

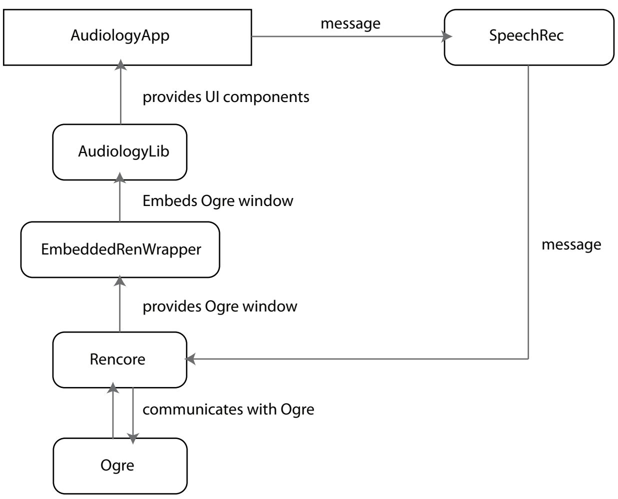


Figure 1: CAS main system architecture

3.2 Using the Clinical Audiology Simulator

When launching the CAS, a student starts with selecting one of the VPs among the different cases offered (refer to Fig. 2 for this section). The application then starts and students have access to the different features of the software, labeled 'Interview', 'Otoscopy', 'Tone test', and 'Submit Results'. The students can choose the order of the procedures but would typically start by interviewing the patient.

The interview component of the consultation takes the form of a loop where the students ask questions to the VPs which will be answered according to the VPs' scripts. The students interact with the VP, who is sitting in a room, by typing the questions he/she wishes

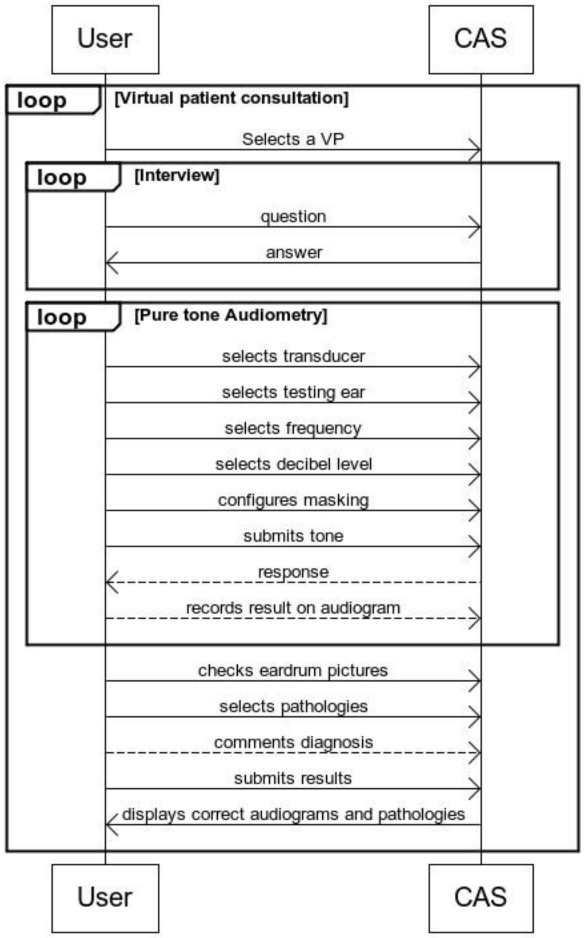


Figure 2: Using the CAS sequence diagram

to ask (see Fig. 3 Interview interface of the CAS). The VP will either answer the students' question, if it was understood, or hint the students to reformulate and/or ask another question (e.g. "Could you please rephrase that?"). The students' aim here is to collect and sort from the VPs' answers the necessary information to complete a Diagnostic Adult History Form that is commonly used in clinic when interviewing patients. The students can stop this process at any time if they consider that all the relevant information got retrieved from the VP.

A student can then check the VPs' ears if he considers it necessary. This will display two eardrum pictures from our collection of pictures retrieved on real patients, which can present additional elements to help identify the appropriate diagnosis.

The student would then follow with the Pure tone Audiometry procedure, which takes the form of another loop. This procedure is about determining the VPs' hearing thresholds. Hearing thresholds represent a patient's hearing levels, which is his ability to hear sounds properly across the range of frequencies the human ear should be able to detect. Hearing thresholds are typically displayed on audiograms, which are graphs of the hearing level in Decibel for a set of frequencies ranging from 250 Hz to 8 kHz. To determine hearing thresholds in the CAS, a student will have to first select a transducer between supra-aural, insert, and bone before deciding the ear to be tested.

This represents the type of headphone used, each having its specificities. Then, the procedure in-

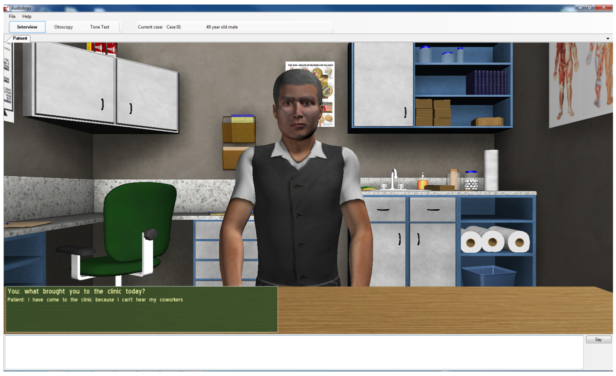


Figure 3: Interview interface of the CAS

volves submitting different intensities of sound over the tested frequencies until the VPs are able to hear them, if at all. When a tone is submitted, the VPs can either answer, or not. Once a VP answers to a particular tone the student should mark the response level on the virtual audiograms. Masking is configured for the non-test ear depending on the patient's responses, if the student considers there could have been conduction from one ear to another during tests. This process is repeated for the whole range of frequencies the students decides to test, for both ears. Fig. 4 represents the interface the students face when conducting a pure tone audiometry procedure on the CAS.

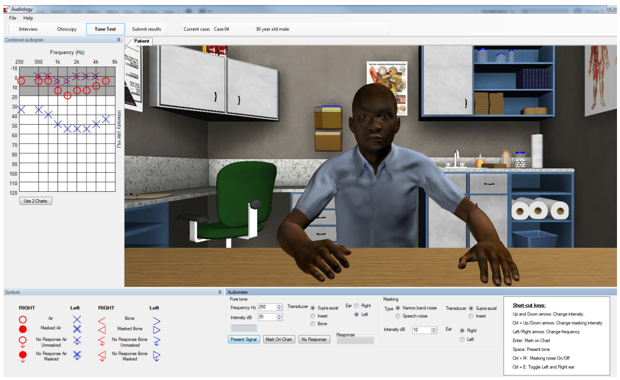


Figure 4: Tone Test interface of the CAS

Following those assessments, the student has to submit his/her results. The student has to determine the pathology(ies) associated with this patient. In addition, a student can choose to add a comment on his diagnosis decision. This information can be recorded and used for assessment. Finally, once the diagnosis is submitted the student will be given feedback in the form of the VP's actual audiogram, and its correct diagnosis.

The CAS was deployed on a total of seven computers to allow participants to practice during their free time. During their designated training period, participants were able to access the computers and practice on the CAS at any time of the week.

4 Evaluation

To fit the participants' curriculum in the Clinical Observation and practice course the investigation was split into two. First is the Pure tone audiometry pilot study. This pilot is followed by the Clinical history taking study, which is our main evaluation.

4.1 Pure Tone audiometry pilot study

The Pure tone audiometry pilot study focuses on learning the procedure and reasoning behind determining patients' hearing threshold. This procedure is typically taught to students in the first part of the year. Our pilot took place during that period of the students' training. This pilot helped us to test the research setup, as well as measure the quality of our evaluation. The main hypotheses were:

1. Students' ability to conduct a pure tone audiometry procedure accurately will increase as a result of using the CAS in addition to traditional methods.
   (a) Students' Perceived level of learning will improve as a result of using the CAS to practice pure tone audiometry in addition to traditional methods.
   (b) Students' grades when assessed in Role-play, and on the simulator will increase as a result of using CAS in addition to traditional methods.

4.1.1 Participants

The University of Canterbury's Master of Audiology Degree (Maud) is spread over two years, with the bulk of theoretical teachings being focused in the first year. Entry into the Maud is very restricted and competitive and only ten to twelve students per year are accepted into the program. The primary reason for this is that audiologists' training, just like any healthcare professional, is resource intensive, particularly in term of practical training. However, within the audiology field in a small country such as New Zealand, training opportunities are even more limited. According to audiology teachers, the first year students are the ones benefiting the most from extra training opportunities as they are relatively new to the field, starting with virtually no previous domain knowledge. It would also prime them on having the set required skills to make the best of their summer placements.

A group of twelve students was recruited, ten females and two males. All participants achieved at least a Bachelor degree in tertiary education. Each student was enrolled in the Clinical Observation and Practice course which includes one day a week of observation and practice at various clinics in the Christchurch area.

Nine students reported English as being their native speaking language, two reported German, and one reported Chinese. All students are fluent in English. The twelve students declared having adequate vision and hearing to use the simulator, as well as conduct and record a medical history from a patient, as well as having adequate hearing. Five students rated their computer skills as *Above average*, six as *Average*, and one as *Below average*. However, the twelve students answered they considered having the necessary level of skills required to operate the Clinical Audiology Simulator (CAS).

4.1.2 Design

The Pure tone audiometry pilot study took place in the first semester. Participants were split into two groups, Group A and Group B. Each group was

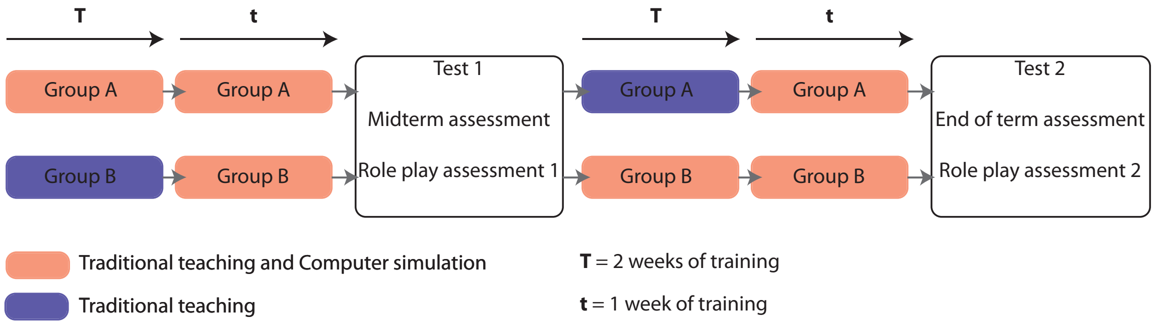
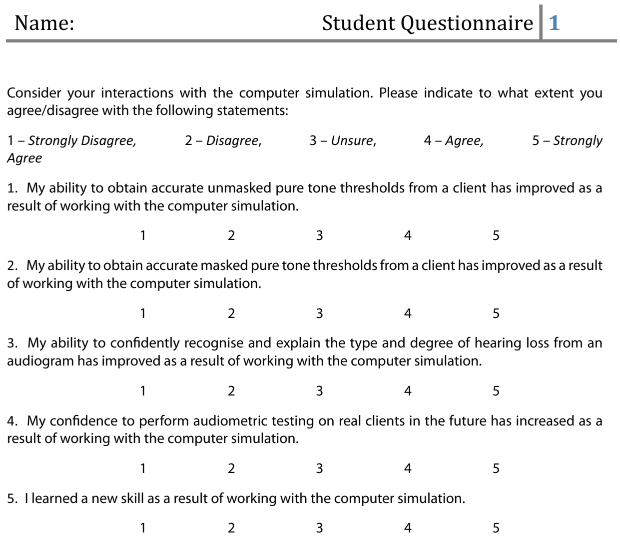


Figure 5: Pure tone audiometry study design

made of five female participants and one male participant, while the mean gpa (grade point average) of the two groups was counterbalanced (Group A *mean gpa*=7.48, *sd*=0.97, and Group B *mean gpa*=7.46, *sd*=0.87).

Following a presentation of the CAS as part of a tutorial session, participants in Group A got access to the simulator in addition to their in-class teaching for a period of two weeks. During that time, Group B was only training using traditional means of learning. After this first training phase, both groups had access to the simulator for a week in order to prepare for 'Test 1', their midterm assessment and extra role-play assessment. In addition, participants answered a questionnaire on their experience with the CAS. After these assessments, the conditions were reversed and Group B had access to the CAS for a period of two weeks in addition to their courses while Group A was only following traditional learning methods. After these two weeks, both groups had again access to the simulator in order to prepare for their end of term assessment. This assessment also got followed by a second role play assessment, and another CAS questionnaire (Fig. 5).


Name: Student Questionnaire | 1

Consider your interactions with the computer simulation. Please indicate to what extent you agree/disagree with the following statements:

1 – Strongly Disagree, 2 – Disagree, 3 – Unsure, 4 – Agree, 5 – Strongly Agree

1. My ability to obtain accurate unmasked pure tone thresholds from a client has improved as a result of working with the computer simulation.

1 2 3 4 5

2. My ability to obtain accurate masked pure tone thresholds from a client has improved as a result of working with the computer simulation.

1 2 3 4 5

3. My ability to confidently recognise and explain the type and degree of hearing loss from an audiogram has improved as a result of working with the computer simulation.

1 2 3 4 5

4. My confidence to perform audiometric testing on real clients in the future has increased as a result of working with the computer simulation.

1 2 3 4 5

5. I learned a new skill as a result of working with the computer simulation.

1 2 3 4 5

Figure 6: Extract of Pure tone pilot questionnaire

4.1.3 Measures

Midterm and end of term assessment grades were used as a measure for this experiment. For both these assessments, participants had to determine and report on paper the hearing thresholds of three different patients using the computer simulator. Marks were allocated depending on how close the hearing thresholds reported were to the actual threshold of the patients implemented.

Role-play assessments took place in the audiology clinic located on campus where research assistants played the role of Simulated Patients (SP) in realistic clinical settings. Participants were asked to perform a pure tone procedure on the SP and report their results. Marks were allocated to participants following typical clinical work marking criteria where in addition to actual threshold validity, points were also granted for method, pace, confidence, and explaining results.

A questionnaire was given to the participants following the role play situations. The main measure of interest was the mean of five 5 points likert-scale questions referring to participants' perceived level of learning as a result of using the CAS (Fig. 6).

4.1.4 Results

We used the Kolmogorov-Smirnov test to attest the normality of the data. For the perceived level of learning scores, the results were non significant for the two groups at Time 1, and at Time 2, which shows normality. For the learning outcomes, similarly, the results of the test show that the data assumes normality for the two learning outcomes (role play grade and simulator grade), for both groups, at Time 1 and at Time 2.

For the perceived level of learning scores, we used a nonparamatric independent sample u-test. This analysis showed significant differences for perceived level of learning between the two groups at Time 1 (U=1.50, p=.007), but not at Time 2 (U=15.50, p=.684).

For the learning outcomes, we used t-tests to analyse the differences in role-play assessments and simulator assessments between the two groups at Time 1, at Time 2, and on the difference off scores between Time 2 and Time 1. No statistical differences were found between the two groups for these measures.

The means and standard deviations for these variables are presented in Table 1.

4.1.5 Discussion

Preliminary data was analyzed to explore if the additional exposure to our VP based simulator could increase audiology students' perceived level of learning and performance. Students were assessed using both the CAS and with a SP based role-play sce-

Table 1: Descriptive statistics for perceived level of learning, role play assessment grades, and simulator assessment grades at Time 1, and at Time 2

		Time 1		Time 2	
	Group of the student	Mean	Std. dev.	Mean	Std. dev.
Perceived level of learning	Group A	3.57	0.50	3.70	0.55
	Group B	1.80	0.90	3.60	0.49
	Total	2.68	1.16	3.65	0.50
Role play grade	Group A	74.31	6.32	78.94	17.33
	Group B	68.29	14.55	80.56	9.42
	Total	71.30	11.15	79.75	13.33
Simulator grade	Group A	82.22	11.26	88.62	7.31
	Group B	84.92	7.08	90.71	6.02
	Total	83.57	9.08	89.66	6.47

nario. First, as expected, the students did improve their score independently of their learning group on perceived level of learning, Simulator assessment, and Role-play assessment over the course of the experiment.

At the start of the experiment we hypothesized that students' perceived level of learning would increase following extended exposure to the CAS. The results support our hypothesis as Group A showed at Time 1 significantly higher perceived level of learning, following their dedicated period of exposure to the CAS. Following group B's exposure, this difference is not significant anymore which could indicate that they caught up with their peers (Fig. 7). Statistical equivalence has however not been calculated.

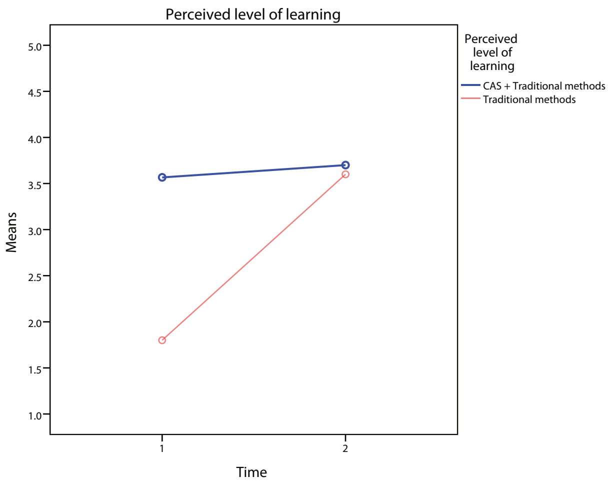


Figure 7: Perceived level of learning scores in function of time

The second hypothesis was that students would get better grades when assessed following their exposure to the CAS. The results, however, show that while there is an increase in students grades when assessed over time, for both types of assessments, both groups score increase similarly.

4.1.6 Limitations

The main limitation of this study is that both groups received one week of training with the CAS right before each assessments. This was one of the restriction of testing students as part of the Clinical Audiology course, and one of the course requirements; every student had to have access to the CAS before those course assessments. However, this allowed us to use both students' midterm and end of term assessments as measures.

This week of common training, however, seemed to have leveled potential differences between the two groups. A few students reported during informal discussions that they got the most of their training done during the week prior to each assessment, whether in group A, or group B. Additionally, it is possible that students generally spent more time training on the CAS to prepare for the end of term assessments than for midterm's. We concluded that there are two ways to remedy this problem.

First is to remove the common week of training both groups had. Second, is to designate specific training times for students in both groups, and have them book specific time slots to ensure that the student group exposed to the CAS receives a sufficient amount of additional training, and control CAS exposure times between the two groups. This would also mean to abandon mid-term assessments and end of term assessment as measures. Indeed as mentioned earlier, to guarantee fairness for the students, they should all have access to the CAS before any of their course assessments. This means that the study needs to run before their midterm, while leaving enough time for the control group to practice as well following the study, in preparation of their exam.

In addition, it seems that an increase in students' perceived level of learning does not correlate with an increase in results. This measure will be removed as well from the following studies which will focus on the transfer of skill assessed with role-play situations.

4.2 Clinical history taking study

This study focused on assessing students on one of the main clinical exams our system supports, clinical history taking. Clinical history taking is centred around learning how to interview patients to retrieve key information that help determining an accurate diagnosis and follow up procedures. Assessments are conducted using role-play with an SP in order to test for transfer of skills from use of the CAS. While SPs are not real patients they allow for a simulated experience which is the closest to real practice. In addition, SPs enable to standardize the assessment, presenting a similar patient to each participant.

This experiment was conducted using the same prototype of the CAS. While still testing the functionality of our system, this experiment also aimed to answer the following hypotheses:

1. Students' confidence when conducting a clinical history interview will improve as a result of using the CAS, compared to students who were only given traditional instruction.

2. Students' ability to retrieve information in clinical history situation will improve as a result of using the CAS, compared to students who were only given traditional instruction.

3. Students' ability to accurately report information in clinical history taking will improve following exposure to the CAS, compared to students who were only given traditional instruction.

4. As a result of using the CAS in addition to traditional instruction, students will take clinical history more efficiently and will require less interaction with patients to retrieve relevant information, when compared to students who were only given traditional instruction.

4.2.1 Participants

The same participants who took part in the pure tone audiometry pilot study were recruited for this study.

4.2.2 Design

Participants for this study were split into two equal groups according to the results of a pre test role play situation. Following discussions with the course coordinator and other audiology expert, Students' grade point average had been determined to not be a valid representation of clinical interview taking abilities, thus could not be considered as a means to group students for this study. Indeed, clinical history taking skills somehow differ from other procedures as students need to ask questions accordingly, all the while interpreting each of the patients' answers to contribute towards a diagnosis.

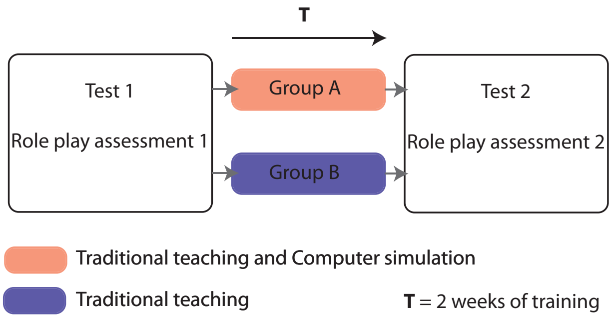


Figure 8: Clinical history taking study design

Following the pre-test (Test 1), participants were split into two groups, CAS and no CAS balanced based on the accuracy, confidence, and efficiency results of this test (see the following section for more details on these measures). Students in the CAS group then practised for a period of two weeks in addition to traditional teaching while students in the no CAS group only followed the traditional instruction. Students practising on the CAS agreed to train at least two hours. At the completion of this training phase both groups undertook a second role play assessment (Test 2), with a similar assessment method to Test 1 (see Fig. 8).

4.2.3 Measures

Interview taking skills are not typically graded as part of the first year audiology students' curriculum. Our main focus was to measure students' abilities during role play situations. While quantifying the improvement of the students is outside the scope of this paper, we assess whether the CAS had a positive impact on students interview taking skills. The role play simulations took place in a realistic setting, with two audiology experts acting as SPs; one for assessment 1, the other for assessment 2. During those experiences students were assessed on accuracy, where students had to retrieve information from the VPs and translate them onto standard history sheets, confidence and efficiency. Assessments were undertaken by two audiology experts to ensure reliability.

Verbal Accuracy: The verbal accuracy score is based on the number of answers participants were able to retrieve from the SP while talking during the interview. This measure was assessed using a transcript of the interaction between participants and SPs. Transcripts were recorded during student-SP interactions by one of the audiology experts. It is presented as a percentage.

Written Accuracy: The written accuracy score points to the number of adequately reported pieces of information on a history diagnosis form used in clinics to report information critical to diagnose the patients. Students were asked to fill in the diagnosis form while interacting with the SP, as they do during clinical exams. Clinical forms were then checked by both audiology experts and marked. This measure is presented as a percentage.

Confidence: For each role play situation, participants rated pre and post simulation confidence in their performance using three seven point Likert scale items.

Number of questions: For each role play situation, this measure is the number of questions a student asked the SP before considering the clinical history complete.

Efficiency: The efficiency score is calculated by using the verbal accuracy raw scores divided by the number of questions asked to an SP.

Table 2: Descriptive statistics for perceived level of learning, role play assessment grades, and simulator assessment grades at Time 1, and at Time 2

		Time 1		Time 2	
	Group of the student	Mean	Std. dev.	Mean	Std. dev.
Confidence	Group A	3.78	0.73	4.86	0.77
	Group B	3.63	0.99	4.33	1.25
	Total	3.71	0.81	4.62	1.00
Verbal accuracy	Group A	42.64	8.91	72.48	8.75
	Group B	42.33	8.92	60.93	7.05
	Total	42.49	8.46	67.23	9.73
Written accuracy	Group A	65.65	16.85	81.09	13.17
	Group B	59.03	21.42	76.60	14.86
	Total	62.64	18.37	79.05	13.44
Nbr of questions	Group A	24.67	8.04	44.50	6.09
	Group B	22.40	6.35	38.00	6.52
	Total	23.64	7.06	41.55	6.86
Efficiency	Group A	1.33	0.21	1.43	0.14
	Group B	1.24	0.30	1.45	0.13
	Total	1.29	0.25	1.44	0.13

4.2.4 Results

We used a multivariate analysis of variance (MANOVA) to analyze the results of the clinical history taking study. Group was used as a between factor and the Time 2 confidence, verbal accuracy, written accuracy, number of questions asked (to SP), and efficiency measures were used as dependent variables. Time 1 confidence, verbal accuracy, written accuracy, number of questions asked (to SP), and efficiency measures were used as covariates. The means and standard deviation for these variables are presented in Table 2.

The MANOVA shows that there were no significant differences between students in the CAS and no CAS group at Time 2 for any of the dependent variables tested.

Students who received additional training with the CAS had a somewhat higher increase in verbal accuracy between Time 1 (m=42.64, sd=8.91) and Time 2 (m=72.48, sd=8.75), compared to students following only traditional instruction, Time 1 (m=42.33, sd=8.92); Time 2 (m=60.93, sd=7.05). However this difference is not significant.

4.2.5 Discussion

This study tested the first prototype of the CAS and explored how the additional use of the CAS to traditional methods of teaching impacts on students' confidence, accuracy and efficiency when practising clinical history taking.

We hypothesized that students' confidence level in their performance would increase following a dedicated training period with the CAS as an additional practice tool. The results show that both groups had a similar increase in confidence. We could find no evidence that practicing with the CAS increases confidence levels.

For accuracy measures, we hypothesized that both verbal and written accuracy would increase for students in the CAS group. Results show that verbal accuracy, which represents the number of answers students were able to retrieve from the SP, increased for all students and no significant differences between groups were found. Also for written accuracy, which is the percentage of adequately reported information from the verbal accuracy scores, we could not find significant differences between groups. Students seem to perform better than in the initial assessment whether or not they had access to the simulator.

Our final hypothesis was that efficiency would improve for students who used the CAS in addition to traditional methods. We characterize efficiency with two measures: the number of questions students ask the SP in role play, and the relation between the number of questions asked and answers retrieved from the VP. The results did not show a significant effect of CAS training on the number of questions asked to SPs in role play. It seems that over time, students generally started by asking only a small amount of questions to the SP during the assessments, then experimented with asking the SP more questions during the second assessment. Our main efficiency measure, the number of questions asked divided by verbal accuracy, however, decreased between the first and the second assessment. This can be attributed to students experimenting with the number of questions they could ask SPs, as mentioned above. No significant differences between groups could be found to support our hypothesis.

Limitations

The main limitation concerning the clinical history study is the VPs' scripting itself, which was still early stage in this project. While students did have positive results using the VPs, some reported having difficulties in retrieving specific information from them. This was due to the way they asked questions to the VPs, that was not adequately recognized by the system. We believe that by improving patients' interaction-scripts, we could obtain more positive results in further studies. It is also important to mention that scripts' implementation and robustness are a crucial element in VP based systems used in those studies (Halan et al. 2010, Rizzo et al. 2010, Rossen et al. 2010). Despite not having captured data on students' frustration, it is clear that a student is unlikely to spend a large amount of time practising if he finds the task daunting due to low responsiveness from the VPs.

Another method that could be followed in further studies could be the approach designed by Rossen et al.(2010), which presents a system aiming to improve virtual agents scripts implementation. It starts by recording interactions between multiple humans taking turn interacting with a VP, before formulating an aggregate encompassing all the previous interactions within a single script. This process should be iterative, and undertaken alongside field experts (Rizzo et al. 2009). It is also a process that takes considerable time commitment in order to capture the wide range of conversational stimuli available (Halan et al. 2010).

We suggest that interaction scripts should start being tested as early on as possible when implementing a VP simulation system, to allow enough time for sufficient refinement.

A further limitation of this study was the lack of control over students' commitment to training using the CAS, and the size of the sample. The Master of Audiology is already an intensive degree and asking students to train in their free time, while dealing with an already full schedule and assessments due for the different courses they were enrolled in proved difficult. Students admitted after the study ended, during an informal meeting, that they did not feel as though they had enough practice time on the CAS. This could explain the lack of significant differences between groups for this study. Another concern was the limited number of participants available due to the small number of students enrolled in this degree.

5 Conclusion

Our current studies have found no clear statistical evidence on the effectiveness of using a VP simulator as an additional training tool for audiology trainees. However, the data suggests that there was some improvement in some measures after using the CAS. Those measures were confidence, perceived level of learning, and ability to retrieve information from SPs in a clinical role situation. Additionally, we identified areas that would benefit from being explored more deeply.

One of the main limitations of this study was the lack of control over students' commitment to training using the CAS. The Master of Audiology is already an intensive degree and asking students to train in their free time, while dealing with an already full schedule and assessments due for the different courses they were enrolled in proved difficult. Students admitted during our final group reflection meeting that they did not feel as though they had enough practice time on the CAS, especially during the study for Pure tone audiometry. This could explain the lack of significant differences between groups for this study. A further limitation concerns the clinical history study and is the VPs' scripting itself. While students did have positive results using the VPs, some reported having difficulties in retrieving specific information from them. This was due to the way they asked questions to the VPs that was not adequately recognized by the system. We believe that by improving patients' interaction-scripts, we could probably obtain more positive results in further studies. Another concern was the limited number of participants available due to the small number of students enrolled in this degree.

While we need more in depth studies to evaluate the effectiveness of the CAS as part of the Clinical Audiology course, at this stage we conclude that training with the CAS did not impact negatively on the students. Taking this into account, using the CAS as a supplementary tool for audiology trainees has a number of practical advantages. Students were able to practice at their own pace during their free time, as the computer room where the CAS was installed could be accessed 24/7. Then, within the few hours of training to which students' were exposed, they accessed 25 cases covering most of the pathologies they encounter during their professional career. This would not have happened during any of their clinical placements as patients are limited and pathologies can be redundant because students could conduct similar diagnoses throughout the day. Finally, the number of patients and opportunities for actual hands-on experience can vary greatly from one clinic to another. This results in students having very different learning experiences. The CAS, on the other hand, standardizes the training, offering the same opportunities to each student.

6 Future work

The next step of this research is to conduct a study using a similar design, while taking into consideration the lessons learned from the studies presented in this paper. Improvements will be made to the CAS according to participants' feedback gathered in the present studies.

Additionally, we plan to control more thoroughly students' practice sessions by having them book training slots before hand for both studies. The CAS will also record practice time and cases seen by students. This should allow us to confirm that students are actually training for a sufficient amount of time, and also focusing on the task at hand. We also want to investigate how many and what type of patients students meet in clinic as part of the traditional training methods. We will be investigating number of patients, the range of pathologies, and the time spent in actual hands-on work with patients. Additional work will be undertaken with field experts to diversify the VPs' recognisable vocabulary and ability to respond to a varied range of questions. Students will also be required to complete a typical Adult History form for each VP assessed in order to practice reporting relevant pieces of information from patients' interviews. Other improvements will include but are not limited to the following: adding different ways of displaying students' results in order for participants to engage more in reasoning while examining their answers, integrating a more varied range of 3D models for the VP characters, and implementing additional VPs to train with.

Finally, we will aim to explore the effect of an additional formative feedback component when using the CAS, providing students with hints while they are conducting their VP assessments. This study will be conducted on a larger sample of participants recruited from Speech and Language Therapy students to allow for more statistical power.

References

Bickmore, T. W., Pfeifer, L. M., Byron, D., Forsythe, S., Henault, L. E., Jack, B. W., Silliman, R. & Paasche-Orlow, M. K. (2010), 'Usability of conversational agents by patients with inadequate health literacy: evidence from two clinical trials.', *Journal of Health Communication* **15 Suppl 2**, 197–210.

Bickmore, T. W., Pfeifer, L. M. & Paasche-Orlow, M. K. (2009), 'Using computer agents to explain medical documents to patients with low health literacy.', *Patient Education and Counseling* **75**(3), 315–320.

Bickmore, T. W. & Picard, R. W. (2005), 'Establishing and maintaining long-term human-computer relationships', *ACM Transactions on Computer-Human Interaction* **12**(2), 293–327.

Collins, R. P. & Harden, R. M. (1998), 'AMEE Medical Education Guide No. 13: real patients, simulated patients and simulators in clinical examinations', *Medical Teacher* **20**(6), 508–521.

Conradi, E., Kavia, S., Burden, D., Rice, A., Woodham, L., Beaumont, C., Savin-Baden, M. & Poulton, T. (2009), 'Virtual patients in a virtual world: Training paramedic students for practice.', *Medical Teacher* **31**(8), 713–720.

Deladisma, A. M., Gupta, M., Kotranza, A., Bittner, J. G., Imam, T., Swinson, D., Gucwa, A., Nesbit, R., Lok, B., Pugh, C. & Lind, D. S. (2009), 'A pilot study to integrate an immersive virtual patient with a breast complaint and breast examination simulator into a surgery clerkship.', *American journal of surgery* **197**(1), 102–106.

Dickerson, R., Johnsen, K., Raij, A., Lok, B., Stevens, A., Bernard, T. & Lind, D. S. (2006), 'Virtual patients: assessment of synthesized versus recorded speech.', *Studies In Health Technology And Informatics* **119**, 114–119.

Ferdig, R. E., Coutts, J., DiPietro, J., Lok, B. & Davis, N. (2007), 'Innovative technologies for multicultural education needs', *Multicultural Education Technology Journal* **1**(1), 47–63.

Fuhrman Jr, L. C., Buff, W. E., Eaddy, M. & Dollar, M. (2001), 'Utilization of an integrated interactive virtual patient database in a Web-based environment for teaching continuity of care', *American Journal of Pharmaceutical Education* **65**(3), 271.

Gorrindo, T. & Groves, J. E. (2009), 'Computer Simulation and Virtual Reality in the Diagnosis and Treatment of Psychiatric Disorders', *Academic Psychiatry* .

Halan, S., Rossen, B., Cendan, J. & Lok, B. (2010), 'High Score ! - Motivation Strategies for User Participation in Virtual Human Development', *Seven* pp. 482–488.

Hayes-roth, B. & Saker, R. (2003), 'Next-Generation Automated Health Behavior Coaches', *Symposium A Quarterly Journal In Modern Foreign Literatures* pp. 28–34.

Heer, I. M., Middendorf, K., Müller-Egloff, S., Dugas, M. & Strauss, A. (2004), 'Ultrasound training: the virtual patient', *Ultrasound in Obstetrics and Gynecology* **24**(4), 440–444.

Johnsen, K., Dickerson, R., Raij, A. & Lok, B. (2005), 'Experiences in using immersive virtual characters to educate medical communication skills', *IEEE Proceedings VR 2005 Virtual Reality 2005* pp. 1–8.

Johnsen, K. & Lok, B. (2008), An Evaluation of Immersive Displays for Virtual Human Experiences, *in* 'Virtual Reality Conference, 2008. VR '08. IEEE', pp. 133–136.

Kenny, P. G. & Parsons, T. D. (2010), Virtual Patients for Virtual Sick Call Medical Training, *in* 'Training, Simulation, and Education Conference'.

Kenny, P. G., Parsons, T. D., Rizzo, A., Reger, G. M., Caroly, P., Pato, M., Sugar, J. & StGeorge, C. (2008), 'Virtual Patients for Future Leaders'.

Kotranza, A., Johnsen, K., Cendan, J., Miller, B., Lind, D. S. & Lok, B. (2009), Virtual multi-tools for hand and tool-based interaction with life-size virtual human agents, *in* '2009 IEEE Symposium on 3D User Interfaces', IEEE, pp. 23–30.

Rizzo, A., Parsons, T. D., Buckwalter, J. G. & Kenny, P. G. (2010), 'A New Generation of Intelligent Virtual Patients for Clinical Training pdf free ebook download from projects.ict.usc.edu'.

Rizzo, A., Patrick, K. & Parsons, T. D. (2009), 'JVRB - Journal of Virtual Reality and Broadcasting - Intelligent Virtual Patients for Training Clinical Skills'.

Rossen, B., Cendan, J. & Lok, B. (2010), 'Using Virtual Humans to Bootstrap the Creation of Other Virtual Humans', pp. 392–398.

Round, J., Conradi, E. & Poulton, T. (2009), 'Training staff to create simple interactive virtual patients: the impact on a medical and healthcare institution.', *Medical Teacher* **31**(8), 764–769.

Ruttkay, Z. & Welbergen, H. V. (2008), Elbows Higher ! Performing , Observing and Correcting Exercises by a, *in* 'Proceeding IVA '08 Proceedings of the 8th international conference on Intelligent Virtual Agents', pp. 409–416.

Stanton, T. (2008), 'Aspiring Medical Assistants Can Now Provide Care for Virtual Patients via Interactive Simulation'.

Tan, Z. S., Mulhausen, P. L., Smith, S. R. & Ruiz, J. G. (2010), 'Virtual Patients in Geriatric Education', *Gerontology & Geriatrics Education* **31**(2), 163–173.

Triola, M., Feldman, H., Kalet, A. L., Zabar, S., Kachur, E. K., Gillespie, C., Anderson, M., Griesser, C. & Lipkin, M. (2006), 'A randomized trial of teaching clinical skills using virtual and live standardized patients.', *Journal of general internal medicine* **21**(5), 424–9.

Vash, J. H., Yunesian, M., Shariati, M., Keshvari, A. & Harirchi, I. (2007), 'Virtual patients in undergraduate surgery education: a randomized controlled study.', *ANZ journal of surgery* **77**(1-2), 54–9.

VERG (2005), 'Virtual Experiences Research Group - University of Florida'.
URL: *http://verg.cise.ufl.edu/*

Towards a 3D Sketch-Based Modelling API

Yi Zeng **Zijiang Song** **Burkhard C. Wünsche**

Department of Computer Science
University of Auckland, Auckland, New Zealand,
Email: yzen015@aucklanduni.ac.nz, zson013@aucklanduni.ac.nz, burkhard@cs.auckland.ac.nz

Abstract

Sketch-based applications are rapidly gaining popularity in 3D modelling because of the intuitive pen-and-paper metaphor. Even inexperienced users with little computer graphics and digital design background can use them to create 3D models rapidly. However, the development of sketch-based applications is usually difficult and time consuming. In this paper, we present a framework for simplifying the development of sketch-based 3D modelling applications. The framework integrates existing techniques for 3D sketch processing with a processing pipeline for sketch input, a state-machine for defining processing parameters and modes, and a customised event handler. The modular design means that the functionality of the framework can be easily extended in the future. Experimental results suggest that the framework is easy to use and the implemented functionalities work correctly.

Keywords: sketch-based modelling, sketch API, sketch recognition, surface reconstruction

1 Introduction

Computer generated 3D models are common and important components of many virtual environments. They are used in a wide range of application fields including computer games, movies, medical simulations, robotics, architecture, urban design, and education. Professional modelling tools such as Maya and AutoCAD are powerful and able to construct realistic 3D model with high precision, but they have complex interfaces with a steep learning curve and are most suitable for expert users. In many applications low precision is acceptable and the emphasis is on having an intuitive modelling tool allowing untrained users to create 3D content quickly and easily.

Sketch-based modelling tools are a promising solution, since sketching is intuitive (pen-and-paper metaphor), gives complete freedom over the input, encourages creativity (Gross & Do 1996), facilitates problem solving (Wong 1992), and allows users to concentrate on the overall design of a 3D model rather than the modelling tool itself. With sketch interfaces even inexperienced users without graphics knowledge can create 3D content quickly (Yang & Wünsche 2010, Olsen et al. 2011).

Due to the rapid uptake of consumer-level touch screen devices, the number of sketch-based modelling applications has increased significantly over the past decade. However, developing a sketch-based 3D modelling application is still difficult and time consuming.

One reason for this is, that currently there is no general framework for 3D sketch processing. For each new application, developers have to spend a large amount of effort implementing their own version of the fundamental tasks in sketch processing, i.e. sketch smoothing, strokes combination, shape recognition, 3D projection etc.

In order to improve the efficiency and productivity of developing new sketch-based applications, we propose a 3D sketch-based modelling framework, which integrates common functionalities of existing 3D sketch-based modelling tools. The framework is fully extendible so that more features can be easily included.

Section 2 reviews previous work on sketch APIs and frameworks. In section 3 we review some examples of sketch-based 3D modelling applications. From these examples we identify common concepts and constraints, which are used in the requirement analysis presented in section 4. Section 5 presents the design of our framework and section 6 discusses implementation details. We evaluate the presented sketch API in section 7 and conclude the project and discuss potential future work in section 8.

2 Related Work

Despite of the popularity of 3D sketch-based modelling most existing APIs and frameworks only support 2D sketching. The arguably most widely known tools are the Microsoft Ink tools, which comprise the `Pen API` for capturing pen motion, the `Ink API` for rendering, grouping, storing and loading ink (pen motions), and the `Ink Analysis API` for handwriting recognition (Windows Dev Center 2013*a*,*b*).

Several tools have been presented, which facilitate the development of components of sketch-based modelling tools. For example, RATA simplifies the development of sketch recognisers (Plimmer et al. 2012).

A review of the literature resulted in the identification of only one API for 3D sketch-based modelling: The SketchUp Ruby API enables developers to extend the functionality of SketchUp as well as create macros to encapsulate complex tasks (Trimble Navigation Ltd. 2013). However, the API only allows interaction with traditional geometric entities, such as points, faces and meshes. The "raw" sketch input does not seems to be accessible to developers.

3 Background

We reviewed different sketch-based modelling applications and identified the following important components:

3.1 2D Sketch Processing

Rendered sketches look more attractive and analysis of 2D sketches and creation of 3D surfaces is made easier when using a simple mathematical representation for them.

Igarashi et al. evaluate each stroke input for potential geometric relations such as horizontal and vertical strokes, connections, alignments, and symmetry (Igarashi et al. 1997). Interactive beautification is performed after identifying the most suitable geometry relation.

Sezgin et al. eliminate noise from free-hand drawings by combining average based filtering and scale space filtering (Sezgin et al. 2001) . The method uses curvature information and pen speed data in order to differentiate between shape features of a curve and unintended wriggles.

A smooth curve can be obtained from sketch input by reducing the number of samples with the Douglas-Peucker algorithm and interpolating the resulting samples using a Catmull-Rom spline (Wünsche 2013).

3.2 Sketch Recognition

Creation of 3D geometry often requires knowledge of the type of shape a 2D sketch represents. Barber et al. use primitive shape properties such as length, oriented bounding box (OBB), curvature, direction changes, length-area ratio etc. in order to recognise geometric shapes such as straight lines, triangles, rectangles, and closed curves (Barber et al. 2010). Plimmer et al. use machine learning algorithms to develop sketch-recognisers (Plimmer et al. 2012).

3.3 3D Geometry Creation

3.3.1 Silhouette-Based Methods

The arguably most popular class of sketch-based 3D modelling techniques uses sketch input to represent the silhouette (outline) of a 3D object. The outline, usually referred to as contour, is expanded to a 3D object by making the assumption that the object is "blobby", i.e., the cross section of each component of the sketched contour is circular.

The 3D surface can be obtained by computing a skeleton of the contour and then fitting circular cross-sections around the skeleton (Igarashi et al. 1999, Igarashi & Hughes 2003, Levet & Granier 2007). More complex shapes can be obtained by sketching contours of local features (Zimmermann et al. 2008). Alternatively implicit surfaces can be used to convert contours to 3D bodies (Karpenko et al. 2002, Schmidt et al. 2006, de Araújo et al. 2004).

Two interesting application of silhouette-based algorithms are garment and tree modelling. For tree modelling the user sketches the outline of the crown of the tree and the algorithm computes a fitting branching structure based on existing templates and a probabilistic distribution (Chen et al. 2008). Garments can be modelled by sketching their outline and the algorithm automatically fits them to the body shape (Turquin et al. 2007).

3.3.2 Contour-Based Methods

Contours include all visible lines and divisions of a shape, e.g., discontinuities in the surface gradient. Contour information can be used to edit 3D meshes by making local modifications (Karpenko & Hughes 2006), or by using free form deformations to adapt the underlying 3D shape (Nealen et al. 2007). If domain specific information is know complete shapes can be obtained using only a few input sketches. For example, Gain et al. model complex 3D terrains by drawing the silhouette, spine and bounding curves of landforms (Gain et al. 2009).

A popular application of contour-based methods is the sketching of technical drawings and 3D CAD models. Computer designed items are often characterised by a blocky shape, flat or arced surfaces, sharp or evenly rounded edges and corners, many parallel and orthogonal edges and faces, and symmetrical features. These features can be captured using silhouettes which can then be interpreted using application specific constraints, e.g., surfaces of CAD objects frequently form 90 degree angles (Zeleznik et al. 1996, Eggli et al. 1997, Mitani et al. 2002).

3.3.3 Skeleton-Based Methods

Skeleton-based techniques are most popular for modelling complex fibrous and branching structures. Ijiri et al. (2005) use sketch input to model the stem and branches of flowers. The 3D shape of a stem is computed by solving a differential equation such that the curvature and appearance of the resulting 3D shape is identical to the 2D sketch.

A more general system is "Thor" (Arcila et al. 2008). The user draws a skeleton using a series of sketches. The initial sketch defines the main shape and subsequent sketches modify it. The user can draw a radius for each skeleton segment and a 3D surface is generated by fitting a generalised cylinder to it.

3.3.4 Cross Section-Based Methods

A less common approach for creating 3D models is to sketch 2D cross-sections. McCord et al. (2008) model orchids by allowing users to sketch the cross section of the labellum of an orchid, which is then expanded into a 3D surface by fitting ellipsoidal contours around.

Cross sections are used in most professional modelling tools, but are usually not sketched but represented by parametric curves. 3D surfaces are obtained by extrusion or by computing the tensor product with a second parametric shape. SketchUp employs some of these principles using an interface mixing sketch and CAD elements (Trimble Navigation Ltd. 2013).

Olsen et al. model 3D buildings from two-dimensional sketched cross-sections. The algorithm analyses the sketch input, extracts shape and detail information, predicts the building type, and creates 3D models by applying an extrusion, rotation of a projection algorithm (Olsen et al. 2011)

3.3.5 General Sketch-Based Modelling Tools

ILoveSketch is a 3D curve sketching system where users are able to draw curves freely in 3D space (Bae et al. 2008). Several tools are provided to select a drawing plane/surface, e.g., coordinate planes from user-defined coordinate systems and planes obtained by extruding a sketched curve. 3D curves can also be obtained from two 2D curves using epipolar geometry.

3.4 Sketched-Based Animation Systems

Sketch-based animation of objects can be achieved by sketching motion paths, which define the motion of the entire object (Steger 2004) or the motions of components of an object such as limbs (Schauwecker et al. 2011). Motion paths can also be subdivided into primitives and matched to pre-animated motions (Thorne et al. 2004).

An alternative solution is to sketch key frame poses, and use bone information (Davis et al. 2003) or body contour information (Mao et al. 2007) to infer 3D motions.

4 Requirements Analysis

Our analysis of successful sketch-based modelling systems for 3D objects shows that virtually all applications use the following steps:

- **Sketch simplification and beautification:** in order to analyse and interpret sketch input it has to be converted into a simplified mathematical form, e.g., a polyline or smooth curve.
- **Sketch recognition:** in order to create 3D geometry from 2D geometry, the geometric properties of input sketches must be identified, e.g., "sketch is a straight line", "sketch is a closed curve" or "sketch is a rectangle".
- **Context:** The interpretation of a 2D sketch often depends on the context. For example, a sketch drawn over a surface can indicate deformation of the surface, whereas a sketch drawn touching a surface can indicate an extrusion process.
- **Constraints:** Limiting the range of possible 3D shapes makes it easier to recognise and interpret 2D sketch input.

Our goal is to develop a 3D sketch API which simplifies the development of a large range of sketch-based modelling applications. We hence need the following functionalities:

- Functions for sketch simplification and beautification.
- Functions for sketch recognition.
- Functions to set context, e.g., modelling/edit/interaction mode, definition of sketch planes.
- Functions to select objects, e.g., closest existing sketch to a new sketch, closest 3D object to a sketch.
- Functions to modify objects, e.g., deformation of a surface using sketch input.
- Functions to derive 3D sketches from 2D sketch input, e.g., use depth values from a 3D object or sketch plane.

5 Design

In order to fulfill the identified requirements developers must be able to specify each stage of the sketch input processing and model generation. Developers must be able to store intermediate objects, such as surfaces defined from sketch input, and use them in subsequent interactions (e.g., deform a surface using sketch input).

Application developers must be able to define different functionalities depending on user input. We hence need to provide event handling. The event handler evaluates each sketch and generates corresponding events, e.g., if the drawing of a sketch was completed, a closed sketch was detected, or a sketch was drawn over an existing object.

In order to minimise the developer's workload and reduce code redundancies we provide functions to set states, similar to the OpenGL graphics API. For example, the developer can select a function for sketch simplification and all subsequent sketches are processed accordingly, until that function is changed.

Our Sketch API consists hence of a sketch processing pipeline, a state machine, event handler, and object database as illustrated in figure 1.

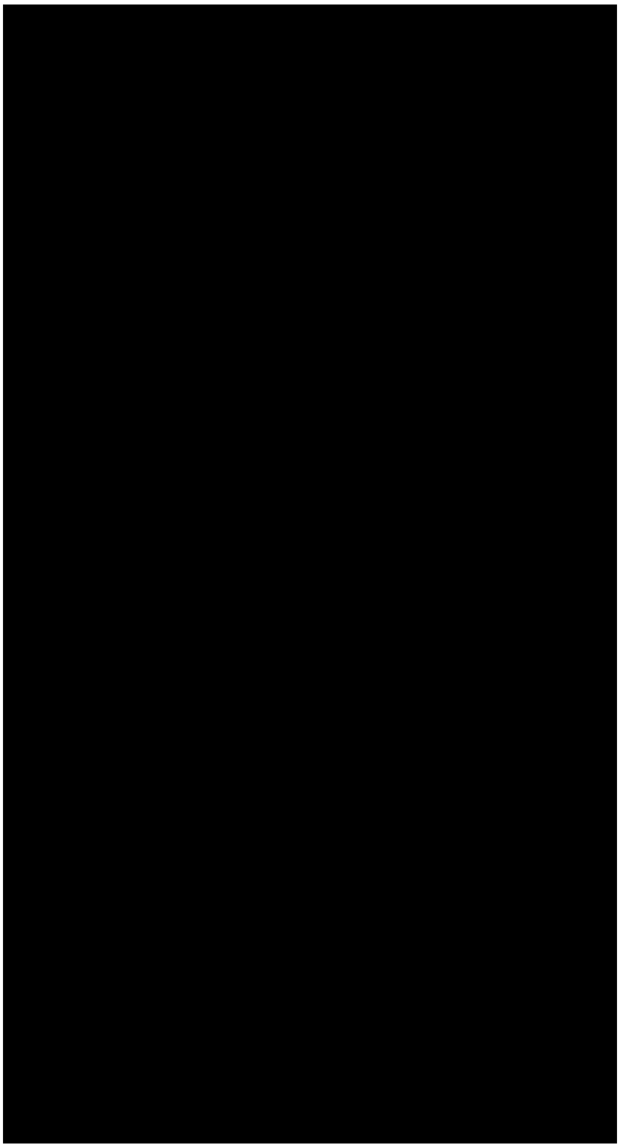


Figure 1: The Sketch API consists of a sketch processing pipeline, and a state machine, event handler, and object database.

5.1 Sketch Processing Pipeline

The core of our framework is the sketch processing pipeline shown on the right-hand side of figure 1. Whenever a sketch is detected, it is processed through the pipeline. Each stage is configurable via setting an appropriate mode and can be enabled or disabled. For example, if a developer wants to simplify sketch input using the Douglas-Peucker algorithm and then project the 2D sketch onto the predefined 3D drawing plane, then this can be achieved using only three lines of code:

```
setSimplificationMode(DOUGLAS_PEUCKER);
setProjectionMode(PLANE);
setPipelineMode(PROJECTION |
                SIMPLIFICATION);
```

5.1.1 Sketch Input

In order to make the API as flexible as possible it should work for different input devices such as mouse, touch screen, and drawing tablets. We hence use an extra layer of abstraction, which converts device specific input into a format suitable for our API. For example, when using the mouse as input device mouse events such as "mouse up" and "mouse down" are converted into "sketchStart" and "sketchEnd" events and the mouse coordinates between any such event sequence are converted into a sequence of 2D points with duplicates removed. When using a different input device only this abstraction layer needs to be modified.

At this point we do not yet record additional parameters such as "drawing speed" and "pen pressure" (which could be simulated with the mouse wheel).

5.1.2 Sketch Simplification

Sketches often contain unintended jags and other errors. In order to simplify the sketch input we allow developers to use the Douglas-Peucker algorithm (Douglas & Peucker 1973). Sketches are smoothed using Catmull-Rom spline interpolation (Catmull & Rom 1974).

5.1.3 3D Projection

Finding the correct 3D positions for a 2D input sketch is arguably the most important step in 3D sketch-based modelling. In our framework, we find suitable z-coordinates for a 2D sketch by projecting the sketch onto user-defined planes and surfaces. Currently we support the following 3D mapping modes:

- **3D Plane:** The sketch is projected on a specified 3D drawing plane. The drawing plane can be defined by the developer or can be selected by the user through the built in "Plane Selector" widget.
- **Extruded Surface:** The sketch is projected on a 3D extruded surface, which is obtained by extruding a sketch along a vector. The surface can be specified by the developer, e.g., a curved sketch followed by a straight line connected to it. Alternatively an extruded surface can be specified at run time by the user by using the built-in "Surface Selector".
- **Arbitrary Surface:** The sketch is projected onto an arbitrary 3D surface by using the surface's z-coordinates.

More details are given in section 6.

5.1.4 Sketch Assembly

The sketch assembly step provides functions to group sketches, e.g., combine multiple strokes into a single sketch, close a sketch, or form meaningful shapes such as arrows.

5.1.5 3D Sketch Recognition

In this stage the assembled sketch is classified into a set of predefined shapes, such as rectangle, triangle, circle, straight line, scribble etc. The developer can enable automatic beautification of shapes, e.g., replacing a sketched circle with a perfect parametric circle. Our current implementation contains a simple approximation for this, but better algorithms for finding the optimal fitting geometric shape have been described in the literature (Arvo & Novins 2000).

5.1.6 3D Object Generation

The last step in the processing pipeline is the generation of 3D surfaces and objects from the input sketch. Currently, the framework supports the following modes:

- **Extrusion:** Extrude a 3D stroke in a specified direction.
- **Filling:** Create a parametric surface from a closed sketch.

5.2 Event Handling

Developers can specify complex functionalities by using sketch events. The current prototype supports the following events:

- **Sketch Begin:** Triggered when a sketch begins.
- **Sketching:** Triggered repeatedly as long as a sketch is still drawn.
- **Sketch End:** Triggered when a sketch ends.
- **Sketch Closed:** Triggered when a closed sketch is detected.
- **Drawing Plane Changed:** Triggered when the drawing plane is changed.

In order to prevent conflicts with the event handler of the underlying graphics library (GLUT) we create our own mouse callback functions which the developer can call using `sketchMouseFunc(handler)`.

6 Implementation

In this section we explain some of the key functionalities in more detail.

Figure 2: Projection of a 2D sketch onto a plane in 3D.

6.1 3D Projection

Sketch raw data consists of 2D screen coordinates. The 3D coordinates of a 2D sketch are determined as follows:

If the current OpenGL state associates the sketch with a sketch plane in 3D, we cast rays from the view point through the sketche's screen coordinates and compute the intersection points with the sketch plane. Figure 2 shows an example.

An extruded surface is constructed by extruding a 3D sketch along a 3D vector. If the current OpenGL state associates the sketch with such an extruded surface we can compute its 3D coordinates similar as above. This is possible since a 3D sketch is approximated by a sequence of line segments and the resulting extruded surface is hence a quadstrip (sequence of rectangles). Figure 3 shows an example.

In many instances we want to sketch on an arbitrary 3D object. In order to make the algorithm as general as possible our only requirement is that the 3D object can be rendered with depth-buffer values. Examples are polygon meshes, dense point clouds, or

Figure 3: Projection of a 2D sketch onto an extruded surface.

ray traced implicit surfaces. If the current OpenGL state associates the 2D sketch with such an object, we can compute its 3D representation by rendering the 3D object and retrieving for each 2D coordinate of the sketch the corresponding depth buffer value. Figure 4 shows an example.

Figure 4: Projection of a 2D sketch onto a renderable 3D object.

6.2 Shape Recognition

Our simple shape recogniser can currently detect rectangles, circles and scribbles (for deleting a sketch).

Rectangles are identified by first detecting whether the sketch is a closed shape (i.e., end points are close together relative to the bounding box size), simplifying the sketch with the Douglas-Peucker algorithm using a high error value obtained from the bounding box size, and then computing the number of turning points.

Circles are identified by first detecting whether the sketch is a closed shape, computing the centre of the bounding box, and checking whether all sketch points have a roughly equal distance to the centre point.

A scribble is characterised by an approximately equal number of significant turning points (angle $\geq 120°$). Whenever a scribble is detected, the sketches covered by the bounding box of the scribble are immediately deleted.

Figure 5 shows an example.

6.3 NURBS Surface from Closed Sketch

NURBS surfaces are common in 3D modelling applications since they can be easily controlled using a control point mesh and knot vector, they have a high-level of continuity (smoothness), discontinuities can be inserted if desired, they have local control, and they are supported by most graphics APIs such as OpenGL.

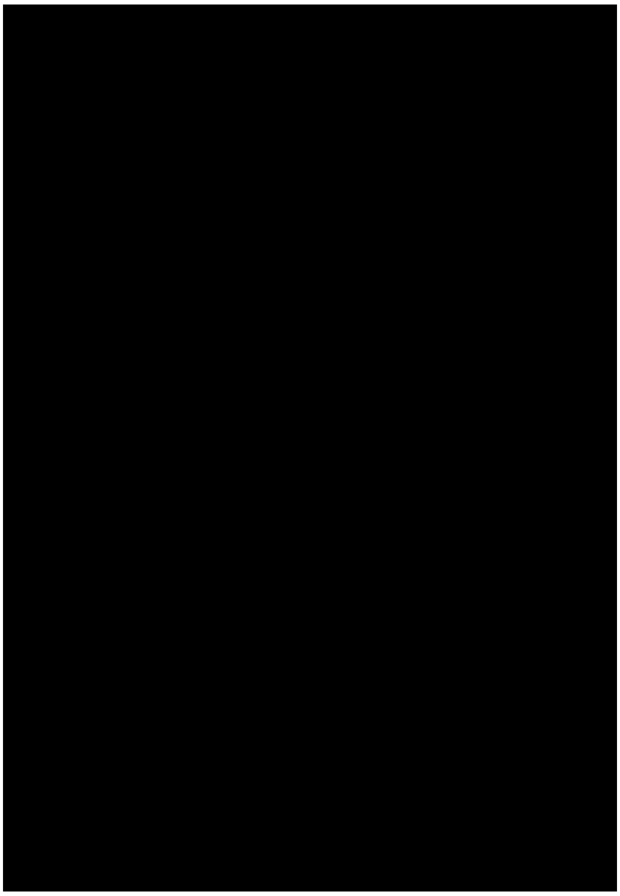


Figure 5: Recognition and beautification of a rectangle (top) and circle (bottom) and a scribble for deleting sketched objects (bottom).

We create a NURBS surface from a closed sketch by computing its oriented bounding box, using it to define the control point mesh, and then trimming the resulting surface using the sketch such that only the surface section inside the sketch remains (see figure 6). The resulting surface can then be modified using sketch input by adjusting control points accordingly.

Figure 6: A NURBS surface defined by a closed sketch.

7 Results

In this section we evaluate the effectiveness and robustness of the presented sketch API.

7.1 Effectiveness

The primary goal of our Sketch API is to enable developers to easily create 3D sketch-based modelling tools. In order to evaluate our tool we implemented a simple tool for sketching 3D leaves. The modelled functionality has been previously been presented in different flower modelling tools (Ijiri et al. 2005, McCord et al. 2008). A leaf is sketched in three steps as illustrated in figure 7.

Figure 7: A leaf is sketched in three steps: (1) Sketch the outline of the leaf using one or two strokes and fit a NURBS surface to the closed sketch (left); (2) sketch a modifier stroke and project it onto the NURBS surface (middle); (3) compute the distance between the modifier stroke and the leave's centre line and warp the NURBS surface in the direction of the surface normal accordingly.

Using our framework the implementation of this functionality is straight forward.

The sketch processing pipeline requires only the stage 3D Projection and Sketch Assembly. 3D Projection is used to project the sketch onto the drawing plane, and Sketch Assembly is needed to connect multiple strokes forming the leaf. We also need to set up callback functions for sketch events. The resulting code is:

```
// sketch pipeline
setPipelineMode(SKETCH_PROJECTION |
                SKETCH_ASSEMBLY);
setProjectionMode(PROJECTION_PLANE);
setAssemblyMode(ASSEMBLY_ENDPOINS);
// callbacks
onSketchEnd(handleSketchEnd);
onClosedShapeDetected(handleClosedShape);
```

These handlers are called when corresponding sketch events are triggered. In our example a NURBS surface is created when a closed shape (leaf's outline) is drawn, and the surface is deformed in 3D space when a modifier stroke is drawn. The resulting code is:

```
void handleClosedShape(Stroke& stroke) {
  // when a closed sketch is detected
  myNURBS = new NURBSSurface(stroke);
  bClosedShapeDrawn = true;
  setPipelineMode(SKETCH_PROJECTION);
}

void handleSketchEnd(Stroke& stroke) {
  // when a stroke is completed
  if (bClosedShapeDrawn)
    myNURBS->deformSurface(stroke);
}
```

The last step is to draw the resulting NURBS surface:

```
if (myNURBS)
  myNURBS->drawSurface();
sketchDisplay();
```

Our results so far indicate that the framework is easy to use and suitable for a wide range of applications. Functionalities are currently very limited, but new ones are added each time we use the tool for novel application. For example, we currently work on using sketch input to complete 3D models obtained from point cloud data.

7.2 Robustness

In the current version of the framework, most of the implemented functionalities are working correctly, such as sketch simplification, projecting sketches onto a drawing plane, creating NURBS surface etc. However, the implementations of some functionalities are not robust, and the processing pipeline can fail as a result. We have identified two issues below:

Projection on Arbitrary Surface

This functionality is achieved by utilising the OpenGL depth buffer. In orthographic projection mode, the algorithm works correctly, because the depth value is a linear function, which means the depth value is accurate in all depth ranges. However, when using a perspective projection problems can occur if the near plane is set too close to the camera. In this case the depth buffer values form a non-linear function and pixels representing objects close to the near plane have a z-value with high precision, and pixels representing objects close to the far plane have a z-value with low precision. In the latter case the 3D coordinates of a sketch drawn over such an object are very inaccurate.

Shape Recognition

The shape recogniser uses a variety of threshold values, e.g., to determine whether a sketch is closed or whether it is a rectangle. We tried to make threshold values work correctly for a large variety of shapes by taking into account the size of a shape. However, many of these decisions are subjective and application dependent, i.e., what is a closed curve to one user might be an open curve to another one. Allowing the developer or user to set these parameters is not desirable, since it would significantly increase the complexity of the tool. A possible solution is to use a machine learning algorithm similar to (Plimmer et al. 2012).

8 Conclusion and Future Work

Sketch-based modelling is an exciting technology with a wide range of applications. By reviewing the current state-of-the-art and evaluating a variety of existing sketch-based modelling applications, we have designed and implemented a framework for 3D sketch-based modelling which integrates basic functionalities of 3D sketch processing.

We have tested the framework by using it to write a simple sketch-based modelling applications. Preliminary results suggest that the framework is easy to use, the implemented functionalities work correctly, and that it can be easily extended.

Future work will concentrate on adding more functionalities, improving the processing pipeline, and performing more extensive usability testing with more complex application scenarios and participants unfamiliar with the tool.

References

Arcila, R., Levet, F. & Schlick, C. (2008), Thor: Sketch-based 3d modeling by skeletons, *in* 'Smart Graphics', pp. 232–238.

Arvo, J. & Novins, K. (2000), Fluid sketches: continuous recognition and morphing of simple hand-drawn shapes, *in* 'Proceedings of the 13th annual ACM symposium on User interface software and technology (UIST '00)', ACM, pp. 73–80.

Bae, S.-H., Balakrishnan, R. & Singh, K. (2008), Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models, *in* 'Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08)', ACM, pp. 151–160.

Barber, C. M., Shucksmith, R. J., MacDonald, B. & Wünsche, B. C. (2010), Sketch-based robot programming, *in* 'Proceedings of Image and Vision Computing New Zealand (IVCNZ 2010)', pp. 1–8.

Catmull, E. & Rom, R. (1974), 'A class of local interpolating splines', *Computer Aided Geometric Design* pp. 317–326.

Chen, X., Neubert, B., Xu, Y.-Q., Deussen, O. & Kang, S. B. (2008), Sketch-based tree modeling using markov random field, *in* 'ACM SIGGRAPH Asia 2008 papers', ACM, pp. 1–9.

Davis, J., Agrawala, M., Chuang, E., Popović, Z. & Salesin, D. (2003), A sketching interface for articulated figure animation, *in* 'SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation', Eurographics Association, pp. 320–328.

de Araújo, B., Jorge, J., Sousa, M. C., Samavati, F. & Wyvill, B. (2004), MIBlob: a tool for medical visualization and modelling using sketches, *in* 'SIGGRAPH '04: Posters', ACM Press, p. 107.

Douglas, D. & Peucker, T. (1973), 'Algorithms for the reduction of the number of points required to represent a digitized line or its caricature', *The Canadian Cartographer* **10**(2), 112–122.

Eggli, L., Ching-Yao, H., Bruderlin, B. D. & Elber, G. (1997), 'Inferring 3d models from freehand sketches and constraints', *Computer-Aided Design* **29**(2), 101–112.

Gain, J., Marais, P. & Strasser, W. (2009), Terrain sketching, *in* 'I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games', ACM, pp. 31–38.

Gross, M. D. & Do, E. Y.-L. (1996), Ambiguous intentions: a paper-like interface for creative design, *in* 'Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST '96)', ACM, New York, NY, USA, pp. 183–192.

Igarashi, T. & Hughes, J. F. (2003), Smooth meshes for sketch-based freeform modeling, *in* 'I3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics', ACM, pp. 139–142.

Igarashi, T., Matsuoka, S., Kawachiya, S. & Tanaka, H. (1997), Interactive beautification: a technique for rapid geometric design, *in* 'Proceedings of the Symposium on User Interface Software and Technology', pp. 105–114.

Igarashi, T., Matsuoka, S. & Tanaka, H. (1999), Teddy: a sketching interface for 3d freeform design, *in* 'Proceedings of SIGGRAPH '99', ACM Press, pp. 409–416.

Ijiri, T., Owada, S., Okabe, M. & Igarashi, T. (2005), 'Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints', *ACM Trans. Graph.* **24**(3), 720–726.

Karpenko, O. A. & Hughes, J. F. (2006), 'Smoothsketch: 3d free-form shapes from complex sketches', *ACM Transactions on Graphics* **25**(3), 589–598.

Karpenko, O., Hughes, J. & Raskar, R. (2002), 'Free-form sketching with variational implicit surfaces', *Computer Graphics Forum* **21**(3), 585–594.

Levet, F. & Granier, X. (2007), Improved skeleton extraction and surface generation for sketch-based modeling, *in* 'GI '07: Proceedings of Graphics Interface 2007', ACM, pp. 27–33.

Mao, C., Qin, S. F. & Wright, D. (2007), Sketch-based virtual human modelling and animation, *in* 'SG '07: Proceedings of the 8th international symposium on Smart Graphics', Springer-Verlag, pp. 220–223.

McCord, G., Wünsche, B. C., Plimmer, B., Gilbert, G. & Hirsch, C. (2008), A pen and paper metaphor for orchid modeling, *in* 'Proceedings of the 3rd International Conference on Computer Graphics Theory and Applications (GRAPP 2008)', pp. 119–124.

Mitani, J., Suzuki, H. & Kimura, F. (2002), '3d sketch: sketch-based model reconstruction and rendering', pp. 85–98.

Nealen, A., Igarashi, T., Sorkine, O. & Alexa, M. (2007), Fibermesh: designing freeform surfaces with 3d curves, *in* 'SIGGRAPH '07: ACM SIGGRAPH 2007 papers', ACM, p. 41.

Olsen, D. J., Pitman, N. D., Basak, S. & Wünsche, B. C. (2011), Sketch-based building modelling, *in* 'Proceedings of GRAPP 2011', pp. 119–124.

Plimmer, B., Blagojevic, R., Chang, S. H.-H., Schmieder, P. & Zhen, J. S. (2012), RATA: codeless generation of gesture recognizers, *in* 'Proceedings of the 26th Annual BCS Interaction Specialist Group Conference on People and Computers (BCS-HCI '12)', British Computer Society, pp. 137–146.

Schauwecker, K., van den Hurk, S., Yuen, W. & Wünsche, B. (2011), Sketched interaction metaphors for character animation, *in* 'Proceedings of GRAPP 2011', pp. 247–252.

Schmidt, R., Wyvill, B., Sousa, M. C. & Jorge, J. A. (2006), Shapeshop: sketch-based solid modeling with blobtrees, *in* 'SIGGRAPH '06: ACM SIGGRAPH 2006 Courses', ACM, p. 14.

Sezgin, T. M., Stahovich, T. & Davis, R. (2001), Sketch based interfaces: Early processing for sketch understanding, *in* 'Proceedings of the Workshop for Perceptive User Interfaces (PUI 01)', ACM Press, pp. 1–8.

Steger, E. (2004), Sketch-based animation language, Technical report, Department of Computer Science, University of Toronto. URL: `http://www.cs.toronto.edu/~esteger/sketchlang/index.html`.

Thorne, M., Burke, D. & van de Panne, M. (2004), 'Motion doodles: an interface for sketching character motion', *ACM Trans. Graph.* **23**(3), 424–431.

Trimble Navigation Ltd. (2013), 'Sketchup ruby api - creating geometry'. http://www.sketchup.com/intl/en/developer/docs/gsrubyapi_examples, Last retrieved 24th August 2013.

Turquin, E., Wither, J., Boissieux, L., Cani, M.-P. & Hughes, J. F. (2007), 'A sketch-based interface for clothing virtual characters', *IEEE Comput. Graph. Appl.* **27**(1), 72–81.

Windows Dev Center (2013*a*), 'Ink analysis overview'. http://msdn.microsoft.com/en-us/library/windows/desktop/ms704040\%28v=vs.85\%29.aspx, Last retrieved 24th August 2013.

Windows Dev Center (2013*b*), 'Microsoft.ink'. http://msdn.microsoft.com/en-us/library/ms826516.aspx, Last retrieved 24th August 2013.

Wong, Y. Y. (1992), Rough and ready prototypes: lessons from graphic design, *in* 'Posters and short talks of the 1992 SIGCHI conference on Human factors in computing systems (CHI '92)', ACM, New York, NY, USA, pp. 83–84.

Wünsche, B. C. (2013), 'Compsci 373 assignment 2 sammple solution - question 3: Sketch-based modelling'. http://www.cs.auckland.ac.nz/courses/compsci373s1c/assignments/CS373Assignment2_SampleSolution.pdf, Last retrieved 24th August 2013.

Yang, R. & Wünsche, B. C. (2010), Life-sketch: a framework for sketch-based modelling and animation of 3d objects, *in* 'Proceedings of the Eleventh Australasian Conference on User Interface (AUIC '10)', Australian Computer Society, Inc., pp. 61–70.

Zeleznik, R. C., Herndon, K. P. & Hughes, J. F. (1996), SKETCH: An interface for sketching 3d scenes, *in* 'Proceedings of SIGGRAPH '96', ACM Press, pp. 163–170.

Zimmermann, J., Nealen, A. & Alexa, M. (2008), 'Sketching contours', *Computers & Graphics* **32**(5), 486–499.

Ephemeral Interaction Using Everyday Objects

James A. Walsh, Stewart von Itzstein and Bruce H. Thomas
School of Computer and Information Science
University of South Australia
Mawson Lakes Boulevard, Mawson Lakes, South Australia, 5095

james.walsh@setoreaustralia.com, stewart.vonitzstein@unisa.edu.au, bruce.thomas@unisa.edu.au

Abstract

The ability for Tangible User Interfaces to enable the intuitive control of existing systems and adapt to individual users' usage scenarios remains an area of development. Previous research in customizable tangible interfaces has focused primarily on the offline creation by the original system developer, instead of offering extensibility to the end user. This paper presents our system to support the ad-hoc creation of 'disposable' UIs using both projected controls and physical objects. To support these controls, a software based patch panel enables data to be mapped to external systems, and from external systems back to the system itself. Using a projector, depth camera and 6DOF tracking system, users can create and map tangible/touch-based ad-hoc user controls to existing system functionality. This allows users to both quickly create new inputs for existing functionality, as well as create new arbitrary input devices from completely passive components.

Keywords: user interfaces, ephemeral, tangible, projected, extensible customizable, reconfigurable.

1 Introduction

Following the concept of Ephemeral User Interfaces (EUI) (Doring et al., 2013) as a temporary means of communication, we extend this concept to allow the user to construct disposable (ad-hoc) UIs to control existing systems and applications using physical objects (tangibles) and projected content in the environment. These temporary UI's are designed to support the creation of UIs to control a subset of a system's existing functionality for short term usage (minutes to hours). For example, when cooking in the kitchen, users often need to quickly create a timer based on the current recipe's task. This could easily be done by allowing the user to rotate a kitchen utensil on the counter to set the time. A projector displays the time, with adjustments made by further rotations. This leverages the affordances of objects available in a natural, tangible interaction.

The motivation for these interfaces comes from the need to enable a system with a known set of functions to adapt to the context and capabilities of the user at time of use, something the original designer cannot envision. For the kitchen timer scenario, we know that a timer will be required, but not the parameters or the context, given the user may have limited space or be limited to one handed interaction. Similarly, someone sitting on the couch can quickly draw a line armrest to control the volume/channel whilst resting their arm on the arm rest. As such, there is a need for ad-hoc controls that enable users to rapidly create controls based on the current context, ideally leveraging the affordances of the immediate environment. Previous touch-based systems (Akaoka et al., 2010, Henderson and Feiner, 2008, Xiao et al., 2013), have shown a need for ad-hoc interaction, but were designed for developers, excluded tangible interactions and did not support the integration with existing systems.

Whilst previous work (Akaoka et al., 2010, Avrahami and Hudson, 2002) has looked at creating interactive prototypes from passive materials, all interaction has been touch based, ignoring the geometric and spatial relationships that Tangible User Interfaces (TUIs) (Ishii and Ullmer, 1997) offer. Our work focuses on the end user, enabling them to utilize the proxemic relationships (position, orientation, visibility, etc.) of objects available in the immediate environment, as well as touch interaction as a means of input. We believe TUIs to be a key component in ad-hoc interaction that is yet to be explored.

In this paper, we explore the application of non-traditional tangible interaction to ad-hoc EUIs. We are interested in the use of everyday objects as props for supporting TUIs and the use of projectors to augment the user's workspace. Through the use of a software patch panel, such as Ballagas et al. (2004), we can isolate the UI from the application being controlled. This allows information to flow from the user defined UI, through the patch panel to the end application. This means the system to create the UIs and the systems/functions to be controlled can be developed independently. This paper makes the following contributions:

- we present extensions to an existing TUI architecture to support ephemeral touch and tangible UIs, including support for incorporating input from external systems,
- a paradigm to support ephemeral UIs for a wide range of existing GUI input controls, and
- a mechanism to develop ephemeral input devices from passive components using touch and geometric relationships.

The first contribution addresses the core system design to support ad-hoc controls and interactivity, including integration with existing arbitrary systems. We present an implementation of a software patch panel for our architecture that is capable of passing parameters to individual applications provided by the end user. This allows the user to integrate the ad-hoc controls with any existing system. This patch panel also allows inputs to be passed into the system from external systems, allowing the incorporation of input methods not supported by the original system, as well as support for feedback loops.

The second contribution describes a comprehensive set of example interactions to control existing applications. These are based on the existing tangible controls and the results of a preliminary study.

The final contribution outlines methods for allowing the user to create interactive, novel tangible UIs on-the-fly from passive physical components that utilize both touch and the geometric relationships both of-and-between objects. This allows the user to create disposable controls, on demand.

Following this, our work focuses on 'what' functions to perform, not 'how' to do them, extending our previous work in tangible programming by demonstration (Walsh et al., 2013). We acknowledge a number of our concepts require ubiquitous and highly portable sensing and projection technologies. In this paper, we are only focused on the architecture and methods to support interactions.

The remainder of this paper is structured as follows: related work is discussed, identifying related projects and influential factors. A summary of the system and how it is used is provided, providing a number of example applications. A preliminary study that served as the initial design phase is then described, leading into a description of the system implementation and design. We then conclude with future work and final thoughts.

2 Related Work

Our work follows previous human computer interaction work relating to TUIs, reconfigurable UIs, and their supporting architectures.

Doring et al. (2013) presented the ideas of EUIs as UIs that have at least one element designed for limited time use. They defined a design space for EUIs incorporating a) materials, b) interactions (input vs. output) and c) aspects of ephemerality. Using this design space, our work is focused on selecting the right material for the job (a), primarily as a form of input (b). Their exploration of ephemerality came from the materials used (fog, ice, jelly, etc.). We however utilize multiple objects that when together, serve as an appropriate input EUI, but when split apart fulfil their original roles (c). *Despite objects being persistent on their own, it is their utility together that is ephemeral.*

2.1 Tangible UIs

TUIs utilize the affordances of physical objects, spaces and surfaces as an interface to digital functionality (Ishii and Ullmer, 1997). Fitzmaurice et al. (1995) began exploring TUIs as Graspable UIs, using 6DOF tracked 'bricks' to manipulate digital elements This allowed users to explore the advantages of bi-manual, spatial interaction with digital functionality. Despite the nature of ad-hoc interaction meaning we are surrounded by tangible objects, previous work has failed to leverage TUIs on an equal level to ad-hoc touch interaction.

The embodiment of TUIs led to the creation of Organic User Interfaces (OUIs) (Holman and Vertegaal, 2008), exploring non-planer displays that are both input and output. This embodiment blurs the distinction between input and output and closely mirrors the feedback loop that we experience in the real world with cause and effect (Sharlin et al., 2004).

Ullmer (2002) proposed a TUI architecture equivalent to the GUI Model-View-Controller (MVC) architecture, identifying three categories of TUIs; interactive surfaces, constructive assemblies, and Tokens and Constraint (TAC). These TACs utilized the unique affordances of individual objects as logical constraints on the object. For example, an elongated groove suggests placing an object in that groove to assign a value across a range. It is these kinds of affordances that this work hopes to leverage.

2.2 Reconfigurable TUIs

Akaoka et al. (2010) explored the creation of active prototypes from passive materials as DisplayObjects. Using markers to track a passive object, predesigned virtual content (buttons, displays, etc.), designated as inputs or outputs, can be dragged from a Physical-Virtual palette onto the object. Pressing 'Play' on the palette allowed users to interact with the device. More intricate interactions between input and output controls was possible using the computer to generate scripts.

Avrahami, and Hudson (Avrahami and Hudson, 2002) used push-pin enabled RFID buttons and sliders to prototype input devices, enabling reconfiguration of physical inputs for non-planar surfaces. Building on this, the BOXES project (Hudson and Mankoff, 2006) looked at using thumbtacks attached to a circuit board to trigger user-defined macros. Upon touching a thumbtack, the software could emulate a touch at a given screen coordinate or simulate any number of predefined mouse/keyboard inputs, essentially defining a macro. Using the tacks with cardboard and tape allowed users to quickly prototype button based interaction on physical prototypes.

Both Phidgets (Greenberg and Fitchett, 2001) and VoodooIO (Villar et al., 2006) explored configurable component based UIs. Both systems offered a number of input controls and could be reused and repositioned, with Phidgets using cables to connect to a PC and VoodooIO using push-pin components to link to a conductive communication layer in a foam substrate. The processing of the input into system functionality was an offline process done by the developer. Whilst both enabled ad-hoc reconfiguration, the user was limited by components for which they do not have an input device.

In exploring touch-based UIs, Light Widgets (Fails and Olsen, 2002) explored ubiquitous touch interaction using cheap, pervasive cameras. Using a PC application to select an input type and a region on a camera's viewport for the control to be located, users could touch that location to interact. Aside from the offline creation of the UI controls, there was no feedback to the user aside from whatever function was being controlled by that input. Tangible interaction outside of touch was also not supported.

Henderson and Feiner explored Opportunistic Controls (OCs) (Henderson and Feiner, 2008) to enable natural navigation of situated Augmented Reality (AR) systems, whilst leveraging passive feedback from the environment. Buttons, dials, etc. would utilize physical surfaces and take advantage of the affordances of those surfaces, e.g. a dial using a rotating bolt. Given the focus on AR for mechanical instruction, the OCs were predefined using knowledge of the environment the user would be in (e.g. located in front of a certain model of aircraft engine). As a future direction, Henderson and Feiner (2010) identified the capability for a user to locate an object, select a widget type and specify the mapping for that object. This work directly addresses that void.

2.3 Interaction Toolkits

Whilst frameworks exist to abstract TUIs and facilitate easier access, they are primarily for the developer. WorldKit (Xiao et al., 2013) provided developers with a software framework that uses a projector/depth camera pair to enable pervasive interaction. By abstracting the sensing and projection system to provide the developer with simple events, the developer can easily create applications that respond to real world manipulation, such as touch input and object presence. Despite enabling pervasive interaction in the environment, the system cannot be customized given the controls and their functions are defined by the original developer of the application. Our work addresses this void.

The Papier-Mâché (Klemmer et al., 2004) project enabled the fusion of different sensor inputs, allowing the developer to focus on events, rather than hardware sensors. This is along a similar line to the Proximity Toolkit (Marquardt et al., 2011) in providing a set of abstracted proxemic events both within and between objects. Kjeldsen et al. (2003) abstracted visual input, but allowed the application to ask middleware for a given input (e.g. a button), and have that control be dynamically created given current context.

Hardy and Alexander (2012) provided a toolkit for developing interactive projected displays. Focusing on developers, it abstracts the projectors and sensing hardware to provide information about touch-based interaction. This allowed the developer to focus on the application content and interactivity rather than managing display surfaces and their relation to sensed input. Our work focuses uses a similar approach to enable UI creation by end users, rather than developers.

2.4 Summary

Despite work looking at reconfigurable touch and tangible interfaces, previous attempts have stopped short of enabling completely ad-hoc interaction for arbitrary TUIs, instead focusing on touch interaction, primarily with some offline component for the mapping of them to a function to control. Following on from WorldKit and the future work identified by Henderson and Feiner in OCs, this work seeks to enable end users to define tangible and projected controls for existing functionality, whilst also integrating existing systems as a form of input.

3 Using Our EUI's

Our implementation uses a projector and depth camera (Kinect) along with an Optitrack 6DOF tracking system, used to identify objects between frames. Using a combination of the Kinect and Optitrack retro-reflective marker trackers, we can detect touches, objects and contours (using the Kinect) as well the position, orientation and visibility of objects (using the Optitrack). In the future we envision that RGBD cameras combined with computer vision algorithms will replace the need for the 6DOF sensing technology currently used. The system runs ~56fps during use. The Kinect faces down onto a tabletop where all controls are initially authored.

To illustrate how to use the system, we shall use an example of navigating a slideshow using whiteboard marker. Under normal usage, the selection of the function/system to control would be based on the user's current context.

As a means to "boot strap" our system, we use a Griffin PowerMate (supporting a button, rotation sensor and blue LED) as the initial means of input, however we do not utilize the rotation function (the justification for the button is provided in Section 4). To provide feedback to the user regarding when the system expects input via the button or touch input, the button's LED glows (1Hz) when the button can be used. In the future, we will investigate other modes of engaging the system that do not require an external input device.

To create a new input control, the user first presses the button. The different functions available for control (defined and grouped hierarchically in an XML file, discussed later) are then displayed as buttons. The user then touches the function they wish to create a control for. In this example, the user would select the 'PowerPoint' group and select the 'Next Slide' function.

It is at this point the system requires an understanding of what application function the user requires a new control for. Depending on how many parameters the function requires (also defined in the XML), different input controls can be used. For example, setting the volume would require a parameter from a valuator. In our slideshow example, the "Next Slide" function does not require any parameters. The different options available for controlling

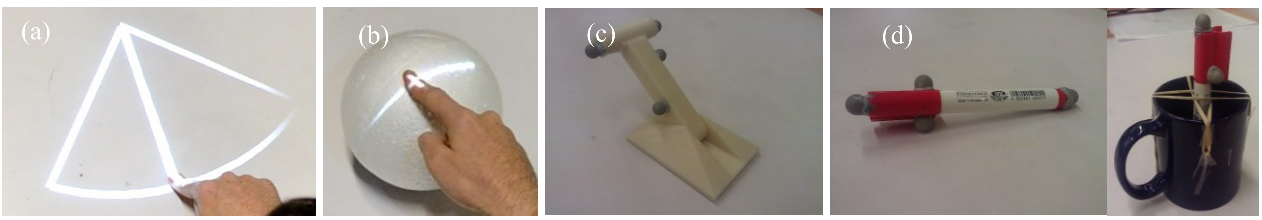


Figure 1: Interacting with dial control (a), slider control on an object (b), interactive lever (c) and improvised joystick (d)

that function are presented to the user as projected touch buttons. Upon selecting one, the user is guided through creating that control. For our "Next Slide" function, we select "Object Orientation" and place the marker we want to use on the table and then press the button to confirm the object selection. The system then prompts the user to orientate the object and the press the button. We hold the marker and point to the right side of the room and press the button. This links this orientation of the marker to the "Next Slide" function, allowing us to point the marker to the right to navigate to the next slide. This means we can now walk around the room, taking the user control (marker) with us, navigating the slides as required, something not possible with previous touch based systems. The same process can then be repeated for going to the previous slide. The whole process of creating a new control takes only seconds and single object can be used in multiple interactions simultaneously, e.g. a marker used as a joystick (Figure 1d) to define both an X and Y value.

When creating projected controls, the process involves the user using their finger to define that control on a surface. For example, to define a dial control the user touches the center of its location and drags out the radius and then continues dragging to define the size of the dial's arc/circle. Projected controls can utilize any physical objects as part of the interaction (e.g. a lever's handle can trigger a virtual button).

To edit controls already created, the user holds the button for more than one second. Projected controls then begin to wobble in a similar fashion to the press-hold-wobble interaction on mobile devices. Users can then touch and drag controls around the table, or drag them off the bottom of interaction area to remove them. To edit object-based interactions, the involved object is placed in the middle of the table, at which time the system presents buttons for each interaction involving that object. These buttons can then be dragged off the table to delete the interaction associated with that object.

3.1 Example Applications

We have created a number of example mappings to control different applications across a number of domains to demonstrate the system's functionality.

Video Editor: The user views the video on an external screen, with the system creating controls for the timeline and cutting/joining sections of the film. The most basic controller would be a slider with (at least) two buttons for cutting/saving the film (Figure 2), but could be more elaborate using a guillotine prop to 'cut' the film and another to join it. Different video clips to be split can be associated with different objects, allowing the user to rapidly switch between clipping/joining different files.

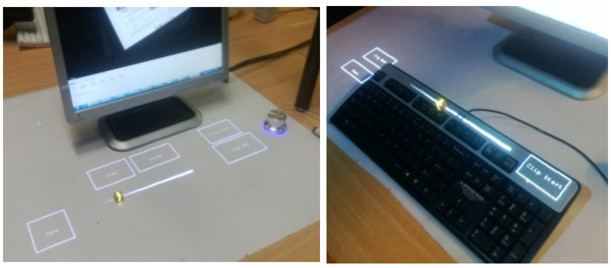


Figure 2: Ad-hoc video editing controls on their own (left) and supplementing the existing controls (right)

Audio Mixer: Allows the user to load and control media whilst adjusting individual audio channels and settings, creating an on demand, customizable DJ-style mixing board. What is novel is the user can create as many controls as required for the particular task, and destroy them when not required. Because the system is not limited to vertical or horizontal controls, the channels could be linked to dials, sliders and levers, etc., located at different positions and orientations surrounding the user, instead of having controls laid out in a linear fashion. Tangible objects provide persistence, visual feedback, and tactile feedback. The use of an application supporting MIDI mappings would enable integration with thousands of existing applications outside of just PC audio applications.

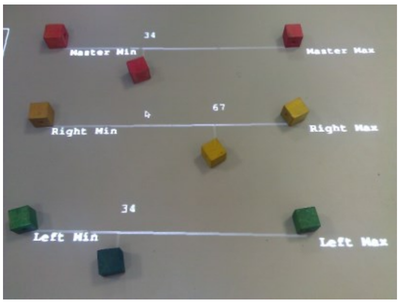


Figure 3: Basic multichannel ad-hoc audio control board

Game Controller: Allows customizable game controllers to be created. Given the ability to use passive objects as active input, users can use a child's pretend steering wheel as an ad-hoc means of controlling racing applications, such as with the AR simulation by Oda et al. (2007).

Since controls can quickly be created with arbitrary materials and turned into functional interfaces, another application is for developing user controls without having to integrate electronics with each iteration, e.g. using 3D printers. Figure 1c depicts a 3D printed throttle-style lever found in airplanes that could work as a functional input device for a flight simulator, without requiring modifications to the game. Despite this application not directly being ad-hoc, we can still leverage near-by materials to quickly create such controls in an ad-hoc manner, where inputs are dynamically created on the fly from passive materials.

We envision applications to developing large scale user controls. By utilizing a realistic simulator using the required mappings, we can design industrial control rooms whilst controlling a working simulation with passive input controls. This allows the user to experiment with configurations for different scenarios (e.g. day-to-day versus an emergency) inside the simulator, creating controls as needed.

4 Preliminary Study

A preliminary study was conducted in the initial design stages to evaluate how users would ideally create controls to interact with existing systems. This study was similar to that used by Henderson and Feiner (2010) for OCs. Participants were given a number of everyday arbitrary objects (blocks, pens, smart phones, scissors, etc.) and asked to create UIs to control different tasks (selection, text entry, path definition, etc.) across different applications (both within and outside the users reach) using three types of UIs: touch, passive tangible and active tangible. Participants were surrounded by writable surfaces (whiteboard and paper covered surfaces) and asked to create controls for the tasks using the materials available. They were told to assume the system was 'all seeing' and asked to sketch out their ideas, experiencing a Wizard of Style evaluation. They were asked to describe the order they expected to be able to performed certain interactions, what navigation aids should be present and when/how to edit existing inputs, etc. Devices, menus and other content described was created using available materials. By evaluating the different types of input devices the users constructed from the available materials, as well as the manner and order in which they constructed them, we evaluated the types of ad-hoc controls the system should support, as well as how they expected to be able to create them within the system. Approximately half of the participants had a computer-science background.

When asked about the procedure for creating controls, participants responded that the system should enable the user to select the function to control first, followed by selecting the input device and then how that device is mapped to the function. It was mentioned that the main thing they were thinking about was what function to control, and thus needed to "offload" that information into the system as soon as possible. This supports the workflow suggested at the conclusion of the work on OCs (Henderson and Feiner, 2010). When asked how the user should be able to select the function to control from a large set, participants said that functions should be able to be grouped, with the user first selecting the function group, then selecting the function itself.

For the primary means of navigating the system, most participants wanted a different form of interaction than that supported by the system, i.e. use of a physical button instead of a touch-based button if interacting by touch. This was described as helping separate defining controls versus navigating the system. The workflow of the final system was followed these results.

Participants used both traditional touch controls (buttons, dials, etc.) and proxemic relationships (between and within objects). Occasional hybrids were created where a tangible object would interact with a touch-based control, triggering the input, in addition to the tangible object's own explicit input, e.g. a lever handle touching a virtual button.

The different types of touch-based controls and interactions using objects served as the first types of interactions that were implemented in the system. The study also served as inspiration for how the user should be able to navigate the system and the information flow between user and system for creating interactions.

5 Supported User Controls

By sensing different types of user actions with physical objects and extending touch interaction to use arbitrary surfaces, we can create a functionally comprehensive set of UI controls to enable the user to both control existing computer applications and create new input devices in their

		Physical						Touch
		Position				Orientation		
		Object Between Positions	Object in Position	Object is Visible	Object Proximity	Object in Orientation	Object Between Orientations	
		Continuous	*Boolean*	*Boolean*	*Boolean and Continuous*	*Boolean*	*Continuous*	*Boolean and Continuous*
Controls	**Button**		☑	☑	☑	☑		☑
	Radio Button	☑	☑		☑	☑	☑	☑
	Slider	☑			☑		☑	☑
	List Box	☑	☑			☑	☑	☑
	Spinner	☑	☑		☑	☑	☑	☑
	Menu		☑	☑		☑	☑	☑
	Tab		☑	☑		☑	☑	☑
Input Device	**Mouse**	☑					☑	☑
	Keyboard							☑
	Joystick	☑					☑	☑
	Steering Wheel	☑					☑	☑

Table 1: Mapping existing controls GUI/physical against how they can be controlled in the system (☑ Supported)

own right. Our initial demonstration of the concept employs the position, orientation, visibility or proximity of one or more physical objects and emulate touch interaction on objects and surfaces using a depth camera. By monitoring these types of interactions, we can support all the types of controls and interactions described in the preliminary study. Table 1 describes how our initial set of sensed user actions can be used to emulate and recreate a wide range of traditional system inputs. The column headings describe the different capabilities for input detection in our implementation, supporting both Boolean and continuous values. For a nullary function (one with no parameters, such as a button press) we sense one of the following properties of the object in relation to the sensed working space: visibility, absolute position (3D), and absolute orientation (all three angles), two objects within a set proximity as well as supporting touch-based buttons. For a single continuous value, the following relative geometric relationships are available: the position of an object relative to two 3D points, the orientation of an object relative to two defined start and end angles, distance between two objects as well as the value of a touch-based slider or dial. Functions requiring two parameters can utilize a projected touchpad.

To enable extensibility, external applications can provide input to the system as if it were native input. These external applications either call an application "SendInput.exe" or connect to the system via TCP socket to send data. The external system sends a keyword to uniquely identify the input as well as any parameters for that input, e.g. the value from a joystick as "joystick 23 60" for the X and Y values. In the case of calling SendInput.exe, the keyword and values are just passed as arguments when the application is executed. This simple approach allows external systems to integrate other capabilities not currently supported, such as gesture, voice, pressure, light, etc. with minimal code.

In Table 1, the checked boxes indicate particular GUI controls that are currently supported and implemented in either touch or tangible form. While we could conceive controls for every position in the table, the checkboxes represent the "sensible" interactions. Using data from 6DOF tracking and depth camera systems, we can map the input sensed as controls to both control existing applications and to emulate/create physical input devices.

5.1 GUI Control Substitution

As discussed previously, a user can quickly create new controls in only three-to-four steps to compliment/replace existing functionality currently controlled by different GUI elements. The top half of Table 1 includes lists how the input of various traditional GUI controls can be created using geometric relationships both within and between objects as well as virtual controls (buttons, sliders, etc.) projected by the system. We have developed a wide range of controls available to users as disposable UI elements. We elaborate on a subset of the developed EUI's based on exiting GUI controls here:

(Toggle)Button: To simulate a button, we can use any tracked object and create an interaction such that when the object is in a certain location (*Object in Position*), the associated button is 'pushed'. For toggle buttons, the persistence of the physical object's presence naturally supports the button's current state. To activate a push button with a physical object multiple times, the object has to be sensed, removed, and sensed a second time. Similarly, the orientation of the object may be used to indicate the state of a button, such as for the PowerPoint "Next Slide" example discussed previously (*Object in Orientation*).

Slider: The user can employ a tracked object(s) and define two positions as start and end positions. Depending on its current position between them (*Object Between Positions*), a value is passed to emulate a slider with that value. This slider is visualized as a projected path or a linear slider with two physical objects as end points (Figure 4). The projected path may be linear, a high order curved path, or an arc. The path can be tracked on 2D or 3D surfaces.

Figure 4: Objects used to as a slider control for volume

Radio Button: Given a set of options, we can use an object's orientation in a single axis (*Object Between Orientations*) to select an option. For each different orientation, a different radio button is selected (Figure 5). Since the object can only have one orientation at any time, only one option can ever be selected. This approach could also be used to emulate a dial. Likewise, the position of a single object may be employed to indicate which radio button is on.

Figure 5: Object used as radio button (digital overlay added for illustration)

List Box: Given a list of items (similar to a menu), the user can use a tracked object, e.g. a pen, and set two different orientations on the table plane, the rotation of the object between those orientations can then be used to interpolate a value and select the appropriate index in the list.

Spinner: A spinner has a small set of discrete values. The user can use the proximity of objects (Figure 6) to set the

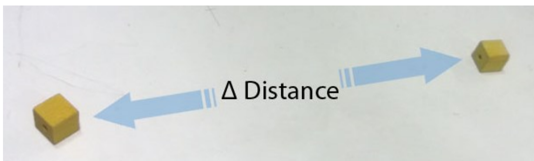


Figure 6: Using physical proximity as valuator (digital overlay added for illustration)

value for the spinner. The distance between two objects may be employed to set the value for an individual spinner value (*Object Proximity*). As the object moves further away, the value increases.

Menu: To emulate a menu system such as a pie menu, we can use the orientation of a fixed object at different rotations to select different items, similar to the radio button functionality. Rotating the object means selecting a different menu item. Given any number of different menu options, we can also utilize projected touch-based buttons.

Tab: To switch to different tabs, the user can associate different tracked blocks with each tab. To switch tabs, the user places one of the tracked objects into the workspace to make it visible (*Object is Visible*). The associated system tab is then selected.

5.2 Input Devices

Using the sensing capabilities of the system, the user can create active user controls from completely passive components. These UIs can supplement or replace existing input devices with user defined ones. We are interested in investigating controlling more complicated interactions instead of just emulating GUI-like elements. The bottom half of Table 1 outlines how some common input devices can easily be created using the system. The system also enables the user to create input controls in place of existing input devices, including the following:

Mouse: As per the Slider example above, the user can define a 2D area using two perpendicular sliders utilizing a common position (an 'L' shape). We can then use the position of an object for each slider to define both the X and Y position of the cursor. The control of the cursor can also be in relative scaled coordinates, similar to a touchpad device.

Keyboard: Using touch on the tabletop surface, a user could emulate a software keyboard.

Joystick: As a tangible example, using a simple whiteboard marker, we can define a joystick (Figure 1d) using the orientation across two different orientation axis (x-min left, x-max right, y-min down, y-max up) and immediately control any number of games. Using a cup and two rubber bands, we can quickly improvise a self-centring joystick capable of controlling applications.

Steering Wheel: We can use any circular object tracked with reflective markers and use the angle between two defined orientations (rotated left and right extremes) as the input value. This provides input akin to the Wii console's steering wheel controller.

Figure 7: Passive steering wheel used as an active input device using tracking markers (visible on top)

6 System Design

To support ad-hoc interaction, we extended the TAM architecture (Walsh et al., 2013). This work focused on programming the logic of tangible interactions, the 'how' of the interaction, whereas we focus on the 'what' of the interaction. One study participant described this as telling the system, "what to do, not how to do it". As such, our work assumes the system already has a set of predefined functions, and instead focuses on how control those functions at run time. In addition, the previous work does not support interacting with external systems and is designed for all interactions and feedback to take place within the system.

The previous architecture physical objects as InteractionObjects (InObjs) with associated Properties (position, color, touch points, etc.). Different Action objects evaluate the Properties according to a given criteria (e.g. rotation around an axis for a rotation input). Using that Boolean result, an Interaction object monitors when the Action occurs, and modifies any number of Properties of different objects as a result. By using the properties of physical objects (location, orientation, etc.) we can leverage them as input for existing systems.

We extend the architecture in four ways: 1) allow Actions to have some form of native representation to indicate their current state, 2) allow Actions to have Properties to communicate a non-boolean state to other components, 3) introduce support for VirtualPropeties as a way to communicate with external systems without incorporating any system-specific code in the core application, and 4) allow external systems to pass information into the system and use that information as an input for the internal patch panel as if it were a normal input from the user.

6.1 Allowing Actions to Have a Representation

Whilst the original architecture was focused on purely tangible interactions, this work has focused on a physical/projected hybrid. Actions that monitor input need to be able to report some kind of state, e.g. buttons not only need to register for a press, but have some representation (i.e. a projected button). Given the Action object evaluates input, it is the only component that is aware of the context of the value (i.e. is the value based on distance, rotation etc.?), it must be responsible for creating any representation for that Action. As such, we assign Action a method to render its state in some form, e.g. in our implementation using OpenGL. This representation is generated based on its current state. For example, a button would render the button display (changing if pressed), or lines to indicate the distance between objects, etc.

6.2 Allowing Actions to Have Properties

Given the purpose of the Action component to monitor the state of an interaction, we add any number of Properties to it to represent its current state and configuration. This value is then read as part of an Interaction, and used to update Properties of other objects.

6.3 VirtualProperties for External Functions

To integrate with existing systems, we require some external communication method. A VirtualProperty (VP) was created to enable communication of a value represented inside the system to an external application. These VPs are associated with single Interaction. A VP takes in a string containing an application name (and required parameters, if any), to run when the Property is set. This string is passed when the VP is created, and could itself be a Property that can be edited at run time. It was thought that by executing an application instead of an API call, we simplify the system by excluding API libraries from the core system, without losing any functionality. This approach also allows the integration of existing applications that can be run/controlled using the command line and allows our system to be used by non-developers.

To format a user control's output as a valid form of input for a specific application, VPs contains a scaling and format setting for how the data should be transformed before the application is executed. This scaling operation includes: minimum, maximum, user defined range, and no modification options. For example, to use the rotation of an object as a 6-value radio dial (Figure 5), you would define a range of 1 (min) and 6 (max). The rotation of an InObj would then be transformed to the range 1-6. Using this, valuators such as sliders or dials can be used to give a value across any given range or even to a Boolean value.

For Actions that provide persistent values, e.g. using the position of an object, VP also contains a rate limiting timer (defined in milliseconds) to restrict the rate of execution for the related application. This defaults to zero for no limit.

All functions that can be controlled are defined (Figure 8) in an XML file, with functions grouped in any arbitrary manner (following the results of the preliminary study). Groups are used to define logical sets of related functions for a given task, meaning they may not all interact with the same application. Rather, they all interact with different applications to manage a common task. As such, functions can appear multiple times across any number of groups. Each XML definition defines a program to be executed when an attribute is set, along with how values pass to that program should be limited/scaled. A user-friendly name is also provided and used within the system to identify that mapping.

```
<output execute="joystick.exe setX %i" ratelimit="0"
scale="range" min="0" max="1080" name="Joystick X" />
```

Figure 8: Example mapping of a single axis of a joystick

6.4 Allow External Systems to Send Data

Our system allows external applications and systems to act as native input controls. For example, our current implementation only senses touch and geometric relationships as forms of input. Using this approach, an application that monitors pressure exerted on digital foam, or the user's voice or gestures can pass this information into the system as a valid control to be utilized by the user.

In a similar approach to enabling external functions, we use a string to integrate external systems for passing information into the system. Input from external systems is supported through a VirtualAction object. These triggers are defined in a similar XML format to outputs, defining a user readable name, keyword and what parameters are provided. This allows external systems to send a string via TCP in the format "*triggerkeyword parameter1value*". In addition to TCP, we wrote a simple application (SendData.exe) that, when executed, directly passes data arguments into the system as if it were sent via TCP.

An associated VirtualAction produces this value as if it were sensed natively, thus registering as a valid input (like a button, etc.). Because this interaction appears as normal input, it can be mapped back to external applications using VP's, only to be read in again. This ability to both read and write to external systems enables the creation of feedback loops (Figure 9), creating support for embodied OUIs.

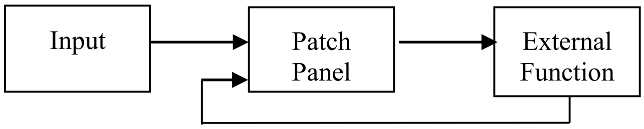


Figure 9: Information feedback loop

This approach simplifies adding new controls to the system, removing the requirement to write system-specific code. For example to integrate Phidgets, you could simply take the example programs and add one line of code to execute SendData.exe and pass in a keyword and the value from the Phidgets. Any new type of input not currently supported by the system (pressure, sound, light, voice, etc.) can easily be incorporated as if it were a native capability of the system. This also enables existing applications that can be controlled via the command line to be controlled via these ad-hoc controls, allowing the user to incorporate new functionality without writing any code, a point of difference when compared to previous homogeneous systems.

7 Future Work

A full evaluation of our implementation remains as future work. Whilst this work enables the ad-hoc creation of tangible controls for existing functions, further work remains in the area of temporal and direct manipulation interactions for virtual content. In addition to this, the object-based interactions do not currently support the full set of proximity based relationships between multiple objects (e.g. an angle between two objects). These could be implemented to ensure full support of proximity based relationships both within and between objects, as used in the Proximity Toolkit (Marquardt et al., 2011).

8 Conclusion

In this paper we have presented our system to support ephemeral interaction using everyday objects and an architecture to support their ad-hoc creation, including the incorporation of new types of input and functionality not supported by the original system. This system supports creating controls to simulate input from existing GUI controls as well as supporting the creation of novel tangible input devices made from passive components. Using design decisions employed from a preliminary study, we have presented an example system and techniques for enabling end users to create a wide range of arbitrary user controls, both tangibly and virtually, to control existing functionality.

9 Acknowledgements

The authors would like to thank the reviewers for their feedback and thoughts and regarding the paper.

10 References

AKAOKA, E., GINN, T. & VERTEGAAL, R. 2010. DisplayObjects: prototyping functional physical interfaces on 3d styrofoam, paper or cardboard models. *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction.* Cambridge, Massachusetts, USA: ACM.

AVRAHAMI, D. & HUDSON, S. E. Forming interactivity: a tool for rapid prototyping of physical interactive products. Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques, 2002 London, England. ACM 141-146.

BALLAGAS, R., SZYBALSKI, A. & FOX, A. 2004. Patch Panel: Enabling Control-Flow Interoperability in Ubicomp Environments. *In:* ANDY, S. & ARMANDO, F. (eds.) *Second IEEE International Conference on Pervasive Computing and Communications.* Orlando, Florida.

DORING, T., SYLVESTER, A. & SCHMIDT, A. 2013. A design space for ephemeral user interfaces. *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction.* Barcelona, Spain: ACM.

FAILS, J. A. & OLSEN, D. 2002. Light widgets: interacting in every-day spaces. *Proceedings of the 7th international conference on Intelligent user interfaces.* San Francisco, California, USA: ACM.

FITZMAURICE, G. W., ISHII, H. & BUXTON, W. A. S. 1995. Bricks: laying the foundations for graspable user interfaces. *Proc. of the SIGCHI conference on Human factors in computing systems.* Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co.

GREENBERG, S. & FITCHETT, C. 2001. Phidgets: easy development of physical interfaces through physical widgets. *Proc. of the 14th annual ACM symposium on User interface software and technology.* Orlando, Florida: ACM.

HARDY, J. & ALEXANDER, J. 2012. Toolkit support for interactive projected displays. *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia.* Ulm, Germany: ACM.

HENDERSON, S. & FEINER, S. 2010. Opportunistic Tangible User Interfaces for Augmented Reality. *Visualization and Computer Graphics, IEEE Transactions on,* 16**,** 4-16.

HENDERSON, S. J. & FEINER, S. 2008. Opportunistic controls: leveraging natural affordances as tangible user interfaces for augmented reality. *Proceedings of the 2008 ACM symposium on Virtual reality software and technology.* Bordeaux, France: ACM.

HOLMAN, D. & VERTEGAAL, R. 2008. Organic user interfaces: designing computers in any way, shape, or form. *Communications of the ACM,* 51**,** 48-55.

HUDSON, S. E. & MANKOFF, J. Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil and masking tape. Proceedings of the 19th annual ACM symposium on User interface software and technology, 2006. ACM, 289-298.

ISHII, H. & ULLMER, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. *Proc. of the SIGCHI conference on Human factors in computing systems.* Atlanta, Georgia, United States: ACM.

KJELDSEN, R., LEVAS, A. & PINHANEZ, C. 2003. Dynamically Reconfigurable Vision-Based User Interfaces. *In:* CROWLEY, J., PIATER, J., VINCZE, M. & PALETTA, L. (eds.) *Computer Vision Systems.* Springer Berlin/Heidelberg.

KLEMMER, S. R., LI, J., LIN, J. & LANDAY, J. A. 2004. Papier-Mache: toolkit support for tangible input. *Proc. of the SIGCHI conference on Human factors in computing systems.* Vienna, Austria: ACM.

MARQUARDT, N., DIAZ-MARINO, R., BORING, S. & GREENBERG, S. 2011. The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. *Proc. of the 24th annual ACM symposium on User interface software and technology.* Santa Barbara, California, USA: ACM.

ODA, O., LISTER, L. J., WHITE, S. & FEINER, S. 2007. Developing an augmented reality racing game. *Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment.* Cancun, Mexico: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

SHARLIN, E., WATSON, B., KITAMURA, Y., KISHINO, F. & ITOH, Y. 2004. On tangible user interfaces, humans and spatiality. *Personal and Ubiquitous Computing,* 8**,** 338-346.

ULLMER, B. A. 2002. *Tangible interfaces for manipulating aggregates of digital information.* Doctor of Philosophy, Massachusetts Institute of Technology.

VILLAR, N., BLOCK, F., MOLYNEAUX, D. & GELLERSEN, H. 2006. VoodooIO. *Proc. of ACM SIGGRAPH 2006 Emerging technologies* Boston, Massachusetts: ACM.

WALSH, J. A., ITZSTEIN, S. V. & THOMAS, B. H. 2013. Tangible Agile Mapping: Ad-hoc Tangible User Interaction Definition. *In:* SMITH, R. & WUENSCHE, B. (eds.) *Australasian User Interface Conference.* Adelaide, Australia: CRPIT.

XIAO, R., HARRISON, C. & HUDSON, S. E. 2013. WorldKit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. *Proceedings of the 2013 ACM annual conference on Human factors in computing systems.* Paris, France: ACM.

Spatial Augmented Reality User Interface Techniques for Room Size Modelling Tasks

Michael R. Marner[1] **Bruce H. Thomas**[1,2]

[1] Wearable Computer Lab, Advanced Computing Research Centre
University of South Australia,
Mawson Lakes Bvd, Mawson Lakes, South Australia 5095,
Email: michael.marner@unisa.edu.au

[2] Email: bruce.thomas@unisa.edu.au

Abstract

This paper present results of our investigations into using spatial augmented reality to improve kitchen design and other interior architecture tasks. We have developed user interface techniques for room sized modelling tasks, including cabinet layout, viewing and modifying preset designs, and modifying materials and surface finishes. These techniques are based on Physical-Virtual Tools, which consist of physical input devices augmented with projected information. These tools and techniques address key user interface issues for spatial augmented reality systems, and we discuss how they can be generalised for other applications. The techniques have been developed in the context of a demonstration application, *BuildMyKitchen*. BuildMyKitchen allows architects to design kitchen cabinets and layouts, and work with clients on the design, in an interactive spatial augmented reality environment.

Keywords: Spatial Augmented Reality, User Interfaces, Architecture.

Figure 1: Resizing the Kickboard Inset using the Resizing Wand and one half of the Resizing Tool.

1 Introduction

We have been investigating the use of Spatial Augmented Reality (SAR) (Raskar, Welch, and Fuchs, 1998) to aid in the design process (Thomas et al., 2011). This paper presents a set of SAR tools to aid in interior architecture work such as kitchen design. An example SAR design application, *BuildMyKitchen*[1] has been developed in consultation with architect Steve Kelly from the University of South Australia's School of Architecture and Design. In developing BuildMyKitchen, new user interface tools and techniques have been created. BuildMyKitchen's user interface is comprised of Physical-Virtual Tools (PVT) (Marner and Thomas, 2010) that support the designer in both the design process and designer/client meetings. The tools that comprise the user interface of BuildMyKitchen are shown in Figure 4. The tasks supported by the tools are: *Cabinet Layout*, *Design Presets*, and *Modifying Finishes*. While these techniques have been developed in the context of kitchen design, they can easily be generalised for use in other application domains and address important interaction issues for SAR systems, such as interaction techniques for room size environments and the lack of virtual projection surfaces.

This paper makes the following contributions to SAR research:

- Cabinet Layout. Physical-virtual tools that allow designers to specify dimensions and distances in a room size SAR environment. These techniques have been optimised for a range of physical movements from floor level to overhead structures, as most Virtual Environment (VE) interaction techniques are designed for the user to be in a single posture, standing or sitting.
- Design Presets. PVT user interface techniques based on a "colour swatch" metaphor for saving and loading preset designs in a SAR environment. These have been developed to allow both the designer and client to quickly access previous developed designs without the need for external technology.
- Modifying finishes using a "magic touch" metaphor based user interface technique for modifying attributes of the final design.
- BuildMyKitchen. This application represents an example exploration of how SAR can be used as part of the design process for interior architecture projects, and serves as a concept demonstrator encapsulating the ideas of the new PVT's.

This paper is structured as follows. Firstly, the motivation for exploring the use of SAR for interior architecture is described. The remainder of the paper describes the functionality of BuildMyKitchen, and shows how the system improves the design process for both the architect and client. This application provides the motivation for

[1] An earlier version of this work (Marner and Thomas, 2013) used the name "PimpMyKitchen", a name inspired by the television programme "Pimp My Ride", which involves customising and improving cars.

the specialised PVT's presented in this paper. The paper describes the new user interface tools and techniques developed to support room size modelling tasks, and how they can be generalised for use in other application areas.

2 Motivation

The motivation for investigating the use of SAR in interior architecture stems from the difficulty and complexity of this design process. SAR uses full scale white replicas (walls, cabinets, and large appliances) to provide a flexible and understandable environment to convey large scale designs. In the case of a kitchen design process, clients currently go through several steps with the designer. Firstly, the client will meet with the architect to discuss their requirements. Different lifestyles are suited to different kitchen designs, and the architect takes these requirements into consideration during the design process. Following this discussion, measurements of the available space are taken, and an initial design is created. This design is presented to the client through 2D plans and 3D rendered images on a computer. Based on these images, the client and architect discuss changes to the design. An additional task for the client is to decide on material finishes, appliances, door handles, etc. To aid in this process, kitchen builders have large showrooms with several finished kitchens. Even still, the client usually will not be able to truly visualise their new kitchen until the final installation in their home.

Figure 2: A design flaw in a completed kitchen, where the dishwasher door and pantry door collide. Damage can bee seen on the pantry door.

A key motivating goal of BuildMyKitchen is to enhance the understanding and decision making ability of the client. While architects are accustomed to visualising the end result from plans and CAD drawings, the client does not usually possess these skills. Even with high quality 3D renders, it can be difficult for the client to envision the scale of the final product. BuildMyKitchen makes it possible to preview designs at 1:1 scale early in the design process. Furthermore, using physical blanks for the kitchen cabinets provides a more intuitive environment than what is possible with a standard computer. This helps not only in deciding on the visual design of the kitchen, but also the functionality of the layout. For example, Figure 2 shows a design flaw in a completed kitchen. Here, the dishwasher door is obstructed by the door to the pantry. To access the dishwasher, the pantry door must be completely closed. Even a slight opening is enough to knock the dishwasher, resulting in damage to both the appliance and the door. These kinds of problems are difficult to predict when looking at a 3D render. Previewing the kitchen at 1:1 scale, with physical mock-ups can help in detecting these kinds of problems. Using SAR allows the projected content of the physical mock-ups to be changed quickly, making iterating the design easier. This ability becomes even more important given the rise in popularity of self-install flat-pack kitchens, where there is no trained architect and limited opportunities to preview the design before construction commences.

Another motivating goal is to provide designers and architects with tools to manipulate and visualise designs on a 1:1 scale. While these professionals are quite capable of visualising designs from drawings in full scale, SAR provides a new medium to explore designs. Discussions with professional architects (Thomas et al., 2011) indicate SAR provides an extra dimension to the design process. To facilitate the manipulation of designs in-situ of the SAR environment and not force the designer to make all their changes in a desktop CAD system, new tools and techniques are required beyond the current state of the art for SAR environments.

3 Background

We are interested in creating 1:1 previews of designs early in the design process. Hare et al. (2009) have demonstrated the importance of physical prototypes to industrial design. DisplayObjects (Akaoka and Vertegaal, 2009) bring spatial augmented reality to product design by projecting user interface controls and other elements onto simple physical mockups. Porter et al. (2010) investigated using SAR to provide interactive controls for control panels early in the design process, rather than using electronic physical controls. (Schwerdtfeger et al., 2008) have demonstrated how laser projectors can be used for quality assurance in industrial settings. WARP (Verlinden et al., 2003) shows how SAR can be used to preview materials on scale models of design mockups. SAR has also been used for maintenance and instruction. CADCast (Piper and Ishii, 2001) shows how SAR can be used to project in-situ instructions for assembly. Suganuma et al. (2008) demonstrate how SAR can be used to instruct a player in how to play Billiards, utilising an overhead camera to detect locations of billiard balls. SAR has also been used to directly project information onto patients to help doctors during surgery (Byung-Kuk Seo et al., 2007).

Our work builds on this foundation, bringing SAR technology to kitchen design and architecture. Our work is inspired by Shader Lamps (Raskar, Welch, Low, et al., 2001); we use white cabinet 'blanks', and use projectors to create the detailed appearance of the design. (Low et al., 2001) used SAR technology to build life sized dioramas for visualizing room size entities with a mixture of physical walls and virtual projected imagery. Bandyopadhyay, Raskar, and Fuchs (2001) demonstrate how SAR can be used to modify the appearance of movable objects using a tracked stylus and 'paintbrush' metaphor.

Physical-Virtual Tools (PVT) (Marner and Thomas, 2010) support interactions within a large-scale SAR environment. PVT are designed to encompass the entire user interface of the SAR applications. The user interface is based around physical tools themselves, in a similar way to Tangible User Interface systems (Ullmer and Ishii, 1997). The operation modes supported by a tool are defined by the shape of the tool itself; picking up a pencil like object will perform pencil like operations (Fitzmaurice, Ishii, and Buxton, 1995). The use of SAR allows for an understandable overloading of the tool's operation. The active mode is conveyed to the user through visual feedback projected directly onto the tool itself. No user interface controls are projected onto the artifact, walls, floor, or ceiling. The user can view and interact with the design artifact from dramatically different viewpoints, such as from different locations within a kitchen.

4 BuildMyKitchen

As previously mentioned, BuildMyKitchen was developed to support the early design process of kitchens and designer/client meetings. To support an architect when developing the initial kitchen design, *blanks* (white simple shaped light weight 3D projection substrates) representing the cabinets are placed in an environment of the shape and size of the target kitchen. One blank is shown in Figure 3. The architect decides on the basic layout of the kitchen by moving the cabinet blanks into the desired locations. The prototype workflow involves the following tasks. Architects first block out key positioning decisions, such as doors and major appliances, to optimise the efficiency of the kitchen, and then work on the next level of detail. Once the blanks are in position, the architect can focus on the layout of the cupboards, drawers, and other components. Some of these components will need to be resizable, depending on the vision of the architect; some, such as appliances, have set dimensions.

Figure 3: An example Cabinet Blank.

The second intended use case for BuildMyKitchen is during meetings where both the architect and client are present. The goal of these meetings is to come to an agreement on the layout of the kitchen, and for decisions regarding material finishes and component selection to be made. There are an overwhelming number of possible combinations of appliances, materials, handles, etc. Rather than have the client select from such a large number of choices, BuildMyKitchen allows the client to select from and customise a set of "preset" designs. These presets are created beforehand by the architect using their experience and design sense. The client chooses a preset on which to base their design and is then able to customise it using a smaller set of options. This set of options is also put together before hand by the architect.

5 Cabinet Layout

A key task for the architect is to decide on the layout of components placed into the cabinets, such as cupboards, drawers, and appliances. As previously mentioned, some of these components, such as appliances, have fixed dimensions that need to be accommodated. Others will be set by the architect based on their design expertise. Some components do not require a specific size, and the component can simply make use of whatever space is available. For example, evenly sizing the height of a set of drawers to make use of the entire height of the cabinet.

Figure 4: The tools that comprise the BuildMyKitchen user interface. Top-left: Resizing Wand, top-right: SAR Swatches, bottom-right: Two-handed Resizing Tool, bottom-left: Stylus.

In addition to components in the cabinets, other design features also need to have dimensions set by the architect. These include:

- Bench thickness. Different benchtop materials are manufactured to different thicknesses. The architect may also choose a specific thickness for the benchtop.
- Kickboard height. The kickboard is the inset material at the base of the cabinet. Choosing the height of the kickboard requires a tradeoff between a functional kickboard and maximum cupboard space.
- Kickboard inset. The inset is the amount the kickboard is set in from the face of the cabinet. Commercial kitchens often have deep insets so workers do not kick cabinets with their feet.
- Bench overhang. The overhang is how much the benchtop overhangs the rest of the cabinet.
- Component spacing. The component spacing is the horizontal and vertical space between components, such as adjacent cupboard doors. Older kitchen designs often have large spaces between components, resulting in smaller doors, where as modern designs tend to maximise the size of the doors, making it easier to see into the cupboards when the door is open.

To accommodate these requirements, BuildMyKitchen automatically places and sizes components according to their properties. This is done using the following basic component types:

- Containers. A container contains zero or more other components, and is responsible for arranging them either vertically or horizontally according to the space available.
- Doors and drawers. A door is placed into the cabinet with the handle either to the left or right. Drawers have the handle placed horizontally, centred at the top of the component.
- Appliances, such as dishwashers and ovens.
- Blank spaces, used for filling an area in the layout.

Drawers, doors, and spaces are able to have preferred dimension set. When layout out a cabinet, containers will try to give components their preferred size. Appliances have dimensions that cannot be altered by the architect, so they are given priority over other components. As containers can be nested, any layout can be produced using the right combination of containers and components. Producing the final layout is a recursive process, with each container allocating space to its children. The algorithm shown in Listing 1 produces a horizontal layout. Vertical layouts are produced in the same way, adjusting component heights instead of widths.

```
void Container::resize(float w, float h, float padding
    , Vector topLeft) {
  int elasticCount = 0;
  float availableSize = w;
  foreach (PanelComponent* component, mContents) {
    if (component->hasPreferredWidth())
      availableSize -= component->getPreferredWidth();
    else
      elasticCount++;
  }
  availableSize -= (mContents.size()-1) * padding;
  wcl::Vector currentTopLeft = topLeft;
  foreach (PanelComponent* component, mContents) {
    if (component->hasPreferredWidth()) {
      if (elasticCount > 0) {
        component->resize(component->getPreferredWidth
            (), h, padding, topLeft);
        topLeft.x += component->getPreferredWidth() +
            padding;
      }
      else {
        component->resize(w / mContents.size(), h,
            padding, topLeft);
        topLeft.x += w / mContents.size() + padding;
      }
    }
    else {
      component->resize(availableSize / elasticCount,
          h, padding, topLeft);
      topLeft.x += availableSize / elasticCount +
          padding;
    }
  }
}
```

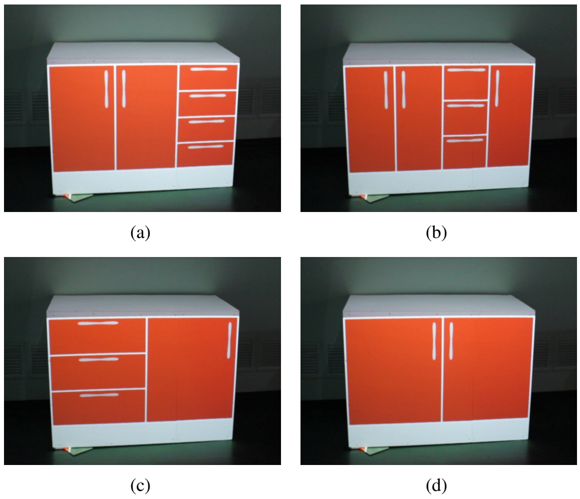
Listing 1: Algorithm for producing a horizontal cabinet layout

5.1 Creating a Cabinet Layout

The architect starts with a cabinet blank with no components added to it. They can then add components to the layout using a simple keyboard interface. As components are added, the system automatically arranges the components to fit the available space. Figure 5 shows four different layouts on the same cabinet. Once the necessary components have been added, the architect can move on to specifying dimensions to the components, as shown in Figure 6. For example, drawers used to store saucepans will need to be wider and deeper than drawers used to store cutlery. At a higher level, the architect can experiment with the overall kitchen layout by moving the cabinet blanks into different positions in the room.

5.2 Resizing Using The Two-Handed Resizing Tool

The two-handed manipulation tool has been developed for performing operations such as resizing components of the kitchen. The tools consists of two identical halves which are held in each hand. To resize a component, the user places each half of the tool against opposite edges of the component to be resized, and adjusts the distance between their hands until the desired size is obtained. This is an intuitive gesture, and is a similar motion to how people estimate and communicate distances in the real world.



(a) (b) (c) (d)

Figure 5: Example cabinet layouts.

This tool is used exclusively for resizing. Resizing in BuildMyKitchen has a single user definable attribute: the snapping mode. The user is able to choose between snapping to 1mm, 5mm, and 10mm intervals when resizing. In addition, when a set of common dimensions are used in kitchen design, the user can choose to snap to the nearest preset dimension. The snapping mode is conveyed to the user by projecting this information onto both halves of the tool. The current dimension is also projected onto the tool, allowing the architect to precisely set dimensions with little effort.

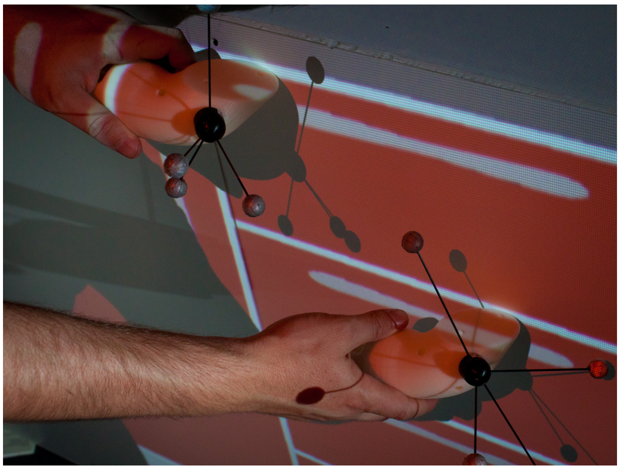
Resizing components on the surface of the blank is accomplished by placing each half of the tool on opposite edges of the component to be resized. The architect resizes the component simply by changing the distance between the two halves. Resizing the kickboard inset and the benchtop overhang is a more difficult problem, because the physical cabinet blank does not match the virtual geometry. Resizing these components is accomplished by placing one half of the tool on the component to be resized, such as the front of the bench overhang, and the other on the top of the cabinet. A line is projected onto the cabinet blank indicating the amount of overhang. The architect changes this dimension by dragging the tool along the top of the cabinet.

5.2.1 Resizing Wand

In addition to the resizing tool, a *Resizing Wand*, as shown in Figure 1, is also provided. This was added after experimenting with resizing kickboard components. As the kickboard is near the floor, using the resizing tool requires kneeling down, which quickly becomes uncomfortable. The resizing wand can be used instead of one half of the resizing tool, allowing resizing kickboard properties from a more comfortable stance. This type of tool could be extended to specifying a range of dimensions outside the user's reach, such as cabinets over benchtops and lighting fixtures on the ceiling, but these techniques require further evaluation against current out of arm reach VE interaction techniques.

6 Design Presets

While the cabinet layout process would be conducted exclusively by the architect, previewing and modifying kitchen designs is a process that involves the client. Rather



(a)

(b)

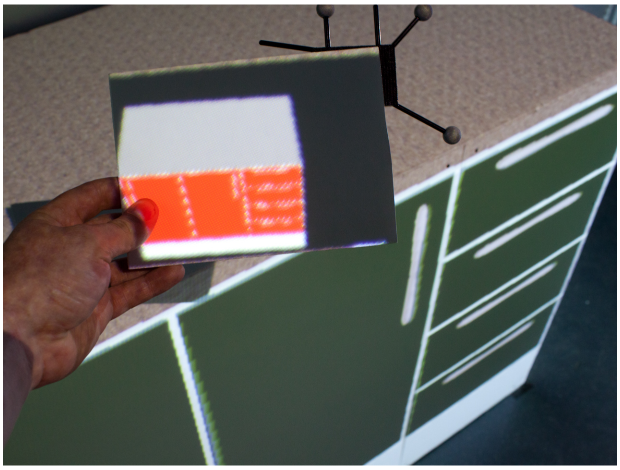
Figure 6: (a) A user resizes the height of a drawer, and (b) the thickness of the bench top.

than giving the client a ‘blank canvas’, they would instead be working with designs created earlier by the designer. These *Design Presets* can be a starting point from which the client can decide on their own design.

6.1 SAR Swatches

To facilitate loading and saving kitchen designs, BuildMyKitchen makes use of *SAR Swatches*. SAR Swatches are based on a metaphor adapted from the colour swatches used to choose paint colours. A SAR Swatch is a small board, and a 3D render of a kitchen design is projected onto the board. Clients can quickly evaluate different designs by holding several in their hands at once. SAR Swatches are also related to Tangible Bits (Ishii, 2008), in that the digital information is represented as a physical object.

The client can preview one of the designs simply by placing a swatch onto the cabinet. Doing this causes the kitchen design shown on the swatch to be loaded onto the cabinet blanks, and the client can experience the design at 1:1 scale on the physical surfaces. The other side of the swatch is used for saving designs. A new design can be saved to a swatch by placing it upside-down on one of the cabinets. The projected visualisation swatch will change, indicating the system is asking for confirmation (Figure 7(b)). To confirm the save operation, the user simply slides the swatch to the right. This motion will save the current kitchen design to the swatch, with the visualisation projected onto the other side of the swatch updating accordingly. The number of swatches is only limited by the number of physical swatches constructed for use; theoretically we could have thousand of swatches. They are uniquely identified by the tracking system.



(a)

(b)

Figure 7: (a) Comparing one design with another using a SAR Swatch, (b) saving a new design to a SAR Swatch by placing the swatch on the cabinet, and sliding right.

SAR Swatches provide a good example of the tradeoff between several physical tools and a single tool. Rather than individual swatches, designs could instead be previewed one at a time on a Personal Interaction Panel (PIP) (Szalavri and Gervautz, 1997) style device. The PIP could then be placed in the load area and the kitchen selected would be loaded. Alternatively, a tool for previewing could be removed entirely, and instead the user would only view designs on the full sized mock-up.

SAR Swatches were chosen because they have several advantages compared to the alternatives described above. Firstly, swatches allow the user to compare different designs simultaneously. This can improve their ability to choose a design. For example, when comparing two designs, the client may decide they prefer one design overall, but with the door handles from another design. This realisation would be difficult to come to if only one design could be seen at a time. Using several physical tools also makes it easier to quickly flick through several different designs in a more natural way than using a menu. Finally, SAR Swatches adapt a device already in use in interior design: the paint colour swatch. Kitchen swatches extend this to storing several aspects of the design on a single swatch.

SAR Swatches are interesting to consider because these tools are the first presented in this dissertation where the primary role is to hold information, not manipulate elements of the design. However, previewing a design on the swatch can be considered as one task, loading a design on the prototype as another, and saving the current design to the swatch as a third task. Each swatch has a single parameter: the kitchen design it is storing.

6.1.1 Generalising Swatches

Swatches solve the problem of loading and saving data in a SAR environment in a way that is specifically suited to the nature of SAR. Text input is difficult in large scale SAR environments because there is no logical location for a keyboard. However, for the kinds of industrial design applications discussed in this dissertation, the data to be saved is often visual in nature. Therefore, there is no need for the user to enter filenames. Instead, they can access the data in a visual manner. This is accomplished through the use of PVT. The physical SAR Swatches are treated like any other object in the SAR environment, and have virtual information projected onto them. The user is able to refer to and access data in a visual manner, and does not have to concern themselves with text entry. This simplifies the user interface by removing the keyboard, which may not have a logical location in a SAR environment.

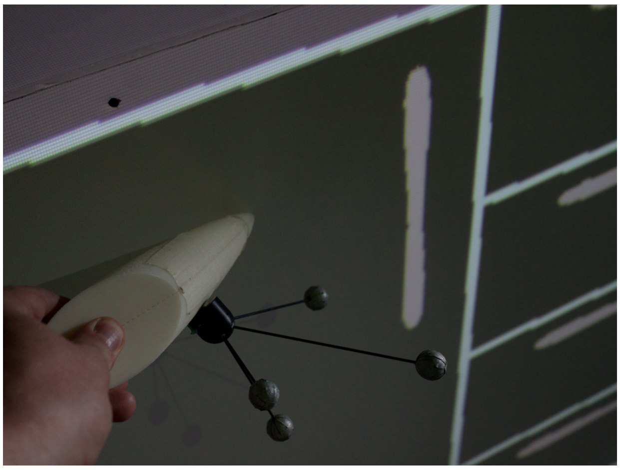
7 Modifying Finishes

The final use for BuildMyKitchen is to allow the client and architect to modify one of the presets in order to arrive at the final design. The client is able to select from several options to customise the following aspects of the design: materials (for surfaces such as benchtops, doors, kickboard, and splashback) and fixture designs (such as handles, power points, and taps).

There are of course infinite possible combinations of features. However, at this stage of the design, the number of choices is deliberately limited by the architect. Just like the kitchen presets, the designer makes available a set of options that will work together, based on their design experience. The client is able to customise their kitchen based on a smaller number of choices, rather than allowing every possible combination.

A *Magic Touch* metaphor is used for the modifying finishes tool, as this greatly simplifies the user interface. To select different options, the user taps the surface of the cabinet blank with the *Stylus*. The system cycles through the choices one at a time. The process is exactly the same for door handles and such. For these small items a selectable region is defined around the projected texture. In the case of a finish being replicated in more than one location as with a handle, the system changes all occurrences to match the selected style. For example there might be different sizes of handles for different sized doors; therefore, the change would modify the style of all the handles but with the correct sized handles for each occurrence. This mode of selection is acceptable because the number of choices is small. In the demonstration system between five and ten options are loaded for each of the changeable components. In addition, this task is fundamentally exploratory in nature. The client is interested in experimenting with different designs, and does not need to load specific finishes at this stage. However, if there were many choices, an alternate system, such as a pie menu displayed with a PIP could be employed.

Reusing the Stylus for BuildMyKitchen demonstrates how the complexity of a physical-virtual tool depends greatly on the application where it is used. Here, the tool is used for a single task: changing a property of the kitchen. This task has no attributes. While the single tool is used to change all the properties of the kitchen design, the active property is not an attribute of the tool. The property being changed simply depends on what the user touches with the Stylus.



(a)

(b)

Figure 8: (a) A user changes the colour of the cabinet doors using the stylus, (b) a finished kitchen design.

8 Implementation

Our demonstration system is running on a standard desktop PC, with Nvidia Quadro FX 3800 graphics cards. The environment is lit using four NEC NP510W projectors, each with a resolution of 1280x800. All projectors were mounted on the ceiling, approximately 3.5 metres above the floor. One projector was centred on the cabinet blank, pointing almost vertically down. This projector primarily illuminated the cabinet top and SAR Swatches. The other three projectors were arranged in a hemisphere around the work area, pointing down at approximately 45°, giving a broad projection volume.

An eight camera Optitrack[2] tracking system is used to track the tools and cabinet blanks. BuildMyKitchen is written in C++, on top of our in house developed SAR software framework, using OpenGL.

9 Conclusions and Future Work

This paper has presented our work into using SAR for kitchen design and interior architecture. We have described our SAR design application, BuildMyKitchen,

[2]http://www.naturalpoint.com/optitrack/

and how it improves the design process for both architects and their clients. BuildMyKitchen features a PVT based user interface. We have discussed PVT-based tools and techniques developed for the application, and how they can be generalized for use in other application domains.

In the future we would like to extend BuildMyKitchen to allow functional analysis of kitchen layouts. For example, showing the available storage space once appliances, plumbing, and other components are in place. We could also optimize the functional layout by highlighting the predicted use of areas of the design, based on data on the clients actual usage patterns. In addition to these analysis tasks, we would also like to improve our graphics system. This would allow accurate previews of lighting configurations, based on models of actual light fittings.

References

Akaoka, E. and R. Vertegaal (2009). “DisplayObjects: prototyping functional physical interfaces on 3D styrofoam, paper or cardboard models”. In: *CHI 2009*. Boston, MA.

Bandyopadhyay, D., R. Raskar, and H. Fuchs (2001). “Dynamic Shader Lamps: Painting on Movable Objects”. In: *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 207–216.

Byung-Kuk Seo, Moon-Hyun Lee, Hanhoon Park, Jong-Il Park, and Young Soo Kim (2007). “Direct-Projected AR Based Interactive User Interface for Medical Surgery”. In: *Artificial Reality and Telexistence, 17th International Conference on*, pp. 105–112.

Fitzmaurice, G. W., H. Ishii, and W. A. S. Buxton (1995). “Bricks: Laying The Foundations for Graspable User interfaces”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., pp. 442–449.

Hare, J., S. Gill, G. Loudon, D. Ramduny-Ellis, and A. Dix (2009). “Physical Fidelity: Exploring the Importance of Physicality on Physical-Digital Conceptual Prototyping”. In: *Human-Computer Interaction INTERACT 2009*, pp. 217–230.

Ishii, H. (2008). “Tangible bits: beyond pixels”. In: *Proceedings of the 2nd international conference on Tangible and embedded interaction*. TEI ’08. New York, NY, USA: ACM, xvxxv.

Low, K.-L., G. Welch, A. Lastra, and H. Fuchs (2001). “Life-sized projector-based dioramas”. In: *VRST ’01: Proceedings of the ACM symposium on Virtual reality software and technology*. inproceedings. New York, NY, USA: ACM, pp. 93–101.

Marner, M. R. and B. H. Thomas (2013). “Spatial Augmented Reality User Interface Techniques for Room Size Modeling Tasks”. In: *Poster Sessions: Proceedings of the IEEE Symposium on 3D User Interfaces*. Orlando, FL, USA.

Marner, M. R. and B. H. Thomas (2010). “Tool Virtualization and Spatial Augmented Reality”. In: *Proceedings of the 20th International Conference on Artificial Reality and Telexistence*. Adelaide, South Australia.

Piper, B. and H. Ishii (2001). *CADcast: a Method for Projecting Spatially Referenced Procedural Instructions*. Tech. rep. MIT Media Lab.

Porter, S. R., M. R. Marner, R. T. Smith, J. E. Zucco, B. H. Thomas, and P. Schumacher (2010). “Spatial Augmented Reality for Interactive Rapid Prototyping”. In: *Proceedings of the 20th International Conference on Artificial Reality and Telexistence*. Adelaide, South Australia.

Raskar, R., G. Welch, and H. Fuchs (1998). “Spatially Augmented Reality”. In: *In First IEEE Workshop on Augmented Reality (IWAR98)*, 1120.

Raskar, R., G. Welch, K.-L. Low, and D. Bandyopadhyay (2001). “Shader Lamps: Animating Real Objects With Image-Based Illumination”. In: *Rendering Techniques 2001: Proceedings of the Eurographics*, pp. 89–102.

Schwerdtfeger, B., D. Pustka, A. Hofhauser, and G. Klinker (2008). “Using laser projectors for augmented reality”. In: *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. Bordeaux, France: ACM, pp. 134–137.

Suganuma, A., Y. Ogata, A. Shimada, D. Arita, and R.-i. Taniguchi (2008). “Billiard instruction system for beginners with a projector-camera system”. In: *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. Yokohama, Japan: ACM, pp. 3–8.

Szalavri, Z. and M. Gervautz (1997). “The personal interaction panel - a Two handed interface for augmented reality”. In: *EUROGRAPHICS 97*. Budapest, Hungary, pp. 335–346.

Thomas, B. H., G. S. Von Itzstein, R. Vernik, S. R. Porter, M. R. Marner, R. T. Smith, M. Broecker, and B. Close (2011). “Spatial Augmented Reality Support for Design of Complex Physical Environments”. In: *Workshop on Interdisciplinary Approaches to Pervasive Computing Design*.

Ullmer, B. and H. Ishii (1997). “The metaDESK: models and prototypes for tangible user interfaces”. In: *Proceedings of the 10th annual ACM symposium on User interface software and technology*. Banff, Alberta, Canada: ACM, pp. 223–232.

Verlinden, J., A. d. Smit, A. Peeters, and M. v. Gelderen (2003). “Development of a flexible augmented prototyping system”. In: *Journal of WSCG*.

Image Warping for Enhancing Consumer Applications of Head-mounted Displays

Edward M. Peek **Burkhard C. Wünsche** **Christof Lutteroth**

Department of Computer Science
The University of Auckland
Email: epee004@aucklanduni.ac.nz

Abstract

Head-mounted displays (HMDs) are highly immersive display devices which are increasingly targeted towards consumer-level video games, E-learning, training and other forms of digital entertainment. Despite the hardware now being available, quality factors — particularly latency — are still issues in large part due to consumer graphics hardware being tailored for throughput instead of latency, and the expectation of a nausea-free experience even on weak hardware. In this paper we discuss the benefits and disadvantages of using image warping as a means to improve frame rate and latency in the context of consumer applications. As part of this, we suggest two appropriate algorithms for performing the image warping. These methods are compatible with other latency reduction strategies such as predictive tracking, and require minimal changes to conventional 3D rendering processes. In addition, they are implemented purely in software and are therefore suitable for use on existing consumer PCs and HMDs. Initial evaluations indicate that artefacts from both warping algorithms are minimally visible for typical environments.

Keywords: head-mounted display, latency reduction, frame rate, image warping

1 Introduction

Today, there are many general consumer applications where people interact with 3D virtual environments. Video games, computer aided design, E-learning, online virtual worlds, 3D mapping and navigation and architectural walkthroughs are just some examples. Additionally, more and more devices are becoming capable of running these applications; what was previously only attainable using high-end desktop computers is now possible on portable laptop computers, tablets and even mobile phones at remarkably detailed visual quality.

What has changed little is the way which we view these virtual environments. It is still almost universally done by showing a single rendered image on a flat panel monitor that takes up a small portion of the user's visual field. This does not provide a very immersive experience, and only recently have more immersive display technologies started to become available at reasonable cost and quality. Stereoscopy is one such technology that has appeared in computer monitors, television sets, handheld gaming consoles and smartphones, however the need to wear special 3D glasses is a factor that has hindered the uptake of the technology.

One of the more immersive classes of display are head-mounted displays (HMDs). These are worn on the user's head and have the benefit of producing stereoscopic 3D, taking up a large portion of the user's visual field and blocking out the real world. Many types also have sensors to track their orientation and/or position, allowing the user to look around the virtual environment with natural head motion. Traditionally HMDs have been expensive, or of low quality, and have therefore been limited to specialised industrial, scientific, military and enthusiast audiences. Recently however, some HMDs that are both low cost and high quality have been designed for general consumer use. A notable example is the Oculus Rift (Oculus VR, 2012) which has gained support from prolific virtual reality (VR) enthusiasts and game developers (Oculus VR, 2013), despite being pre-production hardware.

While adapting 3D applications to use sophisticated VR displays is technically straightforward, certain quality and performance factors such as rendering latency and frame rate become significantly more important for an acceptable user experience. This necessitates tighter control of these factors than previously required in these applications. When coupled with the fact that most consumers would assume that the experience for sophisticated VR displays should be at least as good as that of conventional displays, there arises a need for methods to improve these quality factors beyond what is possible by using conventional methods.

In this paper we discuss a frame rate and latency enhancement technique based on image warping, and how it may be applied to and adapted for these sorts of applications. After exploring related work, we discuss the architecture of our image warping enhancement and the benefits it provides. We then discuss the characteristics of the enhancement with respect to consumer applications.

2 Related Work

Since low latency and high frame rates can be so important for virtual reality systems, much work has been performed over the years in an attempt to enhance them.

A universal way to compensate for any type of latency is to use prediction. In the context of HMDs, by predicting the position of the head at the time the render frame will become visible, instead of the raw sampled position, the average latency of head motions can be reduced. Azuma and Bishop (1994) discuss several ways to predict head position and find it can reduce the magnitude of tracking errors by 2–10 times.

One approach to provide optimal frame rates is to guarantee that the time it takes to render a frame is always less than the deadline for that frame being scanned out to the display. Olano et al. (1995) discuss a method for achieving this for HMDs; using a cluster of graphics processors to distribute computation. The authors also dynamically offset scan lines to further reduce latency by compensating for the time it takes to scan out an image to a CRT display. For current consumer HMD systems this is impractical, as the required cluster of graphics processors is very uncommon, and modern consumer HMDs are LCD or OLED based.

Kijima and Ojika (2002) develop the idea of shifting scan lines by designing a modulator system that is inserted between an LCD panel and the driving circuitry. The modulator has a direct feed to the predicted head position which bypasses additional sources of latency. Since this process requires physical modification of the HMD, it is less suitable for use on existing consumer HMDs, as it is unrealistic to expect average users to be able or willing to perform the modification.

A third class of latency compensation techniques has been researched that, like prediction, is able to be implemented entirely through software. These techniques are based on *image warping*, i.e. taking previously rendered views and modifying them to generate new ones. Mark et al. (1997) discuss how this can be utilised to improve general frame rates, and latency when using a remote display. The authors, however, do not provide an implementation of their system, something that is accomplished in more recent work.

Smit et al. (2007) present and implement an architecture for performing this warping. This is done using dual GPUs, one for rendering the virtual environment to produce what they call *application frames*, and one for rendering warped *display frames* that are then presented to the user. This architecture is tailored for local VR systems, in particular fish-tank virtual reality which combines stereoscopy with head-coupled perspective. In their later work they improve their architecture to reduce crosstalk with stereo shutter glasses (Smit et al., 2008), adapt the system for use on a single GPU (Smit et al., 2009), and explore the quality and performance of different warping algorithms.

In our own previous work (Peek et al., 2013) we have implemented and tested a image warping architecture specifically designed for HMDs. Our user evaluation showed that image warping significantly improves the smoothness of head tracking, and for most users is indistinguishable from ideally fast rendering; at least when head tracking is the only motion in the scene.

3 Image Warping

We extend prior work by examining how the practicality of these techniques is affected by their use in general applications, and with currently available consumer-level hardware. This builds upon our previous work in this area and provides a more detailed consideration of image warping in normal 3D applications. Within this paper, our reference scenario is an interactive video game on desktop class PC hardware using the Oculus Rift developer kit as the system display, and a keyboard and mouse for input. This is likely one of the major use cases for this wave of HMDs, and it is complicated enough to surface the issues we are looking for, while also allowing for some generalisation to other hardware platforms and applications. Considering this problem domain, the following points were considered especially important and had a large influence on the process of our investigation.

3.1 Domain Requirements

The first restriction due to this sort of usage, is that physical modification of computer hardware is unattractive to the user, so any sort of enhancement must be entirely software based. This rules out techniques that involve modification of the HMD hardware itself, such as the *Reflex HMD* proposed by Kijima and Ojika (2002). Conversely, techniques based on image warping via the GPU are practical in this case as they are typically able to be run on any modern desktop GPU.

An extension to this restriction is that is cannot be assumed that the user is willing or able to upgrade their computer hardware, specifically their CPU or GPU, in order to facilitate use of any enhancements. Since a large proportion of typical users' PCs have only a single GPU, techniques that require multiple GPUs (such as the work by Smit et al. (2007)) are impractical unless they are able to be modified to run on a single GPU (as in their later work (Smit et al., 2009)). It may be noted however that while many users have only a single *dedicated GPU* (dGPU), most modern CPUs have an *integrated GPU* (iGPU) that may be used in addition to the more powerful dGPU. However, such iGPUs are significantly slower (up to an order of magnitude), and so sophisticated techniques may not run fast enough to produce any improvement in latency or frame rate.

Furthermore, considering the range of PC configurations in current use, a very large number have only an iGPU as the sole graphics processor. In a July 2013 survey of desktop and laptop PCs used for gaming (Valve Corporation, 2013), the most common graphics processor was an iGPU (Intel HD 3000) and over 14% of systems had an iGPU as their only graphics processor. Such systems pose an ambivalent target, as while they have the most to gain from latency reduction and frame rate improvement, they are also the most difficult ones to develop for due to their limited capacities. The consequence of this is that for consumer applications, any enhancement technique should be capable of running on even modest PC hardware.

The last point of note concerns the structure of the application that is to be enhanced. With HMDs still being only a niche display type, it is unrealistic to expect application developers to enact significant changes to application architecture and rendering pipeline in order to implement an enhancement currently useful to so few users. This is especially true when considering its impact to development and testing costs, or to the quality and performance of conventional displays. The suitable response is to ensure that any proposed enhancement is compatible with popular rendering models (such as forward and deferred shading) with minimal changes, or even entirely separable in the style of NVIDIAs 3D vision (NVIDIA Corporation, 2013) or our own method for head-coupled perspective (Li et al., 2012).

3.2 Overview

This paper discusses how image warping, such as that by Smit et al. (2007), may be implemented taking into account these special requirements. We also tailor our method specifically for HMDs, where previous work (Smit et al., 2007) has targeted fish-tank virtual reality (FTVR) systems. In this section we present a high-level overview of our image warping enhancement, how it would be added to a conventional HMD software application, and two warping algorithms designed for the problem domain.

The central idea behind image warping (as a means to improve frame rate) is that for a rendered frame, subsequent frames are visually very similar and therefore may be extrapolated from the original frame with acceptably small errors. Error-free reference frames must still be regularly generated to prevent errors accumulating over time, and to account for the fact that only certain changes can be extrapolated by warping. To structure this method, we use an architecture similar to that proposed by Smit et al. (2007), in which the simulation and warping run concurrently and at separate rates. The simulation generates simulation frames using conventional rendering as fast as it can. At the same time, the warper takes the most recently produced simulation frame and uses it to generate a display frame at exactly the display's refresh rate: first by warping the simulation frame with up-to-date head orientation, and then performing the HMD lens correction. It is only the display frames that are made visible to the user via the HMD. Additionally, because HMDs are stereoscopic displays, each simulation and display frame contains a left and right eye view which are rendered and warped independently.

This set-up is able to increase frame rate because the generation of display frames is less computationally expensive than simulation frames, and thus able to be performed more frequently on the same PC hardware. However, because image warping shortens the generation of individual display frames, but requires the sequential generation of both simulation and display frames, it reduces some forms of latency while increasing others.

Basic image warping can only be used to extrapolate object and viewpoint motion between frames. Other types of motion require auxiliary data to be generated with each conventionally rendered frame. The required data for each type of motion is as follows.

Viewpoint rotation only requires the rendered image colour

Viewpoint translation requires the colour and depth of each pixel in the rendered image

Object rotation & translation requires the colour, depth and velocity of each pixel in the rendered image

Different algorithms exist for performing the actual image warp. This paper suggests and discusses two appropriate methods for our target domain in Sections 3.4 & 3.5.

3.3 Benefits

To quantify the benefits of image warping, we must compare how it relates to conventional rendering. Here we briefly discuss what variables influence the frame rate and latencies of these two rendering processes. In addition to frame rate, the two types of latency of interest are what we call tracking latency L_t and simulation latency L_s. Tracking latency specifically concerns HMDs in that it is the time delay between a rotation of the user's head and the rendered frame accounting for that rotation being visible to the user. Simulation latency is the time delay between a change in the simulation's state (including response to user input) and that new state being visible to the user. Usage of HMDs is particularly sensitive to tracking latency — an excess of which causes motion sickness — while simulation latency is more acceptable; but only to the extent that while in need not be aggressively reduced, it should also not be needlessly increased. In a GPU-bottlenecked conventional rendering process, as visualised in Figure 2, frame rate and latencies are given by the following equations.

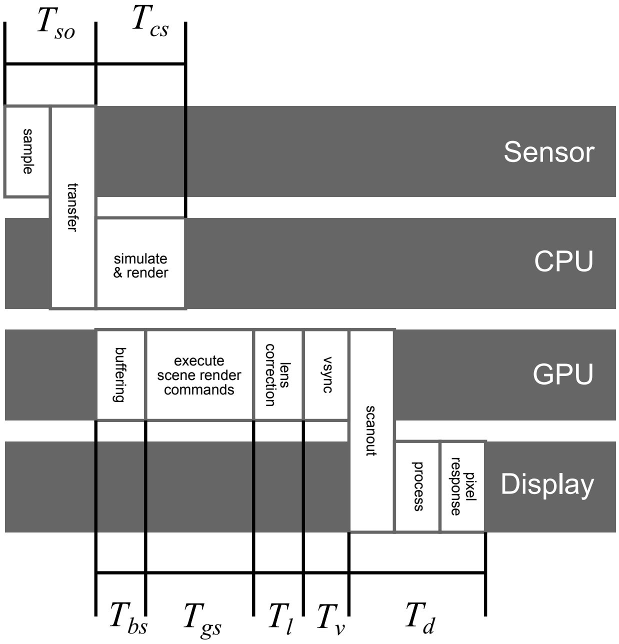


Figure 2: The constituent parts of tracking and simulation latency in the conventional rendering process

$$L_t = T_{so} + T_{bs} + T_{gs} + T_l + T_v + T_d \quad (1)$$

$$L_s = T_{bs} + T_{gs} + T_l + T_v + T_d \quad (2)$$

$$frame\ rate = \frac{1}{T_{gs}} \quad (3)$$

Using this rendering approach, frame rate and latencies are typically improved by reducing scene and shading complexity to indirectly control T_{gs}. Other delays are either impossible to affect from software, or have consequences from being changed.

The rendering pipeline modified for image warping is shown in Figure 3 and results in new equations (4–7) for frame rate and latencies

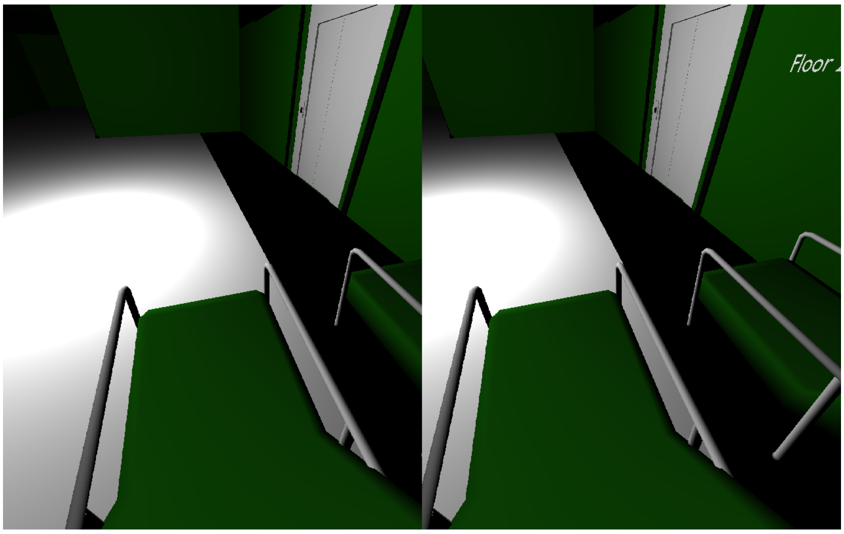
$$L_t = T_{so} + T_{bw} + T_{gw} + T_l + T_v + T_d \quad (4)$$

$$L_s = T_{bs} + T_{gs} + T_p + T_{bw} + T_{gw} + T_l + T_v + T_d \quad (5)$$

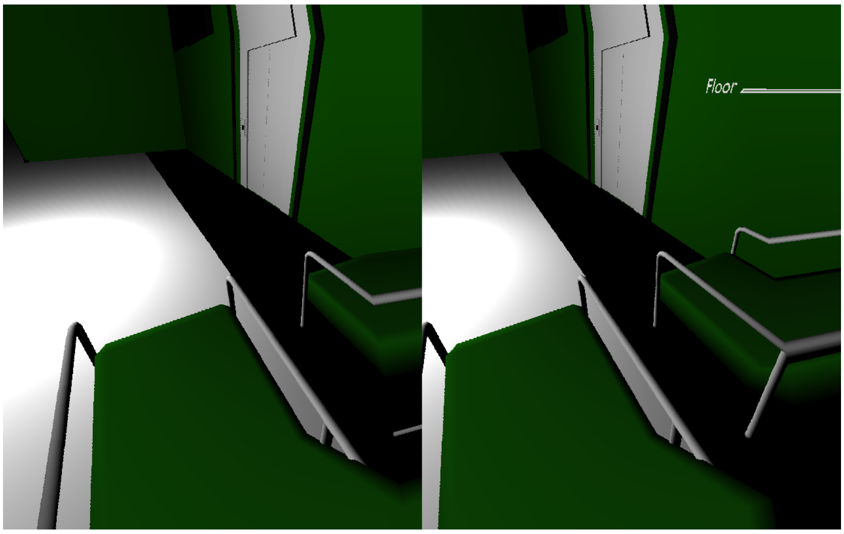
$$frame\ rate_{single\ GPU} = \frac{1 + r}{r \times T_{gw} + T_{gs}} \quad (6)$$

$$frame\ rate_{dual\ GPU} = \frac{1}{T_{gw}} \quad (7)$$

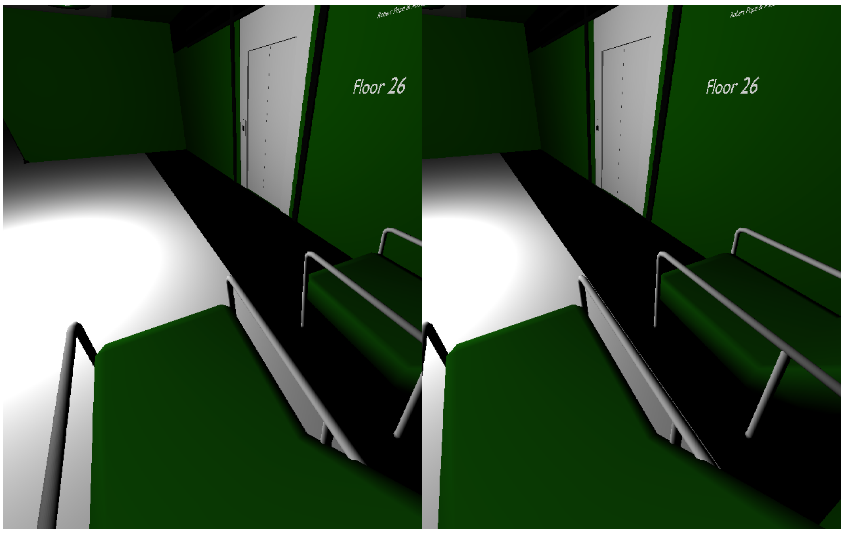
where r is the number of display frames generated per simulation frame.



(a) Simulation frame (t0), rendered before head rotation



(b) Display frame (t1), result of warping the simulation frame after head rotation. Fringe artefacts are evident in the top and right sides of the images, while a minor edge artefact is visible on the right chair arm in the right-eye image



(c) Simulation frame (t2), rendered after no additional head rotation. It corrects the errors introduced in the warped image

Figure 1: Example frame generation sequence, with an (unrealistically fast) head rotation to the upper-right. The rotation occurs between the rendering of the first simulation frame and the first display frame

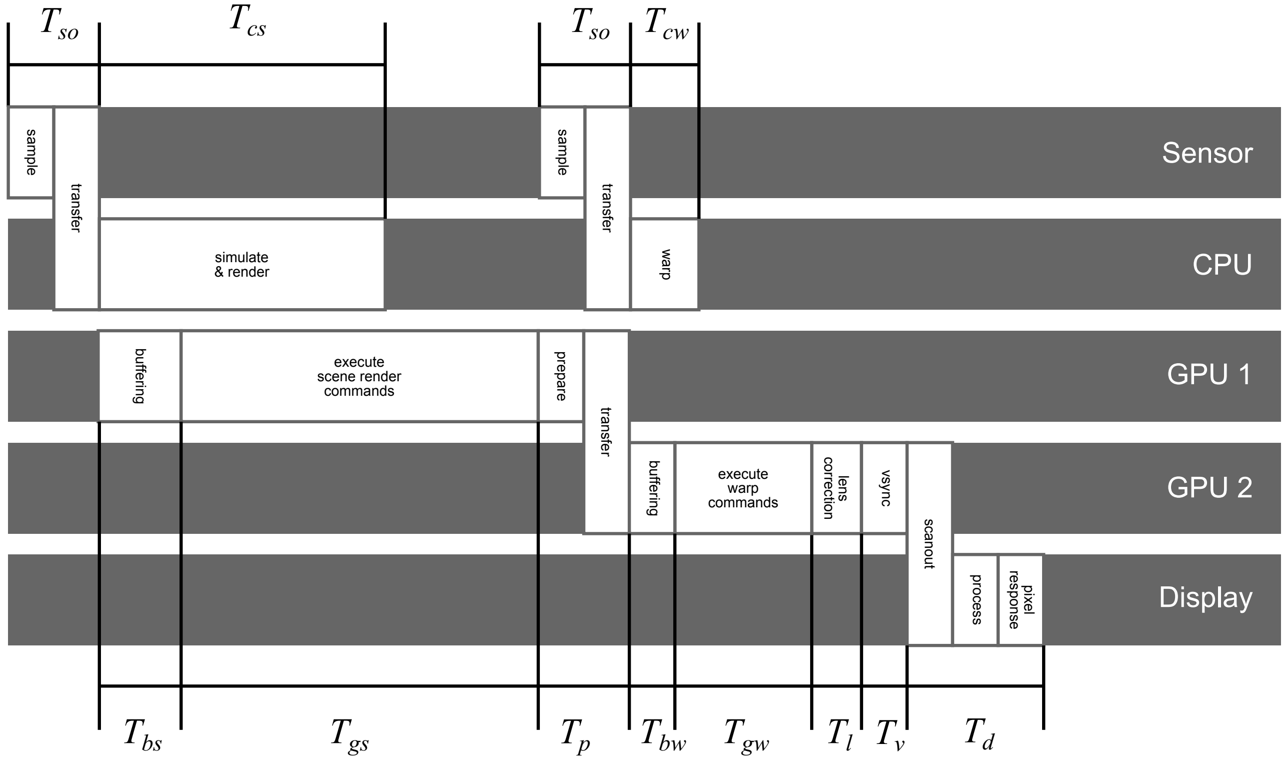


Figure 3: The sources of latency in image warping for a dual GPU system. For a single GPU system, all rendering work on the second GPU is shifted to the first and the cross-GPU transfer no longer exists

3.4 Reprojection Algorithm

The first warping algorithm we present is a reprojection of the rendered image as a textured screen-aligned quad. Consider the following variant of the matrix stack used to transform mesh vertices into screen-space (used ubiquitously in conventional rendering) for time t

$$\vec{v_t}' = PHBM\vec{v} \tag{8}$$

where P is the projection matrix, M is the model/world matrix, and the view matrix (normally V) is split into the simulation controlled body matrix B and the head coupled matrix H. The matrix to reproject the transformed vertex $\vec{v_t}'$ to its position at time $t+1$ is given by

$$R = PH_{t+1}H_t^{-1}P^{-1} \tag{9}$$

assuming there is no change in B or M. The full process for this method of image warping is therefore

1. Sample the head position to obtain H_t
2. Render the scene conventionally to an off-screen texture, using the transformation given in Equation 8
3. Sample the head position to obtain H_{t+1}
4. Render a quad mapped with the texture from step 2, and aligned to the rear face of the screen-space frustum to the back buffer, using the transformation matrix given in Equation 9
5. Flip the back buffer to display the rendered quad to the user

Since H is the only variable that is updated for the warping, coupled head orientation and translations are the only types of motion enhanced by this method. Changes to B are not considered in order to fully decouple the warping from scene simulation, although in future work it could be predicted. M is likewise ignored due to being simulation controlled, but is also per-object, making it significantly more complicated to predict and store than B. It should also be noted that because this method does not consider frame depth information, the head translation will be imperfectly warped, and objects will be distorted as if their depth is the same as the distance to the far clipping plane. The effect of this error is discussed in detail in Section 4.5.

This warping method runs in constant time for a given screen resolution, and is extremely fast as it consists of just 4 vertices transformed by a single matrix, and a single texture lookup per-pixel. This makes it appropriate for use on low-end GPUs (particularly iGPUs) which are too slow to perform more sophisticated warping algorithms.

3.5 Raytrace Algorithm

The second warping algorithm corrects for the translation error introduced in the reprojection method. By utilising the depth buffer information that accompanies key frames, the distance to scene objects need no longer be assumed constant. The disadvantage of this is that the image warping can no longer be performed by a forward texture lookup, and must resort to a linear search through the frame to find the new colour value. Forward lookup based methods such as point splatting do exist (Smit et al., 2007), but raytracing has attractive properties that will be discussed in later sections.

Effectively, the core difference between the two algorithms is that in reprojection, the simulation frame is rendered as a flat quadrilateral, while under raytracing it is rendered as a frustum-shaped height map. To draw this height map, each pixel of the screen is treated as a ray that is cast into the screen and through the height map. The colour of the height map where the ray intersects with it is used to colour the pixel for which the ray was generated.

To perform this intersection, firstly each ray's start and end point is created in screen-space, and then transformed into the coordinate system of the height map by multiplying by R^{-1} (from Equation 9). A ray-plane intersection is then performed against planes at the front and back of the height map's coordinate system to give texture coordinates for the start UV_{start} and end UV_{end} of a linear search. The linear search iterates across the depth values of the simulation frame until it meets the following inequality.

$$sample_{depth}(i * UV_{start} + (1 - i) * UV_{end}) \leq i \quad (10)$$

where $0 < i < 1$ and represents the offset and depth of the search, and n_d is the number of pixels (and hence number of samples tested) between UV_{start} and UV_{end}. Once the inequality returns true, the colour at the successful UV coordinate is returned as the result of the raycast.

This technique's runtime is proportional to the amount of translation of the viewing position. This is because the amount of translation directly affects how separated the start and end points are of the linear search, which subsequently dictates the number of depth samples (n_d) needed: the major performance cost of this method. Another variable that controls n_d is the near clipping distance, which is normally recommended to be made as large as possible to prevent Z-fighting. The positive side of this is that for no translation, no depth lookups are required, giving comparable performance to reprojection. The average number of texture samples needed is approximated by

$$n_d \propto \frac{translation}{clip_{near}} \quad (11)$$

assuming $clip_{far} \gg clip_{near}$.

4 Results & Discussion

The major contribution of this paper is the consideration of how real-world factors influence the practicality of these frame rate and latency enhancement methods. This section contains the discussion of these factors.

4.1 Required Program Modifications

One of the factors influencing the practicality of image warping is the extent to which it requires changes to the architecture of a piece of software in order to accommodate the enhancement. What's desirable is requiring few changes to the conventional rendering process, as well as being compatible with a large range of lighting and shading techniques.

In this regard both methods perform extremely well. Both methods attach to the very end of the rendering process, making the integration point between the simulation software and the enhancement very small. The only process change is that the final rendered image frame is stored in an off-screen texture, rather than the back-buffer of the swap-chain. For reprojection, this is the only technical requirement, while for raytracing, there is the additional requirement that the depth buffer must also present alongside the off-screen texture and with valid data.

While depth buffers are ubiquitously used, some programs render the scene as multiple layers to improve Z precision, clearing the depth buffer in-between rendering layers. This is incompatible with the raytracing algorithm, as it expects objects to have correct depth information. This does however have the benefit of preventing excessive use of this optimisation, which can cause incorrect occlusion where layers overlap, something barely noticeable on conventional displays, but quite distracting on stereoscopic displays (including HMDs).

While the requirement of a motion-field buffer would allow simulation motion smoothing (Smit et al., 2007), this would require a large step-up in the intrusiveness of the enhancement which is one of the reasons it was not considered in this paper.

4.2 Head Rotation and Translation

The image warping enhancement described in this paper is concerned with improving frame rate and latency specifically for head tracking. Furthermore, of the two components of tracked head motion, orientation and translation, orientation is strongly prioritised over translation. The reasons for this are three-fold.

Firstly, we consider coupled head orientation to be the major immersion factor for HMDs, as it is what allows the user to naturally look around the virtual environment.

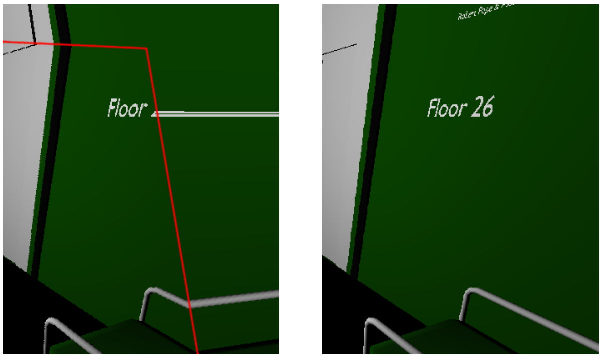
Secondly, it is the latency in orientation tracking that primarily causes motion sickness: by making the world appear to wobble, roll and lag with fast head movement, and by preventing the vestibulo-ocular reflex from correctly stabilising the user's view of the scene.

Lastly, the reference HMD (Oculus Rift developer kit) only contains an orientation tracking system, and none for positional tracking. This means that any translation in the viewpoint caused by this tracking is purely from correcting the rotation to revolve around the user's neck, and not around the midpoint of the user's eyes. This produced translation is quite small, and so becomes negligible compared to changes in orientation.

Another point of consideration is the instantaneous angular velocity of head rotation. This is important as the size of certain types of artefacts (Sect. 4.3 & 4.4), the performance of the raytrace algorithm (Sect. 3.5), and sample-and-hold blurring (Sect. 4.7) all depend on this velocity. The details of this relationship are discussed in the relevant sections, but in general, the slower the head rotation the better. This turns out to be ideal regarding actual usage, as our experience suggests long periods of small or no rotation, with occasional bursts of medium speed rotation. Rapid rotation is rare, and even suppressed due to the weight and inertia of the HMD.

4.3 Fringe Artefacts

The major visible artefact caused by image warping with HMDs is holes around the edges of the screen, what we call *fringe artefacts*. This is caused by rotation of the viewpoint, which results in sampling beyond the limits of usable data around the edge in the direction of rotation. Raytracing naturally fills these holes, while reprojection can be trivially adapted to fill them in the same manner: by transforming the *texture coordinates* of the quad's vertices by R instead of their *positions*. Both of these work by clamping the sampled texture coordinates to stay within the bounds of valid data, effectively stretching the edges of the key frame over the holes.



(a) Fringe error caused by warp; the red line separates the valid image data and the error

(b) How the frame would look if rendered without artefacts

Figure 4: Example of a fringe artefact, with comparison to an error-free rendering

Since these artefacts always appear at the edges of the screen, and due to the large field-of-view provided by the evaluated HMD, these artefacts typically appear in the user's peripheral vision. Peripheral vision is more sensitive to motion than the rest of the field-of-view, which may emphasise the flickering caused by the artefacts rapidly appearing and disappearing as they are corrected. Peripheral visual acuity is however worse than central vision (foveal), and the geometry of the Oculus Rift lenses prevents the user from looking directly at objects in the periphery of the HMD, meaning fringe artefacts are very difficult to *directly* observe.

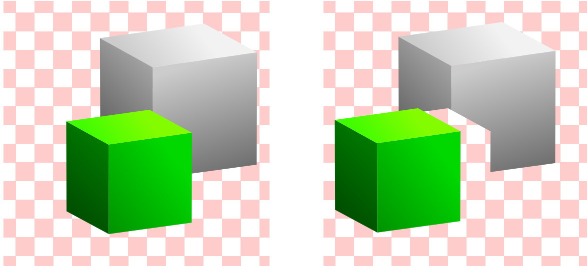
In addition, HMD rendering is frequently designed to allow for rendering beyond the normal edges of the display. This is done to compensate for a shrinking of the rendered image due to lens distortion correction. This can be reused as-is to hide fringe artefacts by pushing them further off the sides of the display, at the cost of increasing T_{gs} due to rendering more pixels than are visible.

4.4 Edge Artefacts

When translating the viewpoint position using the raytrace algorithm, near objects will be shifted by the warp more than distant objects. Where depth in the scene changes suddenly, such as when a near object occludes a far object, the two objects may be warped away from each other creating a visible artefact along the edge between them. Because our raytracing algorithm treats the height map as continuous, these artefacts do not appear as holes, but rather as a stretching of the colour at the edge.

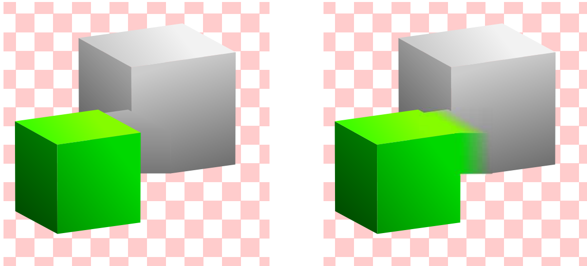
The algorithm may be trivially adapted to stretch image data across the gap in one of two ways. The first approach fills the gap using the farther side of the edge, while the second approach is a linear interpolation of the near and far sides. The benefit of the first method is that it more realistically handles object occlusions, which are the most common cause of edge artefacts, albeit at the expense of being incapable of handling multisampled anti-aliasing. The linear interpolation approach removes this incompatibility, but at the cost of having less realistic fill appearance. A comparison of the the two filling methods can be seen in Figures 5c & 5d.

The size of edge artefacts is directly related to the amount of coupled head translation, which is in turn related to the velocity of head rotation (Sect. 4.2), and to the ratio of the object depths as given by Equation 12.



(a) Simulation frame

(b) Display frame made by warping the simulation frame; missing data causes visible hole



(c) Hole filled using far edge

(d) Hole filled by interpolating between both near and far edges

Figure 5: Example of an edge artefact caused by moving the viewpoint to the right, as well as the two filling methods that could be used to mask it

$$error \propto translation \times \left(\frac{distance_{far} - distance_{near}}{distance_{far} \times distance_{near}} \right) \tag{12}$$

From this equation, it can be seen that as the objects get farther away, or closer together, the error approaches zero. Slow head rotation, coupled with objects at reasonably far distance and frequent key frame generation all serve to minimise the visibility of edge artefacts. Due to the many variables involved in their size and contrast, it is difficult to give exact values for these variables beyond which these artefacts become negligible; however the presence of sample-and-hold blurring significantly reduces their visibility, which is discussed in Section 4.7.

4.5 Translation Errors in Reprojection

It was mentioned within the description of the reprojection algorithm, that while the method attempts to correct for viewpoint translations, it is unable to do so without error due to ignoring depth information. The effect of this is that translation is under-compensated for near objects, but approximately correct for far objects. The amount of error can be determined through Equation 12 where $distance_{near}$ is the distance to the object, and $distance_{far}$ is the distance to the far clip plane. The appearance of this error is that, during head rotation, nearby objects jerk slightly in the opposite direction of the rotation.

Considering the points discussed in Section 4.2 along with the fact that for typical virtual environments most objects are at reasonable distances, the visibility of these errors is quite small.

4.6 Head-coupled Scene Objects

One of the major assumptions of these image warping methods is that no scene objects are coupled to the head position or orientation. Unfortunately there are a few common scenarios that violate this assumption. The most common examples of this are the head-up display and other types of 2D user interfaces. These are drawn at fixed positions on-screen and so may be modelled as scene objects coupled to the camera's position. Examples of other common coupled 3D objects are parts of the player's avatar, particularly the hands and held objects.

The violation in the warping model occurs because these types of objects are not transformed by a matrix stack of the form given by Equation 8. The visible effect of this is that these objects will judder in the direction of any head motion, instead of remaining stationary (relative to the user's head).

The simplest solution to this problem is to avoid using these types of head-coupled objects. Some video games already follow this model and do not show any 2D HUD, and opt to display the equivalent information entirely using in-game objects.

Another option is to shift the rendering of head-coupled objects themselves into the warping pass. There are three main disadvantage to this. It deteriorates the performance of the warping pass due to the increase in rendering complexity, although not significantly as most HUDs tend to be reasonably simple. It more tightly couples the warping pass to the simulation logic, making it more difficult to run them at independent rates. And finally it increases the intrusiveness of the enhancement by requiring more deep changes to the conventional rendering process.

A middle ground could be to modify the warping methods to detect head-coupled objects and then avoid warping them. This would require some way to differentiate between head-coupled and normal objects. Since head-coupled objects are typically rendered in front of all other scene objects, a simple solution would be to apply a threshold to the depth buffer, where objects below the threshold are not warped, while those behind are. While this is trivial to implement, it does introduce edge and transparency artefacts which, due to the great difference in depth between HUD and scene, would be significantly more visible than their manifestations in ordinary scene objects.

4.7 Sample-and-hold Blurring

One of the most interesting factors that deserves attention is a blurring effect due to the finite frame rate and sample-and-hold nature of the HMDs display. This appears as a linear blur during head rotation when fixated on a stationary point in the scene. While a similar blur also occurs when fixated on moving objects, whether on a conventional monitor or HMD, it is how the blur caused by head rotation interacts with and masks warping artefacts that is of particular interest.

The reason this occurs is that while the vestibulo-ocular reflex causes continuous tracking of the object on the display, the motion of the object on the display itself occurs in discrete steps. This means that the focused image of the object follows a rapid saw-tooth pattern across the surface of the retina, causing it to be perceived as blurred since it moves faster than the threshold of persistence of vision.

It has been noted previously that many types of artefacts depend on head translation, and subsequently head rotation. Because the size of this blur also depends on the amount of head rotation, it will naturally mask these artefacts by blurring them by an amount directly proportional to their size. We find the size of this blur effect due to head rotation to be approximated by the following equation

$$blur\ width = \frac{\omega}{frame\ rate} \tag{13}$$

where ω is the angular velocity of the user's head in radians per second, and *blur width* is the visual angle of the perceived blur in radians.

The blur is itself a visual artefact however, albeit one that cannot be reduced much using software. The blur can be minimised by ensuring that the tracking frame rate is at least equal to the display's refresh rate, 60Hz in the case of the Oculus Rift. Otherwise the blur may only be reduced though redesigning the HMD hardware to support a higher refresh rate, and/or strobe its image.

5 Future Work

While the raytrace warp (Sect. 3.5) has fast performance for minimal head movement, the fact that performance deteriorates linearly with the speed of head motion makes the method unusable on low and mid-range hardware. It does not help that the major performance cost is texture bandwidth, which is particularly limited on iGPUs as it must be shared with the CPU. Therefore more work is needed to either optimise this algorithm, or to find another of better performance characteristics that supplements the reprojection algorithm as a higher quality warping method for faster systems.

Another issue with the proposed enhancement architecture is timing issues caused by GPU command buffering. In order to avoid excessive user-space to kernel mode switches on the CPU, and to prevent the GPU from stalling with no work to do, consumer GPU drivers and hardware will buffer rendering commands. This helps to improve *throughput* at the expense of *latency*. This is preferable on conventional display types which are latency-insensitive, but highly undesirable for VR systems such as HMDs which are not. This causes further issues on a single GPU system where the concurrent simulation and warping logic interleave their rendering commands, making controlling the timing of them very difficult. Future work is needed to find ways to avoid command buffering during the generation of display frames (while it remains acceptable for simulation frames), which might be possible though finding a way to give higher priority to warping commands and by manually flushing the command buffers.

In this paper we have given a technical discussion on the perception of the benefits and artefacts of the warping algorithms, and personal experience supports our conclusions. However, a proper user study is required to ensure the proposed enhancements really do give a better user experience with actual users and in representative applications.

The last major point in need of further attention is the practicality of the proposed enhancements as injectable algorithms. Since the enhancement is so minimally intrusive, it might be possible to modify it so that it can be retroactively applied to existing HMD applications in the vein of NVIDIAs 3D Vision (NVIDIA Corporation, 2013) or our method for conversion of desktop VR to support head-coupled perspective (Li et al., 2012).

6 Conclusion

We have presented and discussed the use of and image warping based enhancement for improving frame rate and head tracking latency, that is appropriate for use with consumer HMDs (particularly the Oculus Rift) and applications.

Two algorithms are suggested for performing the actual warping, one based on reprojection and the other on raytracing, that are suitable for use in PC systems with either one or two GPUs, and of variable processing power. The reprojection algorithm has the constant performance of rendering a single textured quad, as does the raytracing method for slow head rotations, although the raytracing computation cost increases linearly with head rotation speed. These warping methods exploit properties of real-world HMD usage such as minimal head translation and require only small changes to the conventional rendering process for HMDs.

Furthermore, we have described two types of error that are visible at the edges of the display and along object edges during head motion, and how the sample-and-hold nature of HMD displays produces a blur which helps to mask them.

Lastly we believe this type of enhancement in their current form should prove valuable for improving user experience in many consumer HMD applications, and even more with the suggested slight modifications.

References

Azuma, R. and Bishop, G. (1994), Improving static and dynamic registration in an optical see-through hmd, *in* 'Proceedings of the 21st annual conference on Computer graphics and interactive techniques', SIGGRAPH '94, ACM, New York, NY, USA, pp. 197–204.

Kijima, R. and Ojika, T. (2002), Reflex hmd to compensate lag and correction of derivative deformation, *in* 'Virtual Reality, 2002. Proceedings. IEEE', pp. 172–179.

Li, I. K. Y., Peek, E. M., Wünsche, B. C. and Lutteroth, C. (2012), Enhancing 3d applications using stereoscopic 3d and motion parallax, *in* 'Proceedings of the Thirteenth Australasian User Interface Conference - Volume 126', AUIC '12, Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 59–68.

Mark, W. R., McMillan, L. and Bishop, G. (1997), Post-rendering 3d warping, *in* 'Proceedings of the 1997 symposium on Interactive 3D graphics', I3D '97, ACM, New York, NY, USA, pp. 7–ff.

NVIDIA Corporation (2013), 'Nvidia 3d vision'.
URL: *http://www.nvidia.com/object/3d-vision-main.html*

Oculus VR (2012), 'Oculus rift - virtual reality headset for 3d gaming'.
URL: *http://www.oculusvr.com/*

Oculus VR (2013), 'John carmack joins oculus as cto'.
URL: *http://www.oculusvr.com/blog/john-carmack-joins-oculus-as-cto/*

Olano, M., Cohen, J., Mine, M. and Bishop, G. (1995), Combatting rendering latency, *in* 'Proceedings of the 1995 symposium on Interactive 3D graphics', I3D '95, ACM, New York, NY, USA, pp. 19–ff.

Peek, E. M., Wünsche, B. C. and Lutteroth, C. (2013), More for less: Fast image warping for improving the appearance of head tracking on hmds, *in* 'Image and Vision Computing New Zealand, 2013. IVCNZ '13. 28th International Conference'.

Smit, F. A., van Liere, R. and Fröhlich, B. (2007), The design and implementation of a vr-architecture for smooth motion, *in* 'Proceedings of the 2007 ACM symposium on Virtual reality software and technology', VRST '07, ACM, New York, NY, USA, pp. 153–156.

Smit, F. A., van Liere, R. and Fröhlich, B. (2008), An image-warping vr-architecture: design, implementation and applications, *in* 'Proceedings of the 2008 ACM symposium on Virtual reality software and technology', VRST '08, ACM, New York, NY, USA, pp. 115–122.

Smit, F., Van Liere, R., Beck, S. and Froehlich, B. (2009), An image-warping architecture for vr: Low latency versus image quality, *in* 'Virtual Reality Conference, 2009. VR 2009. IEEE', pp. 27–34.

Valve Corporation (2013), 'Steam hardware & software survey: July 2013'.
URL: *http://store.steampowered.com/hwsurvey*

Spatial Play Effects in a Tangible Game with an F-Formation of Multiple Players

Manuela Jungmann
SNT- DICA Lab
University of Luxembourg
Campus Kirchberg, Luxembourg
manuela.jungmann@uni.lu

Richard Cox
MArVL Lab
Monash University
Victoria, Australia
richard.cox@monash.edu

Geraldine Fitzpatrick
HCI-Group
Technische Universität Wien
Vienna, Austria
geraldine.fitzpatrick@tuwien.ac.at

Abstract

Drawing on Kendon's F-formation framework of social interaction, we analysed the game-space activity of collocated players engaged in a tangible multiplayer game. Game input from groups of 3 players interacting competitively in a natural spatial arrangement via balance-boards requiring whole-body movements was logged and analysed quantitatively. The spatial analysis of a range of players' activities in game-space revealed synergistic effects combining perceptual-motor factors with game-strategy behaviour which were reflected in preferred game-board playing regions. The findings illustrate the importance for HCI designers of considering interactions between human spatial behaviour, physical space and virtual game-space as games become increasingly embodied and social.

Keywords: Game analytics, tangible, embodied, multiplayer, digital game, collocation, F-formation, spatial analysis.

1 Introduction

Tangible multiplayer games are well-established and popular. Successful commercial interfaces of the genre are the Nintendo Wii gaming console played with the Wiimote and Nunchuck, and Microsoft's ReacTable played with tokens. Games played with these interfaces are embodied through the use of physical objects to interact with digital content. The games are physically engaging because the players' movements determine the game-play, and they make for rich social interaction because the players are collocated and the tangible interaction design includes both perceptual and motor elements (Djajadiningrat, Overbeeke & Wensveen, 2004). An important feature of tangible interaction is the direct engagement with tangible, non-digital artefacts through object-specific manipulations, which may involve the entire body (Hornecker, 2005). While the degrees of embodiment and social interaction vary depending on the design of the interaction and the nature of the game, players share themselves through physical expression, context and space (Buur, Jensen & Djajadiningrat, 2004).

Typically, designers of games with these interfaces focus the design on the virtual space while the structure of the physical environment is less-considered from the point of view of creating a game experience that seamlessly connects physical with virtual space to form a true hybrid (combined virtual and physical) space. In our research, we are interested in better understanding spatial aspects of play-areas and how they can be configured so that physical and virtual spaces come together in a tangible interaction experience for players. Based on this research interest, we designed a multiplayer game that exploits spatial relationships through players' position and orientation in the interface and the physical dimensionality of board games in the gameplay. In the design of the game, we utilised a principled approach by incorporating natural spatial patterns formed in social encounters, which we shall describe further in section 1.2. We also drew on culturally familiar gaming contexts, to maximise a player's presence and physical engagement.

We opted to design our own game for the purpose of creating a research platform rather than modifying an existing game because our approach is equally inclusive of digital technologies and social elements that are spatially-determined (as in traditional, non-digital play and games, e.g. street or board games). While current game consoles such as the Wii still closely adhere to the spatial layout of conventional computer setups where the user faces the screen during the interaction, future collocated gaming will continue to develop towards spatial arrangements in which players are oriented towards each other during gameplay (e.g. Microsoft's holodeck gaming experience). Our research anticipates these future developments.

The focus of this paper is on the study of the players' interaction and spatial relations.

1.1 Related Literature

Not only games benefit from collocated tangible interfaces with embodied interaction, they are also found across many applications in HCI (Hornecker, 2005), particularly for tasks that encompass collaborative learning and designing. Spatial relationships, though, have mostly been considered in context-aware computing where space factors into the design of applications for devices reacting to a situational change (e.g. Schnädelbach, 2012). Even proxemic interaction mostly concerns itself with sensing proximity and the triggering of connectivity switches (e.g. Ballendat et al., 2010). To-date, the challenge with these applications for researchers has been the hardware design - the social dynamics of contexts in which spatial relationships are expanded into hybrid space is typically not explored.

On the other hand, researchers who have empirically studied tangible interaction spaces and games have mostly focussed on teasing out the performance benefits of tangible interfaces and applications by comparing them to more conventional GUIs. These investigations have demonstrated that spatial cognition is enhanced, specifically in collaborative tasks where tangible interfaces promote the discovery of spatial relations between virtual 3D objects (Maher & Kim, 2006), and that physical gestures are signals that improve communication among players (Speelpenning et al., 2011). A study of Wii players showed that players actively construct spatial awareness of each other by announcing their position in game-space (Voida et al., 2010). Yet, these studies do not reveal much about the relationships between physical movement, collocation, and the spatiality of hybrid spaces. Our study aims to address those issues. Sim-Suite, the game we designed, combines movement-based collocated gameplay with the natural spatial arrangement of social encounters (Kendon, 1990) - one in which players are aware of their co-players' presence and actions (i.e. co-players are within a player's arc of peripheral visual attention).

1.2 Natural Arrangement of Player Positions

Kendon (1990) studied how two or more people arrange themselves casually, when meeting in public spaces - "focussed encounters" in his terms. His investigations showed that people cooperate with each other to physically maintain a space between them to which all participants have direct and exclusive access, a space he describes as a "joint transactional space". Kendon noticed that there are two basic configurations of two-person interaction, and that these configurations repeat themselves when more than two people meet up in a focussed encounter. One configuration positions two people face-to-face and frontally across from each other, the other is when two people are arranged in a right angle to each other, in an L-shaped configuration. Both configurations come together in a three-person focussed encounter that Kendon called the F-formation system. In the structure of the F-formation system, Kendon identified three kinds of functional spaces (Fig 1): the o-space, or joint transactional space, is the central space between individuals where they establish systematic relations in those aspects of their behaviour that maintains the o-space. Surrounding the o-space is a narrower, intimate space called p-space where individual bodies and personal belongings are located. Newcomers must receive permission to enter the p-space because it requires all other participants to adjust their spacing. Beyond the p-space is the r-space, a kind of buffer space, which is actively monitored by the participants and non-participants (passers-by, for example). It is also where others may wait to gain entry into the p-space.

F-formations have been used to analyse, inter alia, spatial patterns of interaction in real-world physical environments (Marshall et al., 2011), virtual environments (Nguyen & Wachsmuth, 2011) and blended reality contexts (Dim & Kuflik, 2013). F-space conceptualisations have been applied to non-competitive tasks and social interactions. In our research, we focus on the F-formation system's physical component - the natural positioning and orientation of persons - to support the conceptualisation of the game design. We then study the gameplay in relation to the players' natural positioning in the game and Kendon's associated psychological meaning of cooperative and competitive engagement.

In our game design, we therefore emphasise the concept of embodied facilitation from the perspective of real-world physical structure (the player arrangement) and full-body movement to engage with the digital content (Hornecker, 2005). Kendon (1990), drawing on Cook (1970), describes how, in two-person interactions, face-to-face configurations are preferred for competitive interactions whereas L-shaped configurations are associated with cooperative interactions. In the game we designed, player positions and their orientation to each other reflect the F-formation (Fig. 1). Game outcome predictions are made on the basis of this arrangement (Aim, section 3.1.2).

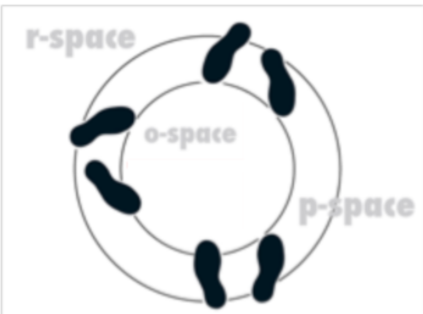


Figure 1: Kendon's F-formation system

2 The Tangible Game - Sim-Suite

Sim-Suite is a tangible game, designed as an installation with traditional and digital materials. It was conceived as cooperative group interaction that draws for its movement elements on street games and full-body play, and is for that reason more closely related to those games than to video games. It has three playing stations. Each station is associated with a unique token and is fitted with a balance-board on which the players move to play the game. Players move their tokens across fields on the virtual board via postural movements forward, backward, left and right on their balance-board. Balance-board movement is captured via infra-red sensors and transmitted to a Phidget interface and forms the input to a game created in Java. The playing stations are embedded in a platform that also houses a large LCD screen panel. The screen is centrally placed between the players, in a horizontal position, just below foot level. Players observe the screen via a window into the platform (Fig. 2). Each player has an equivalent view of the game graphics irrespective of which playing station she is using. A more detailed description of the game and its construction can be found in Jungmann & Fitzpatrick (2009).

Figure 2: Participants playing Sim-Suite

2.1 Game-play in Brief

The game is played with three types of virtual token (Fig. 3) that are placed onto a game-board consisting of a 10 X 10 array of fields. The game is played over a series of timed rounds. New rounds continue until one player wins the game by building a 5-token cruciform pattern using her tokens. At the start of each round a player's token is placed on a field at a quasi-random starting position on the board. The term 'quasi-random' is used because the game software employs an algorithm that maximises the distance of the start position from tokens of the same kind placed by the player in preceding rounds.

The tokens differentiated by shape and colour. Each playing station is associated with a token shape and colour (Fig. 3). A green circle is used by the centre player, a blue square for the player to the left of the centre, and a red triangle for the player to right of the centre position. The gameplay requires coordinating physical movement with strategic decisions of where to place the token on the game-board. A player wins a game when she is the first person to score two points and achieve the game objective. The objective is to create cruciform pattern by placing 5 tokens next to each other. Players navigate by 'teetering' on the balance-board leaning forward, backward, left and right. Over the course of the timed, consecutive rounds a player builds up her cruciform unless she chooses to use her token to block an opponent player; or because her token 'flies off' (i.e. she loses the token by failing to continuously move on the balance-board), or because she does not settle her token on an unoccupied field. To 'settle' her token on a field (i.e. occupy it) she must be in place on the field when the countdown timer ends the round. Note that players must devise an idiocratic manner of moving or teetering on the balance-board to 'claim' a field with their token. Static standing causes a time-out after two seconds and the virtual token flies off the game-board.

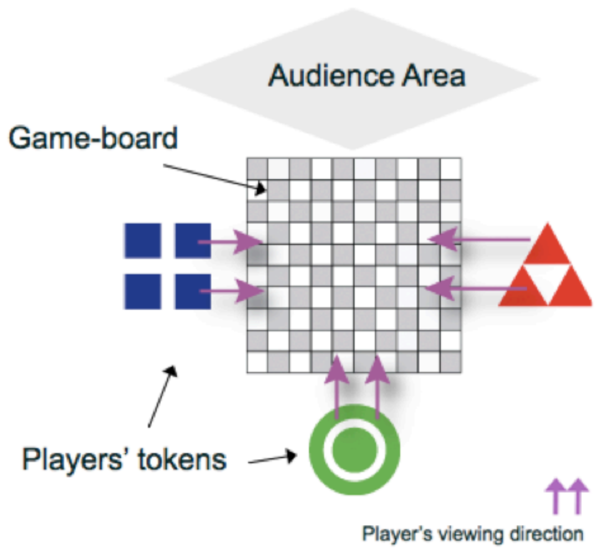


Figure 3: Spatial game-layout with tokens

Three players play simultaneously and must coordinate their actions with each other as well as with the virtual artefacts of the game. The players must also track the game-play of two opponents. The game's complexity emerges from the combination of physical movements and visually tracking multiple perceptual targets.

2.1.1 The role of Kendon's transactional space

The space between the players which is occupied by the virtual game-board represents Kendon's o-space, or joint transactional space. It is utilised for token movements and game-events, which are designed to create feedback loops to the p-space (the zone which is adjacent to the o-space) where players articulate expressive movements on the balance-board to animate tokens and navigate the events in the o-space. The r-space is where the audience assembles, to watch and cheer, and from which they can step up onto one of the balance-boards to play the game.

3 Method

3.1 System Instrumentation, Data Logging and Derivation of Measures

Gameplay metrics and user-metrics (Drachen & Schubert, 2013) in the form of balance-board movements, token placements, token movement paths across fields, and fields occupied by players were derived from raw data logs. The raw data was captured at millisecond resolution from both the game software and balance-board sensors. We also derived metrics of players' interactions with each other. These consisted of players' reciprocal interactions of offensive and defensive play, in which players respond to other players' actions on the game-board by using their tokens and the available interaction mechanisms. By offensive play we mean that players are placing their tokens in strategic positions for the construction of the cruciform token pattern that fulfils the game objective. Defensive play entails blocking each other and finding techniques to prevent other players from completing the winning token pattern.

Instrumenting the game has a key advantage compared to audio-visual recordings that would have been intrusive to players, and which are difficult to transcribe accurately. Our data logging and game analytics approach did not intrude upon participants' engagement in the busy, authentic and ambient festival settings in which data were collected. In our approach to the study of physical player interaction, we draw on a methodology that was applied in obtaining player social interaction data in a massively multiplayer online game (MMORPG). Ducheneaut & More (2004) studied anonymous online players in two locations in the game 'Star War Galaxies'. The researchers logged the real-time content that players posted in public chat boxes at specific locations in the game. Once the data was obtained, they devised a custom-made parser and proceeded in several detailed steps to extract patterns of information via quantitative analysis. We utilised a similar approach to the study of social interaction in our purpose-designed game, yet we focus on physical player behaviour by logging the sensor values that are coupled to the game-mechanics and generated through the movements of the balance-boards.

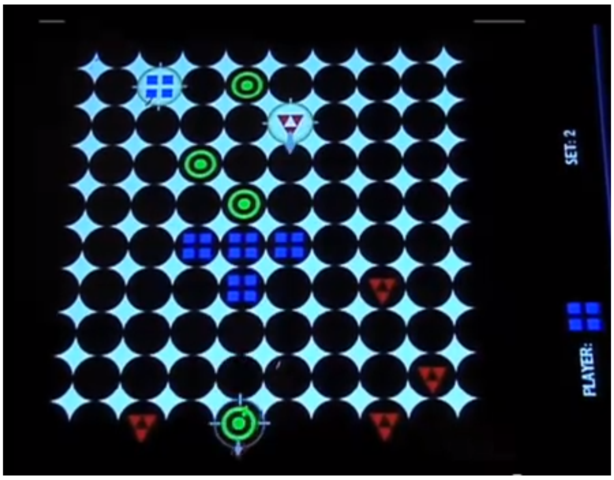


Figure 3: Sim-Suite's virtual game-board with the dashboard partially visible. The square (blue) player's almost completed cruciform has been blocked by (green) circle player. Tokens are highlighted white when placed by the system at round beginning.

3.1.1 Data Sample

Data were collected 'in the wild' from nearly 400 players at a number of digital media, music and science festivals in the UK. Participants played anonymously and they experienced the installation as part of the festivals' activities. The players' ages ranged from 10 to 60 years. Player triads were formed on an ad hoc basis and were comprised of all gender-mix permutations (i.e. all female players, all male players and mixed groups). At all times during each exhibition of Sim-Suite one of the authors (MJ) was present to administer the installation. This was required for setting-up the installation and for ensuring that it ran smoothly. Informal observations suggested that very few participants played repeat games. Player turnover was high in the busy and crowded festival settings.

Seventy-eight games were analysed. The criteria for selecting games for analysis were, 1. completed games with a definite winner; and, 2. games with at least seven rounds. The number of rounds in a game ranged from 7 to 24 (mean = 10.1 rpg). Average round duration was approximately 20 to 24 seconds. Short games (i.e. fewer than 7 rpg) were excluded from analysis because when viewed using a game replayer program, they seemed to be associated with players' casual experimentation and did not contain significant sequences of engaged 3-player interaction.

3.1.2 Aim

The aim was to investigate how players interacted in terms of joint spatial play actions and patterns in the shared game-space. Adopting a spatial game analytics approach (Drachen & Schubert, 2013), we aimed to examine the effects of player positions (F-formations) upon spatial patterns of play, considering players' relative positions with respect to each other and the game-board. On the basis of Kendon's F-formation framework, we wished to investigate whether circle player, the player at the central position of the triad, would be disadvantaged in form of game outcomes. Circle player is orthogonally aligned to the other two players forming two L-shaped configurations with them. We hypothesised from the F-formation framework that circle player's cooperative positioning in Sim-Suite's competitive game would disadvantage circle in relation to the other two players, whose face-to-face positioning supports competitive contexts (Fig.1).

3.2 Analyses

Win rates were unequal across players: Circle = 28 games won, Triangle = 32, Square=18. This was unexpected since our prediction was that circle player would be the outlier in terms of game win or loss rate. To follow-up, we looked at the relationship between a player's F-formation position and game strategies. A potential artefactual explanation was the possibility that square player's balance-board was less responsive than those of the other players. However the average movement rates over fields per round were comparable (respectively 9.8; 9.3 and 8.9 for circle, triangle and square), there was no evidence of defective operation.

Next, players' spatial play patterns on the game-board were characterised in terms of game 'state' sequences. For parsimony, the game-board size was reduced from a 10 x 10 grid of fields to quadrants (quads) each consisting of 25 fields (numbered 1-4, Fig. 2a). The quads' area was not arbitrarily chosen it was selected because it corresponds (approximately) to the 'useful field of view' of 15-20 degrees of subtended visual arc from the balance-board playing position. This is approximately the maximum area that a person can visually attend to in detail at any single point in time (Yokoi et al, 2006; Green & Bavelier, 2003).

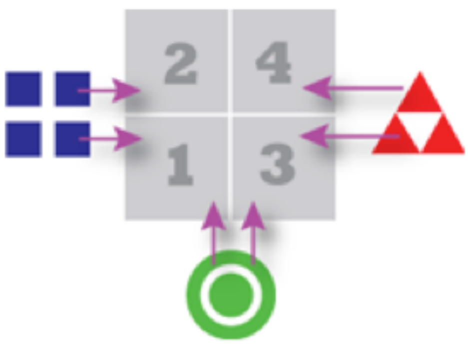


Figure 4: Game-board indicating the four quads used in the analysis.

Players' moves on the game-board occurred at different rates e.g. one player might be stationary and attempting to occupy a field whereas her co-players are moving (usually at different rates) in other parts of the game-board. Quad states are 'snapshots' of tokens placed within the quad division over the course of a round. Quad states were searched for using a computer program that identified states in which all 3 players occupied fields within a constrained time-band (<1 sec.). Approximately 3 quad states across all rounds of 78 games were sampled (2235 quad states in total).

4 Results

A 3 row by 4 column table was computed with the cells containing the frequency of field moves by each of the 3 tokens (players) in each of the 4 game-board quadrants. Field move activity differed across the 4 quads as a function of player's position (X^2=56.2, df=6, p<.0001). The stacked column graph in Fig. 5 shows the trends. Player position (token) is associated with different quad field move and occupancy frequencies. Triangle player is more active in quad 2 than in the other quads, circle in quads 3 and 4, and square in quads 1 and 3.

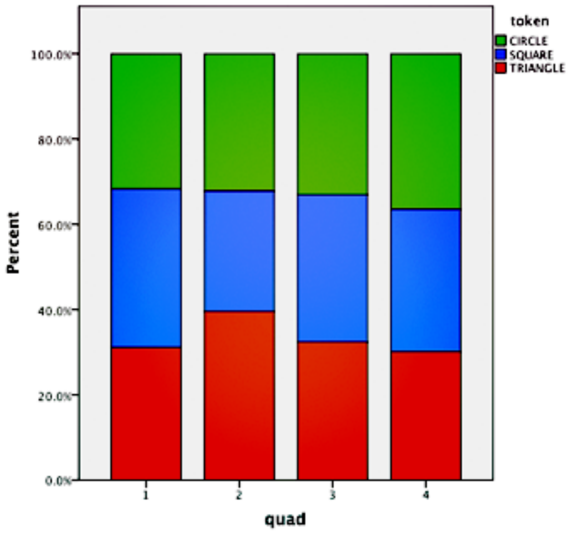


Figure 5: Proportion of game-board field activity in each quad for each player (token).

Further analyses identified three basic patterns of quad activity (Fig. 6). State 1 (S1) occurred when all 3 players were collocated in the same quad (which could be quad one, two, three or four, see Fig. 4). In State 2 (S2), two players were collocated in one quad with the third in a different quad. In State 3 (S3) each of the 3 players was active in a different quad. Players rarely all play in the same quadrant and tend to avoid such crowding. Of the various possible S2 sub-configurations, each token was equally likely to be the player in the different quad (circle 18.6%; triangle 18.7% and square 17.6%).

A closer look at S2 sub-patterns revealed that circle showed a marked tendency to play in either quad 3 or 4 when the other players were in quad 1, compared to the other permutations of S2 quad arrangement. In S2, triangle markedly favoured quad 2 when the other tokens are in quads 3 or 4. There was also a marked tendency for square to be active in quad 1 when the other two tokens are in quad 3.

Hence each player seemed to seek out different 'sociofugal' spaces (Sommer, 1967) in which they could evade the other players.

In S3 states, each token is active in a different quad. The most frequently occurring configuration for this play state was for circle in quad 4, square in quad 3, and triangle in quad 2 (6.5% of S3 states). Twenty-four different S3 configurations were observed. The 10 most frequent configurations accounted for 51% of S3 states. Within these, circle was most frequently active in quads 3 or 4, (70%), square in quads 1 or 3 (70%) and triangle in quad 2 (50% cf quad 1, 20%; quad 3, 20%; quad 4, 10%).

When considered from each player's egocentric play perspective the S2 and S3 patterns show consistent trends (described below).

We also examined the quads in which winning cruciform patterns were built. On games won by triangle, 66% were won with a cruciform built in quad 2 (25%) or quad 3 (41%). For square, 72% of games were won in either quad 1 (33%) or quad 4 (39%). Circle token won most often in quad 4 (36%). For circle, quad 3 or quad 4 located winning patterns accounting for 57% of games won.

Considered from each player's play perspective, the two facing players tend to win along quad diagonals. The diagonals are mirrored - nearest right (quad 1), furthest left (quad 4) for square and nearest left (quad 3) and furthest right (quad 2) for triangle. Circle, on the other hand, wins most frequently in quads 3 and 4 which are horizontally arrayed and distal to that player.

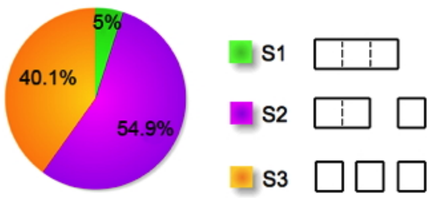


Figure 6: Distribution of gamestates.

5 Discussion

The results strongly suggest that a player's position in physical space - i.e. her play situation relative to those of her opponents and to the game-board - systematically influences her spatial behaviour in the virtual game-space. Systematic effects were observed across a range of different game behaviours. These included 1. the frequency of overall activity across quadrants, 2. quadrant preferences within two of the 'uncrowded' play state configurations (S2, S3) and 3. game-board areas associated with winning token placements. Considered from their egocentric play-position perspectives the two facing players show consistent preferences for playing on the right-hand side of the game-board (triangle in quad 2, square in quads 1 and 3). The player 'to the side' (circle) showed the same tendency to play on the right-hand side, which involved the quads that were on the perpendicular axis (quad 3 & 4) to those of triangle and square. Taken together, these results indicate a relatively strong tendency for players to prefer to play more in their distal and/or right-hand perceptual-motor fields than in their left-hand and nearer game-board areas. In addition, players appear to seek 'safer' game-board regions that are as far away from their opponents' scrutiny as possible. They also seek quads on the 'open' side when to do so is consistent with the 'distal-right' perceptual-motor bias. These spatial gameplay effects are summarised in Figure 7.

Asymmetrical spatial behavioural effects akin to the 'distal right' tendency we observe have also been demonstrated in virtual environments by Gerin-Lajoie et al. (2008). Their study of obstacle-passing by pedestrians walking in virtual and real environments revealed that when participants passed obstacles they tended to require significantly more personal space when the obstacle was on their non-dominant (usually left) side. This bias is consistent with the tendency we observe in our results.

In the study reported here, the handedness of the (self-selected) player sample could reasonably be assumed to reflect that of the general population (i.e. approximately 90% right-handed, 10% left-handed). However, controlling the balance-board required postural adjustments mediated via the legs and feet and therefore leg preference and 'footedness' of participants would be expected to exert a greater influence on balance-board performance than handedness per se.

Research findings on the relationship between handedness and footedness suggest that for responses such as tapping speed, right foot responses are faster in right-handed people but also in 62.5% of left- handed respondents (Peters & Durding, 1979; Peters, 1988). It is reported that, in general "humans are typically right-footed for actions of mobilization and left-sided for postural stabilization" (Sadeghi et al, 2000, p.37). Peters (1988) states that "the specialization for the right foot can simply be seen within the larger context of a preference for focusing on the limbs of the right body half in the realization of a movement goal" (p.190).

Such preferences have real-world implications. For example, a study of shoppers' in-store supermarket travel paths showed that 11 out 14 "canonical paths" identified via RFID tagging were ones in which people walked a right-hand path upon entering the store (Larson et al., 2005).

In the context of playing Sim-Suite the movement goal consists of balance-board 'teetering' and players may have tended to use their left legs for postural stabilization and their right legs for producing 'teetering' responses. It seems likely that lateral asymmetries in movement control accounted at least in part for the spatial play patterns that we observed.

Further research on the implications of lateral asymmetry in perceptual-motor responses seems warranted as an important topic of further research for HCI particularly in embodied controller contexts where the influence of such effects might be underestimated.

In the case of circle and triangle players, the 'distal right' perceptual-motor factor interacts with the strategic need to seek the less scrutinised, player-free side of the game-board in order to minimize blocking by other players. This seems to their advantage as reflected in their higher game win rates compared to square. Square players also manifest a rightmost quad bias but tend to opt for the nearer quad rather than (from their perspective) the more distal quad 3. This is probably because quad 3 is very much right 'under the nose' of both of square's opponents.

5.1 Kendons L-shaped configuration: right versus left-side

The right-hand side play preference and right-side physical positioning between circle and triangle players naturally supported the forming of an L-shaped formation which, in the spatial context of the players positioning around the game-board, provided both players with an "outlet" towards the game-board's player-free side to focus on (Fig.7), and potentially diverted the focus from playing defensively against the partner in the L-shaped formation.

From the circle player's perspective, this would explain why the square player was adversely affected in terms of wins compared to the triangle player. The circle-triangle L-shaped relation regulates the spatial relationship between square and triangle players, because circle player's right-side play preference and right-side physical alignment with triangle player creates a natural rightward alignment between the circle and triangle players. In the circle-square L-shaped relation there is no physical right-side alignment for circle, and square does not access the right-side 'outlet', the player-free side, because that side is on square player's far left-side. Circle's right-side alignment with triangle player and triangle's right-side access to the player-free side seem to exert a strong influence on the overall play-dynamics of the players' interaction.

According to Kendon's circular F-formation system Sim-Suite's player-free, open side connects people in the 'focussed encounter' with the surrounding 'world'. In Sim-Suite we see a conceptual and physical interplay between o- and r-space. Players' seek out advantageous play-spaces on the game-board based on strategic contexts by connecting with the physical aspects of the bordering r-space (the player-free side), to step beyond the boundaries of the focussed encounter.

Sim-Suite's virtual and physical spaces are very closely intertwined. This feature, together with immersive gameplay and the use of virtual artefacts, strongly influences the players' spatial relationships. Tangible and embodied games offer a useful insight into how interpersonal dynamics in virtual space connect with physical space and perception.

6 Conclusion

Predictions based on Kendon's F-formation framework were partially supported in that the most frequent winner (triangle) was one of the players who occupied a face-to-face play position. However, the other facing player (square) lost more games than the non-facing player (circle). This finding did not support our prediction made on the basis of the F-formation framework. The players' play-area preferences we observed appear to interact with F-formation position effects in complex ways.

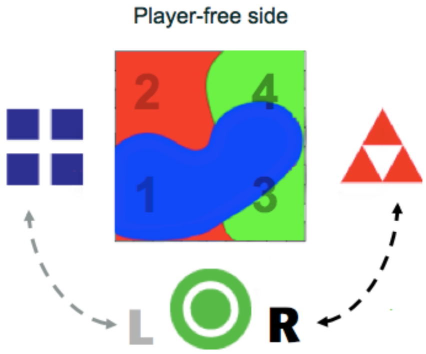


Figure 7: Summary of spatial play effects. The coloured fields indicate the tendency of token movement.

In Sim-Suite, the various o-space (game-board) regions vary widely in their strategic significance to each player. We have seen that several factors interact to affect game win/lose outcomes - these are F-formation player position, various o-spaces and perceptual-motor play biases. Further research is focussed on establishing these factors' relative strengths of influence - to identify which is the strongest influencing factor, which mediates a moderate effect and which have only weak effects.

In conclusion, our results indicate the importance of considering the interaction of several factors in embodied, collocated, multiplayer, blended physical and virtual gaming contexts. The factors include the lateral asymmetry in lower limb movement control and the players' spatial orientation with respect to each other and to the virtual game-space. With the recent advent of technologies such as virtual agents, augmented reality and wearables, blended contexts in which physical and virtual spaces are coextensive and will soon become ubiquitous. Designing game experiences that take into account spatial factors in blended contexts will therefore assume great importance. Designers could respond in different ways. One strategy would be to 'design out' player position advantages or disadvantages by changing the characteristics of particular game-space areas. Alternatively, future games could incorporate AI or machine learning approaches to analyse the spatial aspects of players' behaviour in real-time during gameplay so that in-game adjustments can be made. These could take the form of altering the sensitivity of a player's input system or re-configuring game-play elements 'on-the-fly' e.g. to encourage players to alter their positions relative to each other and/or the game-space. However designers choose to respond to player and game-space spatiality effects, the consideration of human spatial behaviour in blended contexts by the HCI community is likely to assume greater importance in the near future.

7 References

Ballendat, T., Marquardt N., & Greenberg, S. (2010) Proxemic interaction: Designing for a proximity and orientation-aware environment. In *Proceedings of the ACM Conference on Interactive Tabletops and Surfaces* (ITS '10), New York: ACM Press.

Buur, J., Jensen, M.V. & Djajadiningrat, T. (2004) Hands-only scenarios and video action walls: novel methods for tangible user interaction design. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques* (DIS '04), New York: ACM Press, 185-192.

Cook, M. (1970) Experiments on orientation and proxemics. *Human Relations*, **23**(1), 61-76.

Djajadiningrat, T., Overbeeke, K., & Wensveen, S. (2002) But how, Donald, tell us how? On the creation of meaning in interaction design through feedforward and inherent feedback. *Symposium on Designing Interactive Systems 2002*, New York: ACM Press, 285-291.

Dim, E. & Kuflik, T (2013) Social F-formation in blended reality. *Proceedings of the International Conference on Intelligent User Interfaces* (IUI'13), New York: ACM Press, 25-28.

Drachen, A. & Schubert, M. (2013) Spatial game analytics. In M. Seif El-Nasr, A. Drachen &A.Canossa (eds.), *Game Analytics: Maximizing the Value of Player Data.* London: Springer-Verlag, 365-402.

Ducheneaut, N. & Moore, R.J. (2004) The social side of gaming: a study of interaction patterns in a massively multiplayer online game. In J. Herbsleb & G. Olson (eds.), *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (CSCW '04), New York: ACM Press, 360-369.

Gerin-Lajoie, M., Richards, C.L., Fung, J. & McFadyen, B.J. (2008) Characteristics of personal space during obstacle circumvention in physical and virtual environments. *Gait & Posture*, **27**(2), 239-247.

Green, C.S. & Bavelier, D. (2007) Action video game experience alters the spatial resolution of vision. *Psychological Science*, **18**(1), 88-94.

Hornecker, E. (2005) A design theme for tangible interaction: Embodied facilitation. In H. Gellersen, K. Schmidt, Beaudoin-Lafon, M. & Mackay, W. (eds.) *Proceedings of the 9th European Conference on Computer Supported Cooperative Work* (E-CSCW'05), Dordrecht: Kluwer/Springer, 23-43.

Jungmann, M. & Fitzpatrick, G. (2009) Sim-Suite. In *Proceedings of the seventh ACM conference on Creativity and Cognition* (C&C '09). ACM, New York, NY, USA, 425-426.

Kendon, A. (1990) *Conducting interactions: Patterns of behaviour in focused encounters.* Cambridge University Press, UK.

Larson, J.S., Bradlow, E., & Fader, P.S. (2005) An exploratory look at supermarket shopping paths. *International Journal of Research in Marketing*, **22**, 395-414.

Maher, M. L., & Kim, M. J. (2006) Do tangible user interfaces impact spatial cognition in collaborative design? In *Proceedings of the 3rd International Conference on Cooperative design, visualization, and engineering* (CDVE '06), Lecture Notes in Computer Science, Vol. 4101, Berlin: Springer, 30-41.

Marshall, P., Rogers, Y. & Pantidi, N. (2011) Using F-formations to analyse spatial patterns of interaction in physical environments. In *Proceedings of the ACM conference on Computer Supported Cooperative Work* (CSCW '11), New York: ACM Press, 445-454.

Nguyen, N. & Wachsmuth, I. (2011) From body space to interaction space – modelling spatial cooperation for virtual humans. In *Proceedings of the The Autonomous Agents and MultiAgent Systems conference* (AAMAS'11), New York: ACM Press, 1047-1054.

Peters, M. (1988) Footedness: Asymmetries in foot preference skill and neuropsychological assessment of foot movement. *Psychological Bulletin*, **103**(2), 179-192.

Peters, M. & Durding, B.M. (1979) Footedness of left and right-handers. *The American Journal of Psychology*, **92**(1), 133-142.

Sadhegi, H., Allard, P., Prince, F. and Labelle, H. (2000) Symmetry and limb dominance in able-bodied gait: a review. *Gait and Posture*, **12**, 34-45.

Schnädelbach, H. (2012) Hybrid spatial topologies. *The Journal of Space Syntax*, **3**(2), 204-222.

Sommer, R. (1967) Sociofugal space. *American Journal of Sociology*, **72**(6), 654-660.

Speelpenning, T., Antle, A. N., Doering, T. & Van den Hoven, E. (2011) Exploring how tangible tools enable collaboration in a multi-touch table-top game. In *Proceedings of the 13th IFIP TC13 Conference on Human-Computer Interaction* (Interact '11), Lecture Notes on Computer Science, Vol. 6947, Berlin: Springer, 605-621.

Voida A., Carpendale, S. & Greenberg, S. (2010) The individual and the group in console gaming. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work* (CSCW '10), New York: ACM Press, 371-380.

Yokoi, K., Watanabe, K., & Kawai, T. (2006) Dynamic evaluation of distribution of visual attention during playing video game. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology* (ACE '06). New York: ACM Press.

8 Acknowledgement

We greatly appreciate the java code contribution for gameplay by Dr. John Revill and the construction of timber frame and balance-board encasements by Michael Ferguson. Programming assistance from Dr. Dave Barker-Plummer for deriving quad state 'snapshots' is also gratefully acknowledged.

Refining Personal and Social Presence in Virtual Meetings

Jesse Dean, Mark Apperley, Bill Rogers
Department of Computer Science, University of Waikato
Hamilton, New Zealand

m.apperley@waikato.ac.nz

Abstract

Virtual worlds show promise for conducting meetings and conferences without the need for physical travel. Current experience suggests the major limitation to the more widespread adoption and acceptance of virtual conferences is the failure of existing environments to provide a sense of immersion and engagement, or of 'being there'. These limitations are largely related to the appearance and control of avatars, and to the absence of means to convey non-verbal cues of facial expression and body language. This paper reports on a study involving the use of a mass-market motion sensor (Kinect™) and the mapping of participant action in the real world to avatar behaviour in the virtual world. This is coupled with full-motion video representation of participant's faces on their avatars to resolve both identity and facial expression issues. The outcomes of a small-group trial meeting based on this technology show a very positive reaction from participants, and the potential for further exploration of these concepts.

Keywords: virtual meeting, motion sensor, Kinect, avatar

1 Introduction

Video-conferencing technology, once a stuttering impediment to group discussion (Egido, 1988) has moved forward significantly over the past 25 years. High-speed internet connections and audio-visual integration with even low-cost personal and mobile computers, mean that reasonable quality video calls are available to all, and that the centralised technology of the video-conferencing suite is fast becoming an anachronism.

In hand with this ability to readily communicate via everyday technology, at one's desk, or even on the move, is the growing motivation to use this technology as a substitute or alternative to physical meetings. Not only is the budgetary cost of having people travel to meetings a factor in this shift (Lindeman, Reiners & Steed, 2009; Erikson, Shami, Kellogg & Levine, 2011) but there is increasing awareness of the cost to the environment of such travel (Arnfalk & Kogg, 2003). Personal travel accounts for almost one-third of all energy consumption in the developed world, and that energy is based almost exclusively on fossil fuels, a fast-dwindling resource, and accountable for the majority of greenhouse gas emissions (Mackay, 2008).

While Voice Over Internet Protocol (VOIP) audio/video telephony services such as Skype™ are widely and successfully used, they are essentially one-to-one, and although they can extend to multi-party conference calls, there is an absence of a sense of *being* anywhere (other than where the caller is) or of *being together*; the normal telephone sense of speaking-across-a-distance remains (see, for example, Steuer, 1992). On the other hand, virtual worlds, such as Second Life™ do provide a sense of place, and of being somewhere, and have successfully been used to provide virtual conference venues (Lindeman *et al*, 2009; Al Qahtani, 2010; Erikson *et al*, 2011). Meeting participants are represented by avatars, which they control, in a shared space, which may resemble a virtual meeting room, with furniture and projection screens. However, a significant shortcoming of using such environments for meetings is the major overhead for the participant of managing and controlling their avatar, and their view of the shared meeting space. These overheads become the dominant activity and significantly detract from the sense of presence, and engagement with the meeting (Al Qahtani, 2010).

This paper describes the development and evaluation of software (VMX) to reduce this overhead, and to refine the sense of personal and social presence in the virtual world (Dean, 2012). This work acknowledges the fact that the typical participant will be physically located in their own personal space (quite likely an office), and will be seated in front of a computer screen and keyboard. It uses a 3D motion-sensing input device to capture participant actions in this real space, which it then maps to view-controlling operations, such as panning and zooming, and avatar actions in the virtual space. Some avatar actions are directly mapped from the user's movement (eg, turning one's head), and others indirectly mapped (eg, pointing at a screen). Specific facilities are provided for presentations in the virtual space. The real-time image of the participant's face is superimposed on the avatar, so providing for the subtleties of facial expression.

The preliminary evaluation of VMX has shown the combination of these facilities are remarkably successful in promoting a sense of engagement with the meeting, and of "being there".

Section 2 of this paper provides a brief overview of relevant prior research and experience in relation to virtual meetings, personal and social presence in virtual environments, and the capabilities and use of mass-market motion sensing technologies in virtual world interaction. Section 3 then develops the requirements and explains the design for software to interpret participant actions and map these to avatar and camera control, recognizing the capabilities of the current technology. Details of the implementation of the prototype VMX software system based on the Kinect™ (Kinect) motion sensor, which incorporates and explores these ideas, and limitations and issues with this implementation, are summarized in Section 4. A trial evaluation of the environment is described in Section 5, including a discussion of the feedback from participants, and some further ideas which arose from the experience. In the conclusion (Section 6) the overall findings are discussed, and directions for future research with this technology are proposed.

2 Background

Research relating to virtual meetings and collaboration at a distance, be it through telephone, video or computer/communications technology, has a history of more than forty years (eg, Chapanis, Ochsman, Parrish & Weeks, 1972) and crosses a range of disciplines including technology, psychology, sociology, education and management/organisation. This paper is specifically concerned with the use of virtual worlds as places to conduct meetings or gatherings, which may range from small group discussions to large group presentations or performances. Within this context, it sets out to develop software based on mass-market motion sensing technologies to refine the concepts of personal and social presence in a virtual world, and by reducing the additional overheads of virtual participation, enhance the level of engagement.

2.1 Meetings in virtual worlds

In 2009, an IEEE Virtual Reality conference program committee meeting was conducted in Second Life (Lindeman *et al*, 2009). Some 39 members of the committee participated during the meeting, and on average attended about 60% of the approximately 9 hours total. Attendees were widely distributed across the globe. Although the overall reaction was favorable (in preference to travelling to a two-day meeting), participants felt it was not the same as a face-to-face meeting, with personal engagement less satisfactory in the virtual world. They found it difficult to identify other participants amongst the 'plastic' avatars, found the absence of body language an issue, and the document presentation tools less than ideal.

A somewhat larger and more ambitious meeting held in Second Life is described by Erikson et al (2011). A large corporate meeting spread over 3 days was attended by approximately 500 participants from around the world. It was a single track conference, with a schedule designed to accommodate attendees from all time zones. Overall, participants felt the technology worked well, but criticisms included: keynotes were wasted in the virtual environment, and would have been better simply streamed; avatars in general tended to look the same, and were difficult to distinguish or identify; feelings relating to social events were mixed, with some suggestion they need to be either highly structured or focused on quite small groups. In spite of these shortcomings, the authors present a positive view of the future potential for large group meetings in virtual environments.

A further small-scale study has provided much of the motivation for the work described in this paper (Al Qahtani, 2010). A trial meeting involving seven participants was held in a hired virtual meeting room in Second Life. The meeting involved a slide presentation followed by a discussion. Reactions were generally positive: participants found the virtual conference more immersive than a conventional video conference. However, largely in common with the studies described above, the following issues were identified:

(i) the need for a participant to overtly control their avatar is a distraction, and can dominate other activity;
(ii) providing gestures is laborious, and the gestures, which are stylized carbon-copies are unsatisfactory;
(iii) individual avatars can be difficult to identify, even in a small group;
(iv) there is an absence of any sense of audience response or level of engagement, such as subtle non-verbal cues like facial expression or eye-gaze direction;
(v) 'in-world' presentations are difficult to deliver for the avatar-as-presenter, and not easy to view by the participant;
(vi) general navigation within the world (getting to the conference room, sitting down, controlling one's view, etc) seems overly complex, particularly for those participants with no prior experience of Second Life;

These six points form the key issues which our VMX system attempts to address.

2.2 Presence

Although motivated by the potential use of virtual reality in psychological therapy, Schuemie *et al* (2001) provide a relevant and comprehensive review of research and issues related to the concept of presence in virtual worlds. Although their review is not conclusive, and they suggest that a sense of *presence* alone may not be the key to the successful virtual environment, the research they review does identify factors such as immersion, engagement, social interaction, naturalness, social reality and interpersonal communication cues as being important precursors to that sense of presence. A simple dichotomy between "real" and "virtual" worlds has also been questioned (Taylor, 2002), with a stress on the significance of the persona and appearance of the avatar in the virtual world contributing to the sense of presence and reality.

Discussions of presence often focus on the exchange of non-verbal communication with an emphasis on the use of facial expression to establish trust, particularly in negotiation situations (Purdy & Nye, 2000; Bekkering & Shim, 2006). In Al Qahtani's (2010) study, where the participants were all known to one another, it was identified that non-verbal communication would have been helpful for two purposes. First, to establish whether another participant was paying attention; in a virtual meeting there is always the possibility that a participant is AFK (away from their keyboard), or doing something not connected with the meeting, whereas a 'live' avatar imposes some social pressure to take part. Second, although sound location can be helpful in determining who is talking (DiVincenzi, 2011), it gives no idea whether their speech is being directed at someone specifically; an avatar with directional gaze could provide this cue.

2.3 Human motion sensing

Computer-based systems for sensing and analyzing human motion have evolved over several decades (Dean, 2012). Most recently applicable technology has evolved from the once complex, cumbersome, intrusive, inconvenient and expensive, to simple, low-cost, non-intrusive systems based on depth sensing cameras, of which Microsoft's Kinect is probably the most pervasive and well-supported.

Kinect and its supporting software was introduced by Microsoft as part of the Xbox 360 gaming system in late 2010 (Kinect). It is designed to track a human player, to allow them to interact with games by verbal command, body movement and gesture. The unit incorporates several components: microphones which pick up voice commands; an infrared projector which displays a pattern of infrared light; and two video cameras, one tuned to visible light and one to infrared. The visible light camera produces an ordinary image. The infrared camera provides data which can be converted to a depth view of the scene – for each pixel, measuring the distance between the camera and a point in the scene. Further analysis of depth information allows estimation of body position and configuration in 3D.

The utility of Kinect in improving video conferencing facilities has been demonstrated with the Kinected Conference project (DeVincenzi *et al*, 2011). This project has implemented several features that take advantage of the ability of the Kinect audio array to determine the position of a speaker in the field of view, and the ability of the Kinect depth camera to identify the spatial location of objects in the view of the video camera. By using this information the software is able to perform a number of visual enhancements to the video feed being sent to the remotely connected participants of the meeting, such as focusing the camera on speakers, freezing parts of the camera image, and overlaying spatially contextual graphics.

2.4 Background summary

Although conferences held in virtual worlds show promise, they are unlikely to become an unqualified alternative to face-to-face meetings. The limitations are very much associated with appearance and control of avatars (in subtle, fine and gross movement), as well as camera control. For a real sense of engagement, participants need to be relieved of the burden of overt avatar control (being puppeteers), yet be confident that the avatar is a reasonable representation of themselves and projection of their body language. Also, in order to be able to focus on conference participation, gross avatar movement and camera control (controlling your view) need to be as unobtrusive and natural as possible. Current consumer level motion sensing technology has the potential to assist in meeting these requirements.

3. VMX Design and Development

As noted earlier, there are many possible scenarios for virtual meetings. One possibility is to simulate a large conference, or a performance (Yong, 2003). In this situation a presenter might want information about audience response to their material. For an audience member, view of the participant is probably best provided by conventional streamed video (Erikson *et al*, 2011). The sense of being part of an audience, hearing others gasp or laugh and possibly exchanging occasional comments with an immediate neighbour, could be benefits of a more immersive virtual environment. For such scenarios, technical issues concerning the multiplexing and combination of large numbers of location/movement/video/audio feeds are challenging, while issues of maximising the benefit of detailed exchange are not so interesting. On the other hand, a small meeting (perhaps 4 to 10 people), provides a situation in which technical communication issues are manageable with current technology, and the intimacy and familiarity of such settings enables us to focus on the challenges of supporting issues of personal and social presence in a convincing manner. Accordingly, our work has focused on virtualising a small meeting. However, even in small meetings, seeing and talking to other participants is usually not enough; for example, frequently, one member will lead discussion by presenting a report or plan. We have therefore developed a system (VMX) to support a small meeting, with provision for members to share documents and make presentations. The work focuses on addressing the issues (as listed in Section 2.1) identified by Al Qahtani (2010) in experiments with meeting in Second Life.

3.1 Avatar control

The first issue to be addressed is that of avatar control. In Second Life, participants must be 'puppeteers', using keyboard and mouse controls to operate their avatars. The Second Life system gives good control of gross movement, allowing users to explore their world, albeit requiring some familiarity and experience with those controls. However, it provides poor fine control. While it is possible to orient the whole body, determining the gaze direction, it is not possible to refine gaze direction by eye movement. While it is possible to perform a number of animations; for example, to wave or jump up and down, by selecting from a menu, it is not possible to provide detail in that interaction, such as pointing to an item of interest. Issuing commands itself can be slow, although speed can improve if the participant has practiced with the software. For example, if a speaker asks for a show of hands in a vote, participants must search through a list of animations to find the appropriate one, and that may take some time. At best their speed of response is an indication of their speed of menu search, rather than a sign of their enthusiasm. The lack of fine detail means that there is no expressiveness in the gesture itself, as might distinguish a reluctant partial raise of the hand from something more enthusiastic. If the player does not issue commands, nothing happens to their avatars; they remain stationary. In particular it is impossible to distinguish the avatars of people quietly listening from those of people who have stopped participating altogether. We hypothesised that tracking a participant's body and using body motion to animate their avatar could have three advantages. First, it would not be necessary for the participant to issue commands, freeing them to focus on the content of the meeting. Second it would make fine grain animation possible, both in timing and motion, limited only by the speed and detail in which body movement was captured. Finally, it would help to show what a person was doing, even when they were not intentionally communicating (eg looking away from their screen).

We note that one of the advantages some perceive with a virtual meeting setting is the anonymity conferred. Appearance can be arbitrarily mapped onto an avatar, meaning that people are not required to dress appropriately, wear make-up, have neat hair, etc. They might value the option of doing other things whilst 'attending' the meeting (perhaps reading email, or playing games on their computer), safe in the knowledge that others don't know what they are really doing. However, a feature of real meetings is that there is social pressure to participate properly. Others will notice if someone isn't paying attention. Whilst there may be circumstances in which some degree of anonymity might be valuable, our work focuses on exploring a strong sense of social presence, and this requires that participants are willing to allow others to be closely aware of their actions.

3.2 Mapping from the real to the virtual

A virtual meeting involves participants in a real world setting (*not* the meeting), with information being captured and mapped to a virtual setting (*the* virtual meeting). The simplest model is that people sit at a desk, in front of a computer, quite likely their normal work scenario. This maps nicely to a virtual environment in which they are depicted as sitting at a conference table. Potentially, such a system provides good possibilities for measuring fine movement. We simply need a sensor on the computer to capture an image of the person (see Figure 1). The Kinect sensor is a good fit for this environment.

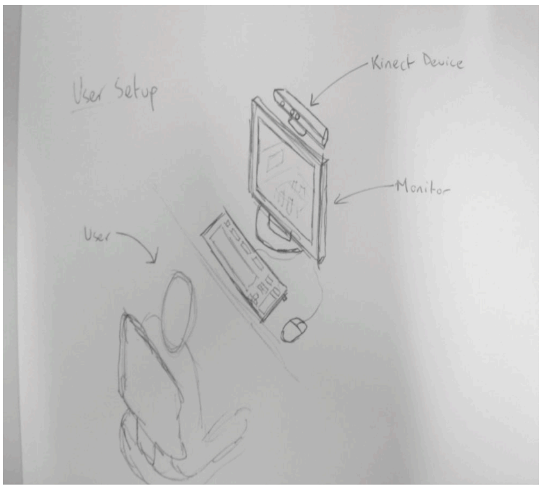


Figure 1: The participant in their real-world environment

However, with this scenario, there is little possibility for mapping gross movement; the participant cannot move in such a way as to naturally map to their avatar moving around the meeting room. In fact that need not be problem; people usually stay in their chairs during meetings. If they get up to get a drink, for example, they are effectively leaving the meeting temporarily. Perhaps only common exception to the fixed seating rule is that someone may move to the front in order to make a presentation.

A minor extension to our real world setting accommodates presentations as shown in Figure 2. As before, a participant normally sits in a chair in front of a computer. A sensor mounted on the computer monitors them, and they continue to view the virtual world on the computer display. In addition however, a large display screen (or a whiteboard) is arranged approximately 2 metres back from the computer, fully within the viewing angle of the sensor. The space is sufficient to allow the user to stand and present at the screen, and the sensor can capture a full body view allowing the user to step close to the screen and gesture. This modified arrangement is still suited to the capabilities of the Kinect device and still requires only a single sensor. The user can shift between presentation and sitting modes simply by moving. There are a number of options for mapping the movement to a presentation position. One possibility is to have a presentation screen behind each participant in the virtual world; another is to animate walking to the front of the room. In our experimental system, we have chosen to simply jump the avatar to a presentation position when the presenter moves to their real-world screen.

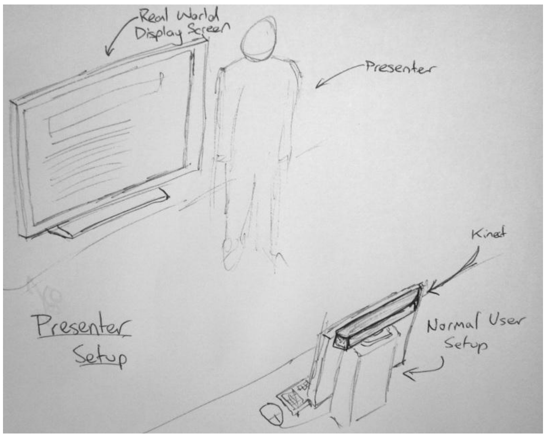


Figure 2: The modified real-world environment allowing for presentations by the participant

3.3 Avatar appearance

The second issue to be addressed is that of avatar appearance and facial expression. VMX avatars have been implemented as 'pipe' models based on the 'bones' deduced by the Kinect body position recognition system, with the head shown as a torus containing a live video image of the participant's face; Figure 3 shows an early implementation view. In a real meeting there is little interest in looking at the back of someone's head, so the video of the face was made one-sided; from behind the head is an empty torus, but the orientation makes it possible to work out the general direction in which the person is facing. When a person is facing away from the observer, their head does not completely block the view.

Using a live video image satisfies both of the requirements of participant identification and conveying facial expression.

An interesting question was the size of the avatar head (and video image). Figure 4 shows three different head sizes. While the smaller head gives a better view around the room, larger heads give better opportunity to observe facial expressions. VMX provides a keyboard control to allow the user to adjust head size. The default is a view that is roughly anatomically correct.

Figure 3: One participant mapped around the conference table. This is an early screen shot, before chairs were implemented. Note the arm gesture, the face, and the back-of-head transparency.

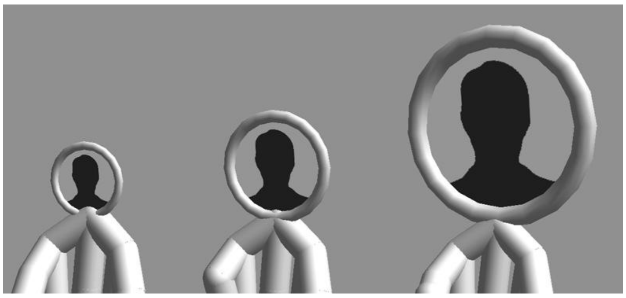


Figure 4: Effect of different avatar head sizes

A final detail in avatar display is that the avatar 'pipe' body is coloured by automatic selection of a colour from the user' clothing – from near the centre of their torso.

An avatar's position at the meeting table is determined at present by login order. Once the location is determined, the system maps orientation and position automatically. Small movements by the user in the real world are mapped to appropriate movements at the virtual table. However, in doing this mapping, a problem arises.

It makes sense that a participant might look around the table by turning their head. To look at the neighbour to the left, the virtual head would be turned 90 degrees to the left. Unfortunately, the participant's view of the virtual world is provided by an ordinary size computer monitor in front of the user. Turning by 90 degrees means that they are no longer facing the monitor, and therefore cannot see anything, although their avatar turns correctly for the intended view. The VMX system solves this problem by making avatar head orientation an amplified copy of body orientation. Users are sitting in ordinary office chairs, which permit swivel. A small turn of the chair (and hence the body) is mapped to a larger head movement on the avatar, which is in turn reflected by change in view orientation in the image of the virtual scene. This system allows a user to turn their avatar, and thus view, plus or minus 90 degrees while continuing to comfortably view their screen. User testing showed this to be a remarkably natural mechanism to which people adapted very quickly.

3.4 Presentations at the virtual meeting

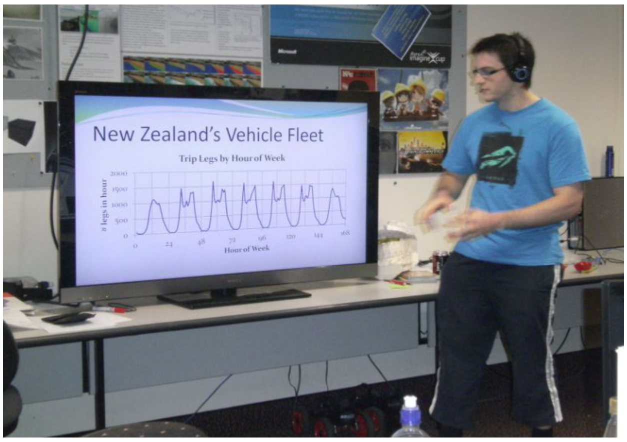
Presentation mode is based on a user standing in front of a large, but not huge, display (Figure 2). In our experimental setup, the presentation display in a participant's real-world environment is a 52 inch video display; large in a personal setting, similar to a medium sized office whiteboard, and therefore appropriate. It is a comfortable size for hand gestures, although gesturing may at times mean that the presenter is obscuring part of the display. It is however, not as big as a typical conference room display, and our experiments showed that the display in the virtual meeting room should be larger than life size.

There are two options for managing content on the display screen. One is to use video of the real display. That would nicely accommodate a whiteboard, but would lead to lighting, resolution and occlusion problems. The alternative is to use a digital image. This latter option was chosen. A PowerPoint slide show could be displayed on the real large screen, and independently mapped onto the virtual display screen. The result is a crisp image on both, with the virtual display larger than the real one, relative to the room and the avatars (see Figure 5).

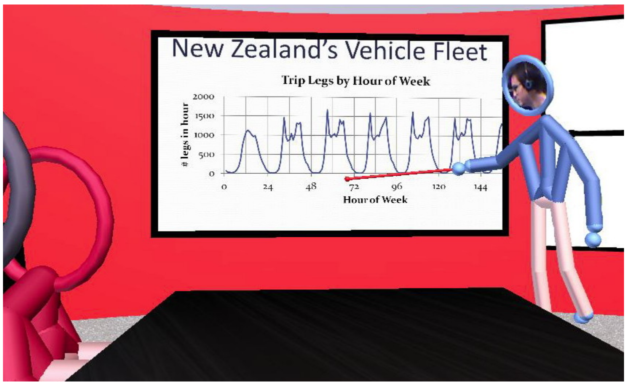
At this point, a new complexity arises. Being able to point to features of a displayed document was an important goal of the project. The scale remapping makes this difficult. The VMX implementation addresses the problem as follows. Video of the presenter's hand is tracked in the real world by the Kinect camera. If the user makes a pointing gesture, with their index finger, the system determines the point targeted on the real screen, and in the virtual world draws a virtual pointer from the avatar's hand to the corresponding point on the virtual display screen. Figure 5 illustrates this situation. (Note also the blue avatar body, matching the presenter's T-shirt colour.) Evaluation showed that this approach worked with acceptable accuracy.

A benefit of the screen mapping and virtual pointer system is that the software has some control over avatar placement, and if necessary could automatically ensure that the avatar didn't occlude any of the display, although this feature is not implemented in the current VMX system. Note also that the user doesn't need to hold anything. The pointer is generated whenever the user makes a pointing gesture and that gesture is in the direction of the screen.

The fact that the presenter's hands are free leaves open the possibility of further gesture usage. VMX implements gestures to scale, pan and page the display document, although testing has shown that these are not easy to use.



(a)



(b)

Figure 5: Mapping of a presenter and a presentation; (a) the real-world, and (b) the mapping into the virtual meeting room

3.5 Viewing control

VMX provides additional movement controls. It is possible to watch a presentation in seated mode, and this has the advantage of allowing a user to keep an eye on other participants. Figure 6 shows a presentation viewed from one participant's seat; three of the other participants are watching the presentation, while the fourth is distracted, and looking down at something in their real world space. The prototype implementation allows participants to set their viewpoint to a third-person perspective (Salamin, Thalman & Vexo, 2006) from elsewhere than their seated position. The head-of-table view of Figure 5(b) demonstrates a better-chosen viewpoint for the presentation.

An experimental feature of VMX is table reshaping to accommodate presentations, and to obviate the need for a third-person perspective. The concept is shown in Figure 7. There is no reason to keep table layout fixed; the format suited to presentation may be different from that suited to discussion. As the mapping from a participant's real world space to the virtual world is an artifice, it can be changed dynamically, with the only constraint that the change should be animated to avoid participant confusion.

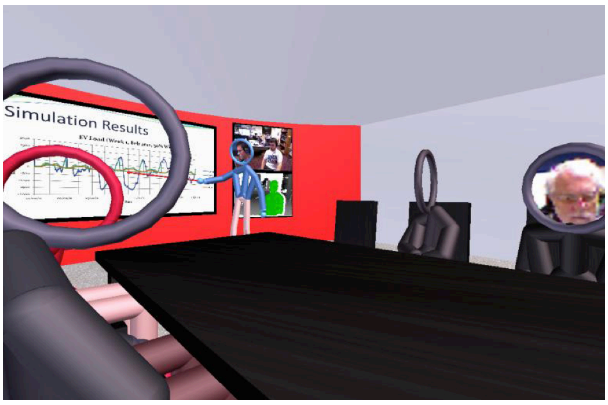
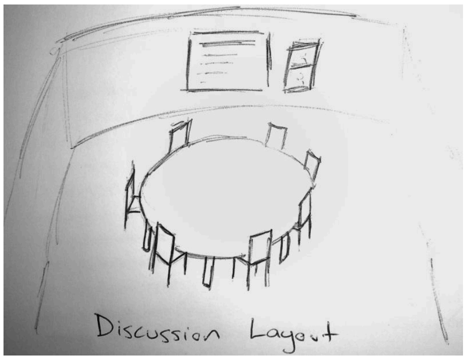
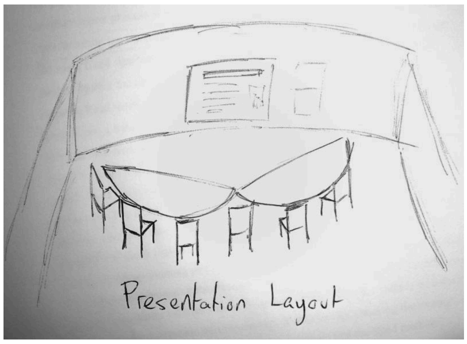


Figure 6: A presentation in progress, as viewed in first-person by a seated participant



(a)



(b)

Figure 7: The table can be morphed from a discussion layout (a) to one more suited to a presentation (b)

4. VMX implementation details and issues

The following brief comments relate to pertinent features and issues with the current VMX implementation.

(i) The prototype VMX system is implemented in C# using the XNA 3.1 graphics library and version 1.0 Kinect SDK. Networking is a client / server model, with clients streaming position and video to a server for distribution. Video is sent via UDP; position uses a TCP channel. The system is quite demanding of network bandwidth; careful setting of video resolution and size was necessary to get acceptable performance for 6 users in a local area network environment (only 35x35 pixels for facial features).

(ii) The Kinect system cannot provide bone locations for legs and feet when a user is seated, because the participant's legs are not in the sensor view. The

VMX system calculates and imposes a seated posture for these non-tracked bones.

(iii) Kinect does not resolve head direction; the head is just a single bone. VMX assumes that the head is oriented at right angles to the line between the shoulders. Shoulder orientation is thus taken as a proxy for head orientation.

(iv) Kinect does not track finger positions. To compensate for this lack, VMX implements an image analysis system to identify hand gestures, in particular the finger point gesture. The image analysis takes advantage of Kinect camera depth information to separate the participant from the background.

(v) Kinect also uses image analysis to identify the location of the large presentation screen, for the purpose of resolving the target of presenter pointing.

(vi) The VMX prototype does not incorporate audio facilities. For testing purposes we used an independent program – Teamspeak (www.TeamSpeak.com) – to provide the audio channel. The only disadvantage is that a user must establish audio connection independently of their meeting login. However, from an experimental development viewpoint, it was convenient to have a reliable system to use, and therefore not to have the evaluation results complicated by any issues that might have arisen with audio.

5. Testing and evaluation

In order to evaluate the VMS software a trial meeting was conducted with five active participants plus an observer, all present in the virtual meeting room, but physically located in separate spaces. The agenda for this meeting included a research presentation followed by a round-table discussion. Figures 5 and 6 show scenes recorded from this trial. All participants received brief instructions on the use of the VMX software at the start of the meeting. Following the meeting participants completed a questionnaire divided into five sections, covering the features of the VMX software, the experience of delivering a presentation, the experience of the meeting in general, how VMX compared with other forms of meeting, and other and general issues.

Points of note raised during the meeting and in the questionnaires can be summarised as follows:

- Participants reported having difficulty in seeing past the heads of others, in spite of the transparent back-of-head feature. Some overcame this problem by moving the camera to a third-person perspective, but none used the head-shrinking facility for this purpose.
- Participants reported some difficulty in discerning facial expressions, In part this was due to the lower than ideal resolution (35x35 pixels for faces) dictated by network bandwidth issues, and in the case of the presenter, exacerbated by their distance from the display screen.
- Most participants reported being able to recognise when someone was applauding, pointing, raising their hand to ask a question, and other gestures involving hand/arm movement, although at least one instance of a raised hand did go unnoticed by the presenter (this can happen in real life too).
- Some reported that body language was at times obscured by jitter in skeleton position data.
- In general, it was possible to detect in which direction participants were looking, and who was speaking at any given time. However, there were suggestions that a flag above the current speakers head would be a useful addition.
- There were some reports of activity outside of the meeting being observed, through head movement, body movement, and eye-gaze direction.
- There were some problems when the presenter unintentionally moved out of the camera frame, and the presentation view controls (scroll, zoom) required practice to be effectively used.
- At the suggestion of one participant, the presentation screen was used to record notes during the discussion phase of the meeting. Small modifications to the system could make this feature even more effective.
- Opinions on the relative merit of automatic (first-person) and manual (third-person) camera control were mixed, but there was general agreement that there was a place for both.
- When compared with other forms of remote meeting (video conference, teleconference, non-Kinect virtual world meeting) all participants reported that the VMX experience was as good or better. They found it more relaxed that a videoconference, and more immersive and engaging than other forms of remote meeting. None, however, suggested that the VMX meeting was a good as a face-to-face meeting.

In summary, we would suggest that this outcome from the initial evaluation of a prototype system is very positive, and shows significant promise for moving forward with the notion of meetings in virtual worlds.

6. Conclusion

6.1 Summary

This project set out to address a number of issues identified in existing virtual meeting software as described in the background discussion (Section 2). We believe that VMX successfully solves or at least considerably improves the situation in each case.

First, the issues surrounding fine grained avatar control. In VMX, user body movements are directly measured and replayed on the avatar. This successfully transmits explicit gestures, such as hand raising, in a way that requires no other action by a user except simply making the gesture. Because all movements are transmitted, there is a continuous sense of 'liveness'. The avatar orients wherever its user is facing. If the user thumps the table, mops their brow, or just wriggles in their seat, their avatar follows suit. Within the limitations of the avatar representation, reproduction of movement is faithful in timing and intensity. An immediate, enthusiastic raising of the hand is clearly different from a slow, incomplete gesture.

VMX displays live video of participants' faces on their avatars' faces (head tori). This personalises the avatar and provides the primary means of identification. The face is not always in view, so identification may have to wait until a person turns, but that is not too different from what might happen in a real world meeting. A secondary means of identification is provided by colouring the avatar body. This makes the avatars look different from one another, and serves as a memory aid to retain identification when the avatar is facing away. A second effect of face video is the transmission of non-verbal cues, such as facial expression and eye gaze. Eye gaze reproduction is not very meaningful in the left/right dimension. VMX's head orientation is a more useful indication of gaze target. However, the vertical dimension of eye gaze shows clearly when a person's attention is focused on the table surface of their real world, rather than the meeting. Overall, the combination of avatar body position mapping, and face video provides an excellent rendering of a participant's level of engagement with the meeting.

Presentation mode enables a participant to address the meeting, supported by a 'projected' document. VMX's extension to Kinect's tracking capability provides a virtual pointer so that audience attention can be directed to items on that document. Face video has the curious effect of allowing any speaker to achieve what only very good speakers can achieve in the real world. By looking at the Kinect sensor, a presenter can appear to be looking directly at each and every audience member.

Finally, navigation. VMX does not provide large scale navigation because there seems to be no reason to simulate movement that is at best marginally relevant to the task of attending a meeting. In VMX, a person is at the meeting table, or giving a presentation, and they 'teleport' between the two states. Indeed, this 'limitation' provides a practical advantage. There is not the opportunity offered in the real world to stumble, drop papers, or knock over chairs. Other simulated circumstances might dictate a different approach. For example, at a virtual cocktail party, participants should be allowed to circulate around discussion groups.

VMX implements an innovative feature in small scale navigation. Using body rotation to orient a person's avatar might lead to them no longer facing their real world screen. To avoid this, the system amplifies their angle of rotation. Whilst this may seem unnatural, participants adapted to the effect very easily, and after a short time found that they were looking around the virtual scene without conscious effort – in a swivel office chair, this could be described as steering by the seat of the pants.

6.2 Further work

The VMX system described is a prototype, and is under on-going development. A range of refinements based on experience so far are being considered, including further gesture use for presentation control and the meeting table manipulation already described.

The possibility of extending VMX to a large conference or performance setting is a greater challenge, given the communications limitations experienced with the small-scale prototype. However, a possible solution would be to implement local groups in the audience, allowing people to attend in small groups (similar in size to those that VMX currently supports), and within these groups having participants communicate directly with one another. Providing full audience feedback to the presenter could be achieved by an aggregation tree of video feeds, merging into a single video image stream of the audience. A similar aggregation of audio could allow audience members a general impression of overall sound – allowing an accurate reproduction of applause, for example.

7. References

Al Qahtani, S.H. (2010): Virtual worlds as meeting places. Unpublished COMP591 Honours Dissertation, Department of Computer Science, University of Waikato.

Arnfalk, P. and Kogg, B. (2003): Service transformation—managing a shift from business travel to virtual meetings. *Journal of Cleaner Production*, **11**(8), 859-872.

Bekkering, E. & Shim, J.P. (2006): Trust in videoconferencing. *Commun. ACM,* **49**(7), 103-107.

Chapanis, A., Ochsman, R.B., Parrish, R.N. and Weeks, G.D. (1972): Studies in interactive communication: I. The effects of four communication modes on the behavior of teams during cooperative problem solving. *Human Factors*, **14(**6), 487-509.

Dean, Jesse (2012): Using motion controllers in virtual conferencing. MSc Thesis, University of Waikato.

DeVincenzi, A., Lining Yao, Ishii, H. & Raskar, R. (2011): Kinected conference: augmenting video imaging with calibrated depth and audio. *Proceedings of the ACM 2011 conference on Computer supported cooperative work* (CSCW '11): 621-624, ACM, New York.

Egido, C. (1988): Video conferencing as a technology to support group work: a review of its failures. *Proceedings of the 1988 ACM conference on Computer-supported cooperative work* (CSCW '88): 13-24, ACM, New York.

Erickson, T., Shami, N.S., Kellogg, W.A. and Levine, D.W. (2011): Synchronous interaction among hundreds: an evaluation of a conference in an avatar-based virtual environment. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11): 503-512, ACM, New York.

Kinect: *Kinect Sensor*. Retrieved 19 August, 2013, http://msdn.microsoft.com/en-us/library/hh438998.aspx

Lindeman, R.W., Reiners, D. and Steed, A. (2009): Practicing what we preach: IEEE VR 2009 virtual program committee meeting. *IEEE Computer Graphics and Applications*, **29**(2), 80-83.

Mackay, D.J.C. (2008): *Sustainable energy – without the hot air*. UIT Cambridge. ISBN 978-0-9544529-3-3. Available free online from www.withouthotair.com.

Purdey, J.M. & Nye, P. (2000): The impact of communication media on negotiation outcomes. *Intl. J. of Conflict Management*. **11**(2), 162-187.

Salamin, P., Thalmann, D. & Vexo, F. (2006): The benefits of third-person perspective in virtual and augmented reality?. In *Proceedings of the ACM symposium on Virtual reality software and technology* (VRST '06): 27-30, ACM, New York.

Schuemie, M.J., van der Straaten, P., Krijn, M., & van der Mast, C.A.P.G. (2001): Research on presence in virtual reality: A survey. *CyberPsychology & Behavior*, **4**(2), 183-201.

Steuer, J. (1992): Defining virtual reality: Dimensions determining telepresence. *Journal of Commnication* **42(**4),73-93.

Taylor, T.L. (2002): Living digitally: Embodiment in virtual worlds. In R. Schroeder (Ed.) *The Social Life of Avatars: Presence and Interaction in Shared Virtual Environments*. Springer-Verlag, New York.

Yong Li Lan (2003). Shakespeare as Virtual Event. *Theatre Research International*, **1**, 46-60. doi:10.1017/S0307883303000142.

Involving Geographically Distributed Users in the Design of an Interactive System

Saturnino Luz[1] **Masood Masoodian**[2]

[1] School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
Email: luzs@cs.tcd.ie

[2] Department of Computer Science
The University of Waikato
Hamilton, New Zealand
Email: masood@cs.waikato.ac.nz

Abstract

We report the process of designing an interactive system for use in disease surveillance and patient case management at remote communities in the Amazon region. The design aimed at coordinating and supporting the tasks of a distributed community of users in the region, and involved collaboration of designers and developers located in three continents, working on a common research project. In spite of this high degree of geographical distribution, both of users and designers, a participatory design approach was pursued. The challenges faced by this project as regards securing end-user involvement, eliciting requirements, generating and validating design ideas, and iterating the design process are discussed. A prototype which embodies the experience gained in applying these methods in this particular setting is also described.

Keywords: Mobile devices, disease surveillance, interactive geographical maps, medical information systems.

1 Introduction

Software engineering projects are increasingly being conducted by geographically distributed teams (Herbsleb, 2007; Cusumano, 2008). The challenges faced by such teams have been documented in a number of studies (Damian and Zowghi, 2003; Ramesh et al., 2006) which highlight issues such as how to best manage collaboration among team members of different cultural backgrounds, how to minimise communication problems, how to improve the technical infrastructure for cooperative work, and how to implement effective knowledge and information management mechanisms to support the work of the development teams and project leaders (Herbsleb, 2007).

Most of these studies, however, focus on large scale projects, often in the context of cross-border organisations, where the distribution of development is primarily driven by business considerations, such as the desire to exploit local market opportunities, the "need to capitalise on the global resource pool" (Herbsleb, 2007) and even the need to deal with time pressures by exploiting time-zone differences. Far less attention has been devoted to HCI issues in the distributed development of interactive systems. Such issues are typical of, for instance, research projects, where development is mainly motivated by scientific or applied research questions and spans multiple institutions. In these contexts, user-centred and participatory design methodologies are often preferred, placing additional strains on activities such as requirements elicitation which have proved challenging for distributed teams, even those working with more traditional development methodologies (Damian and Zowghi, 2003).

This paper reports the case of one such distributed project which has focused on exploring the potential of interactive mobile systems to assist local healthcare providers and epidemiology researchers in treating patients and monitoring emerging and neglected infectious disease in the Amazon region. We report our experience and the methods we employed in designing a system for this geographically dispersed user community while following a participatory design philosophy. This process has involved several iterations of requirements gathering, design, development and user evaluation.

2 Disease surveillance and Healthcare in remote regions

Management of healthcare in remote regions to serve populations located in large, sparsely populated extensions is acknowledged as a difficult problem (Dussault and Franceschini, 2006). In the Amazon region, this problem is compounded by the natural geographical obstacles, the consequent scarcity of human resources, and more recently, the phenomenon of global climate change. Climate change has altered the pattern of land use and cover in the region and it is expected to continue to do so (Cesario et al., 2011). These factors are contributing to the spread of diseases, such as American Cutaneous Leishmaniasis, and increasing the likelihood that diseases not yet found in the Amazon region, such as Bartonellosis, will reach the region. This situation creates requirements that are not addressed in conventional telemedicine systems, which focus mainly on issues of remote assistance to diagnosis and treatment (Hailey et al., 2002).

Health management in this context requires a more comprehensive approach capable of dealing with issues of human ecology and disease surveillance, in addition to patient care (Cesario et al., 2011). Cooperative work needs to be flexible and take localised diagnosis, data collection and data analysis tasks into account. Therefore the support of interactive computer systems needs to be provided at different levels in aid of diverse but inter-related activities (Luz et al., 2013, 2012).

At one level, primary care activities are often carried out in remote locations by nurses and community healthcare workers. The latter have generally little training and would therefore benefit from support, specially under circumstances involving the identification of diseases that are rare in the areas in which they operate. Communication with specialists and researchers might help improve primary care providers' performance.

At a higher level, the healthcare system needs to collect and maintain patient records which will be accessed, among others, by epidemiological surveillance bodies which have the need to gather accurate information on disease occurrence, including patient location, along with a number of other relevant environmental variables.

In the case of the project reported here, these collaborations can also cross country borders. One of the diseases targeted by our project is Bartonellosis, which is endemic in some areas of Peru, has spread in recent years, and risks reaching the disease-free Peru-Bolivia-Brazil tri-national borders (Cesario et al., 2011).

Within these complex practical work settings, we have explored a number of technological facets as possible ways of supporting the activities of healthcare professionals, researchers, policy makers, and administrators. These include:

- mobile support for epidemiological research and data collection,
- mobile assistance to disease identification and case notification,
- visualisation of cases and distributed activities reported to a central database, and
- support for synchronous and asynchronous communication among healthcare providers.

3 Geographically distributed design

Participatory design literature often cautions designers and developers of technology to be aware of their differences with potential users, who according to the principles of participatory design must be actively involved in the design process. Specific techniques, such as envisionment videos (Buxton, 2007), have therefore been developed to facilitate this process (Masoodian, 2000).

Further complexity arises when the design and development team itself consists of members from different cultural, scientific, and linguistic backgrounds. Due to the nature of contemporary research, where international and inter-disciplinary collaboration is encouraged, most projects tend to encompass such diverse backgrounds. This is of course the case of our project.

Given the characteristics of the problem described in Section 2, the proposed solution we have been investigating, and the activities it needs to support, our team encompasses researchers from two main disciplines, namely, medicine and computer science. However, within the discipline of medicine itself, our team members have different focuses and educational backgrounds. They include: a professor of medicine, a medical researcher and an epidemiologist.

In addition to diverse backgrounds, the team has had to cope with working and cooperating across geographical distance. The design and software engineering teams are located mainly in Ireland and New Zealand, while the medical specialists who act as co-developers and facilitators of end user participation are located in research institutes in Brazil. The distances make it impractical to hold frequent face-to-face meetings. Collaboration has therefore required not only the regular use of communication tools such as email and (audio and video) conferencing software, but also the use of participatory design artefacts (e.g. sketch interfaces and envisionment videos) that could be exchanged for asynchronous use to support requirement analysis and design ideation.

At the initial stages of the design process, sketches and simple storyboards, which could be easily scanned and distributed among the groups and used as support material for presentation, proved quite effective at helping establish common ground among the different specialist areas and sensitise the designers and engineering team to potential situations of concern. As a shared understanding developed, more elaborate tools were introduced. A valuable experience in this context has been the production of the envisionment video itself, as a design sketching tool, involving all the team members. As part of this process our team worked together remotely to sketch the initial storyboards, develop the script, film, and then iteratively edit and produce the final video.

A similar process has been followed for the design and development of our software prototype (discussed later). This has involved dividing the system into different functional components (patient case data collection, computer-assisted diagnostic, and visualisation), each of which have been designed and developed iteratively through the involvement of interaction designers, software engineers and specific medical expert team members.

4 Geographically distributed users

Early on in the design process it became clear that supporting a complex activity such as disease surveillance would necessarily mean supporting the tasks of local healthcare providers and mobile researchers who act as gatherers and users of epidemiology data. This posed the initial challenge of identifying the main actors, their background knowledge and their work patterns.

As mentioned before, the geographical area covered by the prospective users of the system spans a region of the Brazilian Amazon that borders two other South American countries: Peru and Bolivia. Within these countries, the project stakeholders are located in places that are themselves separated by great distances (Figure 1). In particular, the focus of data collection is within a sparsely populated rural area whose healthcare needs are served by community healthcare providers and nurses.

4.1 Overall Strategy

Due to the complex and distributed nature of healthcare and disease surveillance in the region, we employed observational and qualitative methods to help the team reach common ground as to the context of the activities that might be supported by technology, the main actors involved and their typical tasks.

Once a shared overall picture of the context of the project was formed, the team sought to secure the

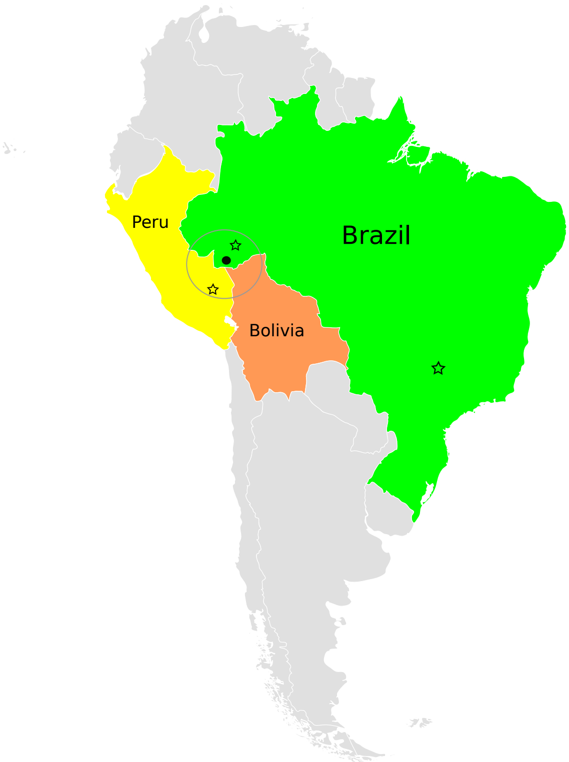


Figure 1: Map of South America showing the areas over which the user group is distributed. The larger circle indicates the region where patient data are collected. The stars indicate the locations of project participants and the filled circle the municipality where fieldwork has been conducted.

involvement of the prospective user group by recruiting *informants* who acted as the main contact points between the developers and the users. These informants are researchers who have conducted investigations on different aspects of disease surveillance, prevention and treatment in the region, independently of this project, but have an interest in tools and technology that might improve such tasks. These informants were briefed on the nature of the project and the possibilities of technology support. They contributed to the refinement of use cases, provided feedback on the sketched interfaces, and participated in interviews.

As the project evolved, key informants were kept abreast of the developments and employed the above mentioned techniques in their cooperation with the international members of the design team, including low-fidelity prototyping and envisionment videos. They also employed these tools to test design ideas with the local user groups. We also gathered feedback across a larger user population by means of questionnaires. These key researchers have also played an important role in the design of a high-fidelity prototype which currently serves as a basis for further development iterations.

In the following sections we describe these methods and their outcomes in further detail.

4.2 Fieldwork: main actors and their tasks

The fieldwork has been conducted by two members of the project team over an extended period. They have medical and epidemiology backgrounds and lived in the Amazon region for the duration of the initial fieldwork. Contacts with local health workers were established and information exchanges with medical experts from Peru and Bolivia were initiated. This qualitative research produced a comprehensive document which outlined the disease surveillance and healthcare situation from a general perspective, and documented the work practices of the prospective users in great detail. This document was circulated to the international partners, along with the relevant medical and healthcare policy literature. The design team then structured these data in the light of existing analytic frameworks, paying special attention to issues of mobility in healthcare contexts (Doherty et al., 2010; Bardram and Bossen, 2005), in order to identify main actors and plausible design statements, and devise a strategy for requirements analysis. This qualitative analysis was complemented by surveys and focused investigations of specific issues, along the lines of the methodology proposed by Kane and Luz (2011) in the context of information sharing among specialists from different disciplines.

Fieldwork revealed that the various actors involved in healthcare provision within the groups represented in this project perform different but complementary sets of activities and make use of specialised but interrelated knowledge and representations. Our efforts therefore focused on supporting the articulation work (Schmidt and Bannon, 1992) necessary in order to improve the activities of the different stakeholders though effective information sharing and use.

Closest to the patients, and performing most of the primary care in the region are *community health workers*, *local doctors* and *nurses*. Community health workers are members of the community who have no tertiary (university) training in healthcare but nevertheless perform essential functions related to healthcare delivery, receiving basic training in the context of specific interventions. They also perform other developmental and promotional roles, acting as bridges between the community and formal health services (Lehmann and Sanders, 2007). Their activities are complemented and assisted by the activities of trained nurses, medical assistants and doctors based in local health centres. Due to geography of the region, and also to the nature of their job, it is in the work of community health workers that mobility requirements are most apparent. This level of mobility and the lower levels of medical training of these workers suggest a need to support coordination between (mobile) community healthcare workers and nurse, their more direct but less mobile counterparts in primary care. This type of asymmetry (of mobility, training and educational level) observed between community health workers and nurses is even more pronounced between the former and medical specialists and researchers.

At the levels of secondary and tertiary care, one finds *medical specialists* who treat patients referred to them by the local health centres. In addition to treating patients, doctors often provide guidance to the primary care agents and engage in research activities. While this type of guidance is rarely formalised, reflecting legal and institutional constraints, it relies nevertheless on certain coordination mechanisms (Schmidt and Simone, 1996) which could, in principle, be reinforced through technology. Doctors from regions where a disease is newly emerging will typically seek information from colleagues from regions where the disease is endemic.

Two types of actors use the patient data gathered by healthcare providers: *healthcare service managers* and *epidemiology researchers*. The former are concerned with prevention policies, tools and strategies. The latter are concerned with modelling and assessment of risks for the purpose of disease surveillance. In this particular setting, the epidemiologist members of the team aim to aggregate data from several sources, including climate, land use, disease vector distribution and terrain features, as well as the medical data pertaining to the spread of cases, both geographically and temporally.

Through the identification of these potential user groups it became clear that they have different but somewhat interrelated needs for interactivity and data. Healthcare workers working in the field need mobility, access to local data (maps, residences visited etc) and, occasionally, assess to specialist assistance. Epidemiology researchers and managers need global, compiled data. The following use cases illustrate the nature of these needs.

4.3 Use cases

In the following sections we outline potential use cases which emerged from the initial observational studies. These use cases have been further investigated through the prototyping activities reported in Section 4.4.

4.3.1 Diagnostic assistance

Due to their limited resources and medical training, local health workers can benefit from assistance in identifying conditions and triaging patients. Since physical distance and mobility constraints make carrying of bulky laboratory equipment and instructional material impractical, such assistance could take the form of electronically stored medical atlases for identification of the cutaneous lesions that characterise diseases such as Bartonellosis and Leishmaniasis, and support for synchronous or asynchronous communication with experts. A typical scenario would involve the use of a mobile device to photograph the lesion, attempt automatic identification against a precompiled statistical model, fallback to identification assisted by image (medical atlas) comparison in case of uncertainty, and transmission of image and patient data to a specialist centre for further investigation.

4.3.2 Data collection and epidemiological surveillance

In the context of diagnostic and treatment of emerging diseases and epidemiological surveillance these primary care workers could also play an important role in recording details of suspected cases for notification, identification of disease vectors (carriers) which in the case of Bartonellosis and Leishmaniasis are sandflies, and educate the local communities. Other stakeholders with a local presence, such as field researchers, may also contribute to gathering different facets of epidemiologically relevant data from visits to local families. This includes accurate location data (GPS coordinates), data about time spent outdoors by family members during sandfly feeding times, insect bite prevention measures, history of insect bites or infestation and places where insect bites occurred, number and type of domestic and peri-domestic animals (known reservoirs of the Leishmaniasis parasite) at the home, outdoor occupational and recreational activities, travel and household hygienic facilities. These data can be integrated into early warning systems and shared among fieldworkers.

4.3.3 Mapping, visualisation and coordination tasks

Information collected by researchers would help in coordination tasks such as mapping of at-risk families to be visited by a healthcare provider, a record of households visited, statistics regarding the effectiveness of specific interventions etc. As these data have spacial and temporal facets, covering a number of different categorical and numerical variables, geographical visualisation can potentially play an important role in supporting awareness and coordination of activities among the various stake holders.

4.4 Prototyping

Our analysis of the use cases discussed above identified four categories of tasks that could be supported using technology. As mentioned earlier, these related broadly to patient case data collection, disease identification, visualisation of cases, and communication between healthcare providers.

The requirements of each of these categories of tasks in terms of their scope and the type of functionality that needs to be supported by technology have changed over time as we have worked with our users in refining them iteratively. The initial design iterations involved sharing of hand-drawn sketches (Figure 2) and storyboards (Figure 3) illustrating tentative functions identified in the observational studies and fieldwork. These sketches and storyboards formed the basis for four initial meetings and various email exchanges encompassing healthcare professionals, system developers and other stakeholders. They were essential in helping establish an initial common ground and in giving the prospective users of *nu-case* an idea of what was possible in a fairly unconstrained manner.

These initial low-level prototypes were then refined. The consensus that was emerging on the range of possible and potentially desirable features of the system was incorporated into a concept video which showed scenes of the prototype then under development in use in a series of scenarios that illustrated the above described use cases[1]. Frames of this concept video are shown in Figure 4. The video contained short verbal descriptions (in the form of captioned voice-overs) of the scenarios. It was presented to a diverse group of stakeholders, including medical doctors, healthcare service managers, nurses, community health workers, medical researchers and epidemiologists. Following the presentation, participants were given questionnaires designed to elicit their opinions on the usefulness of the functionality described, on potential risks, on the perceived usability of the system, on the suitability of the devices for the region and tasks, and on the potential utility of the system's functions to different healthcare providers. A detailed analysis of the feedback gathered in this phase of the project is presented in (Luz et al., 2013). In addition, the concept video has been used in a separate activity designed as part of the training of medical students in health informatics (Carloni et al., 2013).

From a development perspective, as the design and development team itself is geographically distributed, it has been necessary to develop each component of the prototype individually by different members of the team over time, while making sure that those components are integrated effectively. This has required careful selection of the development platform, data structures, map format etc.

[1] The full video is available at http://bit.ly/YFpAun

Figure 2: Parts of the initial sketches, illustrating patient case data collection using *nu-case*.

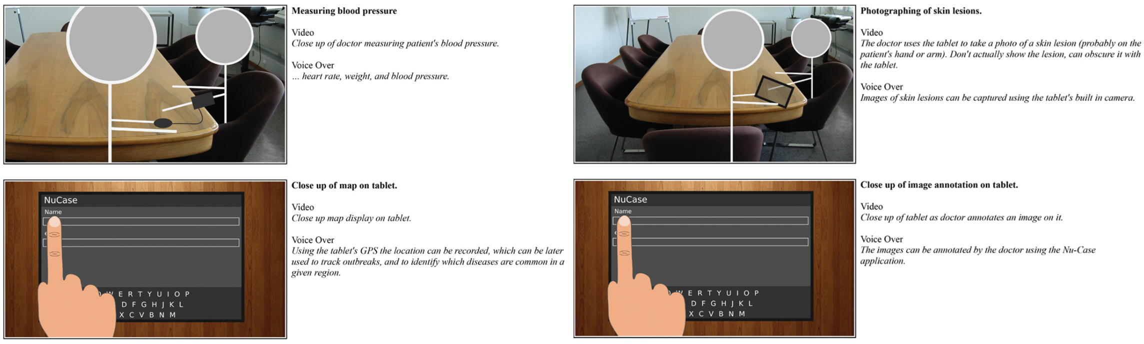


Figure 3: Parts of the storyboards, illustrating patient case data collection using *nu-case*.

User feedback was gathered at each prototyping iteration. The initial iterations focused on key actors and co-developers, while higher fidelity prototyping was used to elicit feedback from a broader user group.

4.4.1 Patient case data collection

Initially the type of functionality we had envisaged for this component was to allow collection of simple patient data, generally in text format, which would be used mainly for record-keeping purposes. Discussions with our users, and medical science team members, however, indicated that there was a need to expand the functionality supported by this component to include a range of data such as audio notes, precise GPS coordinates, photographs of patients and their skin lesions, images of slides for laboratory tests (e.g. blood smears) and other image-based diagnostic exams (e.g. X-rays).

These requirements are motivated by the fact that collection of accurate patient data is not currently carried out by our users working in the remote region, and such data is much needed not only for better diagnostic support, but also for notification of cases, and more effective visualisation and monitoring of the spread of neglected and emerging diseases.

Clearly mobile devices, with their extensive data collection capabilities (audio, video, photographs, GPS), are well suited for this purpose. We have therefore developed our prototype for Android™ smartphones and tablets[2]. Figure 5 shows a screenshot of part of the data collection component of our mobile system, called *nu-case*.

[2]http://developer.android.com/

4.4.2 Disease identification and notification

Providing diagnostic assistance to primary healthcare providers was among the first functions that we considered prototyping. The initial idea was to allow the user to attach a portable microscope to the mobile device and use it to take images of blood smears, which could then be automatically compared with blood smear samples of specific diseases (e.g. Bartonellosis and American Cutaneous Leishmaniasis) by the system to provide diagnostic advice. We therefore developed the necessary image analysis and machine learning functionality to support this process (Cesario et al., 2012). Comparison of blood smear images with malaria cases, as acute Bartonellosis can sometimes be misdiagnosed as malaria.

Despite some success with this work however, during the second iteration of prototyping, medical users suggested that a more useful approach would be to allow the user to make visual comparison of skin lesions that characterise diseases such as Bartonellosis and Leishmaniasis with photographs of existing cases using the system. This would have the advantage of resembling an accepted practice (consultation of medical atlases) while adapting it to the mobility requirements. The system could also be improved if the user could add photographs of skin lesions collected in the field during diagnosis to the image database and aggregated on a central server for curation by specialist and eventual use for future diagnostic assistance.

4.4.3 Visualisation of cases

Visualisation of geographical disease data using maps is an important component of epidemiology. Therefore, support for geovisualisation of patient case data was considered to be an essential requirement.

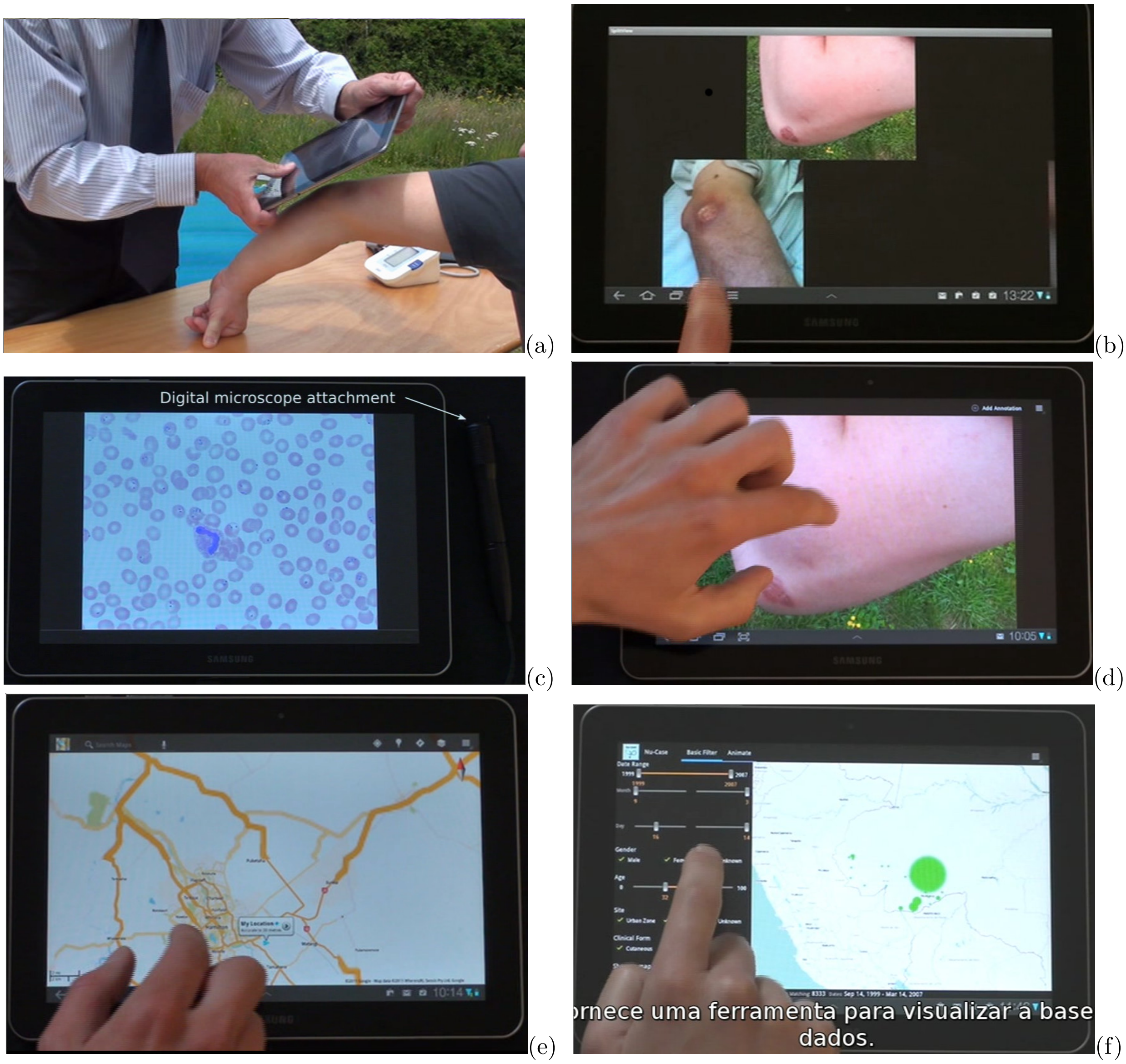


Figure 4: Frames of the video produced as part of the *nu-case* prototyping activities: (a) a doctor taking an image of a skin lesion for the electronic medical record, (b) support for image-assisted differential diagnosis, (c) blood smear image collection and machine-assisted analysis, (d) annotation of collected images for the medical record, (e) coordination of data collection through geographical information, and (f) visualisation of epidemiological data.

Since initially we did not have any data collected by the prototype system itself, we focused on the use of historical data made available to the project by the medical science team members. These data, collected over a decade, did not provide precise GPS location of reported cases, but had information about the municipality of infection and the municipality of notification for each case. We therefore developed a spatio-temporal visualisation of quantitative case data over the map of the region of interest. The evaluation of this visualisation in turn led to several improvements to the prototype.

We are currently working on the next version of the visualisation component which utilises more comprehensive patient case data newly collected in the region by a collaborating field researchers, based in the Amazon region. This data set not only includes the precise GPS data, but also other details such the patient's occupation, place of employment, type of employment, type of dwelling, family composition, proximity to rivers, presence of domestic animals etc. These data requirements were elicited through use of the first version of the visualisation prototype in user workshops. Figure 6 provides an example of the visualisation component of our *nu-case* prototype, showing several cases from the new data set.

4.4.4 Communication among care providers

The prototype system includes support for synchronous and asynchronous communication modes. One form of communication envisaged is between local healthcare providers and medical specialists to provide diagnostic assistance. This communication could be synchronous, and involve sharing of medical images and other data when reliable internet access is available, which may not be the case when working in the field. Alternatively, the communication could be asynchronous, and involve gathering of patient case data when there is no network connection, and uploading them to a central data repository later, as network access is restored (for instance when the

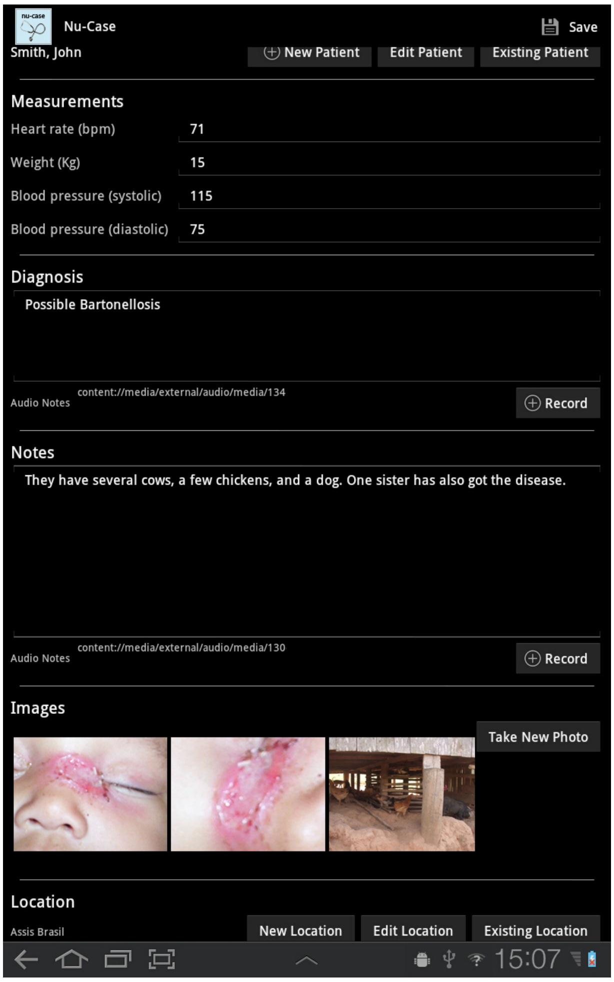


Figure 5: An example of a patient case in the data collection component of *nu-case*, which includes textual data, audio annotations, photographs, and GPS location data.

healthcare provider returns to their clinic).

Over several iterations, based on the feedback provided by our users and medical team members, our system has evolved to included a range of features to support more asynchronous communication, particularly for diagnostic assistance.

The system also aims to provide support for instance for downloading instructional and reference material (e.g. updated lesion images annotated by specialists, as shown in Figure 7) and the latest data on relevant geographical changes, disease propagation patterns etc.

5 Conclusion

In this paper we have described the design of an interactive system for assisting healthcare providers in remote regions with the task of diagnosis and monitoring of neglected and emerging infectious diseases. A distinguishing feature of this design process is the fact that it has been conducted by an interdisciplinary team of geographically distributed researchers and practitioners, for a group of users who are themselves geographically distributed.

Our experience has shown that although projects such as this are often complex in their nature, they can benefit from methodologies such as participatory design and related techniques. It has also shown that iterative prototyping can be successfully employed in

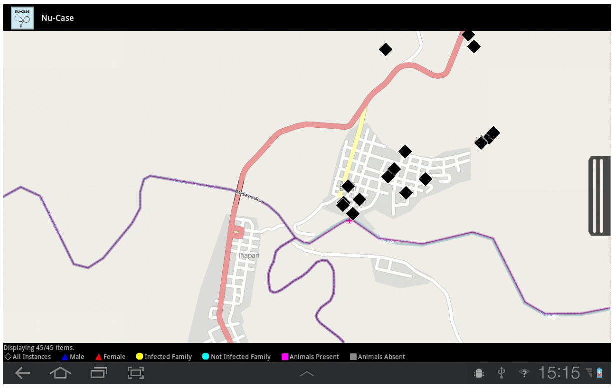


Figure 6: Visualisation component of *nu-case*, showing several individual cases on the map of the region of interest. Shown cases are selected by the user based on a number of case data attributes.

Figure 7: An example of the image annotation facility provided by *nu-case*, showing a lesion area on a photograph annotated by a specialist.

distributed development of interactive systems, provided that there is sufficient involvement of local informants and co-developers.

Our project is an ongoing one, and we aim to run field-trials of the *nu-case* prototype system with close involvement of our user group, once further development and testing of the various components of the system have been carried out.

Acknowledgements

The development of *nu-case* has been supported through several grants from the University of Waikato Summer Research Student Scholarships programme. Saturnino Luz is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College Dublin. We would also like to thank our collaborators in Brazil, without whom the design and development activities reported in this paper would not have been possible.

References

J.E. Bardram and C. Bossen. Mobility work: The spatial dimension of collaboration at a hospital. *Computer Supported Cooperative Work (CSCW)*, 14(2): 131–160, 2005.

B. Buxton. *Sketching User Experiences: Getting the*

Design Right and the Right Design. Morgan Kaufman, 2007.

M. B. Carloni, M. Cesario, R. Cesario, S. Luz, E. Margarido, M. Masoodian, and D. F. Pires. In-class use of the nu-case mobile telehealth system in a medical school. In *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on*, pages 437–440, 2013. doi: 10.1109/CBMS.2013.6627835.

M. Cesario, R. R. Cesario, and M. Andrade-Morraye. Environmental change and health impacts in amazonia. *Human Health and Global Environmental Change*, 1:26–33, 2011.

M. Cesario, M. Lundon, S. Luz, M. Masoodian, and B. Rogers. Mobile support for diagnosis of communicable diseases in remote locations. In *Proceedings of the 13th International Conference of the NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction*, CHINZ '12, pages 25–28, New York, NY, USA, 2012. ACM. doi: 10.1145/2379256.2379261.

M. A. Cusumano. Managing software development in globally distributed teams. *Communications of the ACM*, 51:15–17, 2008.

D. E. Damian and D. Zowghi. RE challenges in multi-site software development organisations. *Requirements engineering*, 8(3):149–160, 2003.

G. Doherty, J. McKnight, and S. Luz. Fieldwork for requirements: Frameworks for mobile healthcare applications. *International Journal of Human-Computer Studies*, 68(10):760–776, 2010. doi: 10.1016/j.ijhcs.2010.06.005.

G. Dussault and M. C. Franceschini. Not enough there, too many here: understanding geographical imbalances in the distribution of the health workforce. *Human Resources for Health*, 4(1):12, May 2006. ISSN 1478-4491. doi: 10.1186/1478-4491-4-12. URL `http://www.biomedcentral.com/1478-4491/4/12`.

D. Hailey, R. Roine, and A. Ohinmaa. Systematic review of evidence for the benefits of telemedicine. *Journal of Telemedicine and Telecare*, 8(suppl 1): 17, 2002.

J. D. Herbsleb. Global software engineering: The future of socio-technical coordination. In *2007 Future of Software Engineering*, pages 188–198, 2007.

B. Kane and S. Luz. Information sharing at multidisciplinary medical team meetings. *Group Decision and Negotiation*, 20:437–464, 2011. doi: 10.1007/s10726-009-9175-9.

U. Lehmann and D. Sanders. *Community health workers: What do we know about them? The state of the evidence on programmes, activities, costs and impact on health outcomes of using community health workers*. Geneva, 2007.

S. Luz, M. Masoodian, M. Cesario, and B. Rogers. Supporting collaboration among healthcare professionals and disease surveillance in remote areas. In *Proceedings of the 25th International Symposium on Computer-Based Medical Systems*, CBMS '12, pages 1–6, Rome, 2012. IEEE Press. doi: 10.1109/CBMS.2012.6266333.

S. Luz, M. Masoodian, and M. Cesario. Disease surveillance and patient care in remote regions: an exploratory study of collaboration among healthcare professionals in amazonia. *Behaviour & Information Technology*, 2013. doi: 10.1080/0144929X.2013.853836.

M. Masoodian. Facilitating communication during user-centred design. In *Proceedings of the Workshop on How to Make User-Centred Design Usable*, pages 45–50. TRITA-NA-D0006, CID-72, KTH, 2000.

B. Ramesh, L. Cao, K. Mohan, and P. Xu. Can distributed software development be agile? *Communications of the ACM*, 49(10):41–46, 2006.

K. Schmidt and L. Bannon. Taking CSCW seriously. *Computer Supported Cooperative Work*, 1(1-2):7–40, 1992. doi: 10.1007/BF00752449.

K. Schmidt and C. Simone. Coordination mechanisms: Towards a conceptual foundation of cscw systems design. *Computer Supported Cooperative Work (CSCW)*, 5(2):155–200, 1996.

Contributed Posters

Depth Perception in View-Dependent Near-Field Spatial AR

Markus Broecker[1] **Ross T. Smith**[1] **Bruce H. Thomas**[1]

[1] Wearable Computer Lab
School of Information Technology & Mathematical Sciences
University of South Australia
Email: markus.broecker@unisa.edu.au
ross.smith@unisa.edu.au
bruce.thomas@unisa.edu.au

Abstract

View-dependent rendering techniques are an important tool in Spatial Augmented Reality. These allow the addition of more detail and the depiction of purely virtual geometry inside the shape of physical props. This paper investigates the impact of different depth cues onto the depth perception of users.

Keywords: Augmented Reality, Spatial AR, Projector-based Rendering, Depth Perception

1 Introduction

Spatial Augmented Reality (SAR) provides a very direct form of mixed reality by augmenting physical models of objects using projected light (Raskar et al. 2001). This approach offers therefore more affordances than other types of augmented reality: the user is able to interact with physically existing models and there is a sense of 'presence' usually lacking in see-through displays (Bennett & Stevens 2005).

SAR operates in the following way: physical models, which act as projection surface, have a 3D geometric representation and each projector has a corresponding virtual camera. Projectors are aligned to geometry using pose-estimation techniques. After alignment, projectors augment the physical models by projecting the rendered view from their respective camera.

2 View-Dependent Rendering

View-dependent rendering provides perspective-correct rendering in a SAR environment from a single user's perspective. It is used to provide additional detail to coarse physical models (Menk et al. 2011) or to create purely virtual geometry and space (see Figures 2 or 1). As there is no intrinsically known 'central camera position', the user's position has to be tracked.

Creating these virtual geometries is a two-step process:

1. The user's position is known from tracking. The view of the purely virtual geometry (the inside of the box in this example) is rendered from the user's position and stored in a frame buffer.

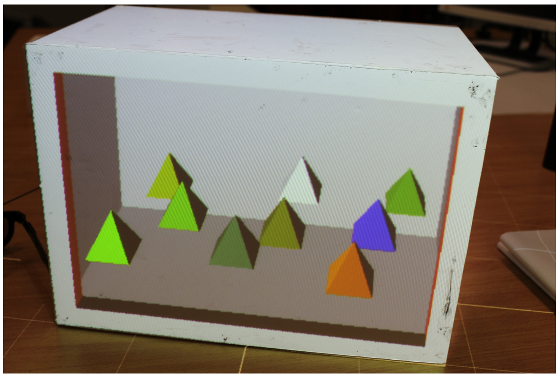


Figure 1: The virtual inside of a box and virtual wall depth.

2. When the geometry of the box is rendered for each projector, projective texture mapping techniques, using the same parameters as before, are used to project the virtual content onto the geometry of the physical object.

This creates a view-dependent image for a single tracked viewer and independent of any projector's position or projection parameters.

3 Depth Perception

Many individual depth cues are used by the visual system to provide an estimate of relative and absolute distances of objects within an image. Individual depth cues can be sorted by strength into three distinct 'action distances' (Cutting & Vishton 1995). In SAR, only near-field and mid-field actions are of interest.

Using view-dependent techniques in a SAR system creates two sets of conflicting depth cues: there are the 'real' depth cues, provided by the physical model of the box and the environment and there are the purely 'virtual' depth cues from the virtual content. It is desirable that the virtual depth cues' strength is at least as strong as the real depth cues so that the virtual content is perceived as strongly as the physical prop.

4 User Studies

Two user studies were designed to measure depth perception of projected virtual spaces rendered using view-dependent techniques. To do so, an indirect measuring task was created. A box (as seen in Figure 1) had a virtual window cut into one side. The inside was purely virtual and depicted nine pyramids at varying heights and positions within a 3×3 grid. Participants were asked to select the apex of a highlighted pyramid by placing the tip of a tracked pen on the top surface of the box at the location where they suspected the projected position of the apex would lie. The selection was blind – no feedback of the selection was provided to the participant. Task distance error was measured and was used to indicate the effectiveness of a depth cue. The order of selection, the relative positioning of pyramids to each other and the order of the conditions tested for were randomised to minimise any learning effect. Participants were not trained to perform the selection task prior to the study.

4.1 Difference between Real, Static and Head Tracked Perspectives

A first study investigated the difference of depth perception between a completely physical mockup of the virtual content, a static perspective (with head-tracking disabled) and the head-tracked perspective which provided depth parallax. Participants had no time constraint during selection. Three participants had previous experience with SAR installations, the others not. Thirteen valid data sets (N=13) were collected with 468 data points for the virtual and 234 data points for the real condition. The participant's mean age was 28 years (min 17, max 64) and the gender distribution was three females and ten male participants. All participants can be classified as 'expert' computer users with more than 50 hours of computer use per week, but only three had previous exposure to spatial AR systems. The results are listed in Table 1.

Condition	Mean	Median	SD
Real	14.53	14.98	3.97
Head tracking	16.24	16.79	3.51
Static	17.75	17.87	4.47

Table 1: Aggregated distance error in the first user study in mm sorted by mean error.

We found no significant difference between the three conditions ($F(2, 24.0) = 3.3, p = 0.054$).

4.2 Head Tracking with Additional Depth Cues

The second study compared head tracking to different other additional depth cues. The following four conditions were tested for: head tracking by itself and head tracking with one of the following: texturing on virtual content (Figure 2), shadows and virtual wall depth ('tunnel effect', see Figure 1). These conditions were chosen after an initial pilot study. Participants were asked to perform the selection as fast and accurately as possible. Eleven data sets were collected (N=11) with 396 data points per condition. Seven participants neither had experience with SAR installations nor participated in the first user study but all of the participants can be classified as 'expert' computer users. The participant's median age was 26 years (min 20, max 35); the gender distribution was one female participant and ten male. Table 2 shows the results of the second study.

One-way repeated measures ANOVA showed that there were no significant differences in accuracy between the different depth cues ($F(3, 30) = 1.658, p = 0.197$). Further post-hoc analysis revealed no other

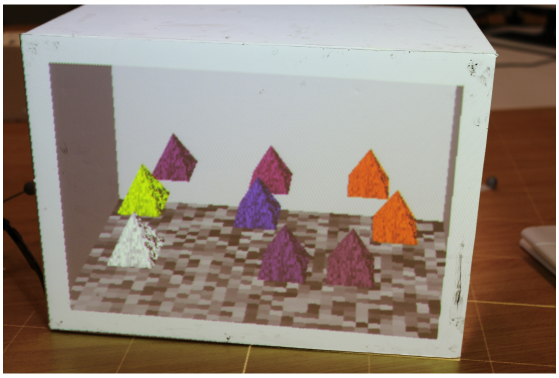


Figure 2: Textured pyramids.

Condition	Mean	Median	SD
HT + shadows	22.99	23.47	4.95
Head tracking	24.43	25.67	6.93
HT + texture	25.85	25.85	7.68
HT + wall depth	25.85	26.42	7.89

Table 2: Aggregated distance error values of the second user study in mm sorted by mean error.

significant differences between the error values, based on different depth cues. Therefore, different depth cues did not improve distance error over head tracking alone, which confirms the findings of the first user study. The increased error in all conditions, compared to the first user study can be explained by Fitt's Law, as there was no time-constraint for selections in the first study.

5 Discussion

This study was unfortunately unable to conclude that a certain depth cues significantly improves depth perception of virtual geometry. The performance of head tracking (parallax) improved selection error compared to the static perspective, however not significantly. Further work should investigate the impact of stereoscopic depth cues in near-field SAR.

References

Bennett, E. & Stevens, B. (2005), The Effect that Touching a Projection Augmented Model has on Object-Presence, *in* 'Information Visualisation, 2005. Proceedings. Ninth International Conference on', IEEE, pp. 790–795.

Cutting, J. E. & Vishton, P. M. (1995), Perceiving Layout and Knowing Distances: The Integration, Relative Potency and Contextual Use of Different Information about Depth, *in* W. Epstein & S. Rogers, eds, 'Handbook of perception and Cognition.', Vol. 5: Perception of Space and Motion, pp. 69–117.

Menk, C., Jundt, E. & Koch, R. (2011), 'Visualisation Techniques for Using Spatial Augmented Reality in the Design Process of a Car', *Comput. Graph. Forum* **30**(8), 2354–2366.

Raskar, R., Welsh, G., lim Low, K. & Bandyopadhyay, D. (2001), Shader Lamps, Technical report, MITSUBISHI ELECTRIC RESEARCH LABORATORIES, Eurographics Rendering Workshop.

Designing an Educational Tabletop Software for Children with Autism

Valerie Picardo[1] **Samuel Metson**[1] **Rashina Hoda**[1] **Robert Amor**[2]
Angela Arnold-Saritepe[3] **Rebecca Sharp**[3] **Denys Brand**[3]

[1]SEPTA Research
Department of Electrical
and Computer Engineering

[2]Department of Computer
Science, Faculty of Science

[3]Applied Behaviour Analysis
Programme, School of
Psychology, Faculty of Science

The University of Auckland
Private Bag 92019 Auckland, New Zealand

{vpic007, smet013}@aucklanduni.ac.nz
{r.hoda, r.amor, a.arnold-saritepe, r.sharp, denys.brand}@auckland.ac.nz

Abstract

Discrete Trial Training (DTT) is an intervention method used by behaviour analysts for teaching skills to children with Autism Spectrum Disorder (ASD) around a table using physical materials and artefacts. Traditional DTT suffers from these main challenges: inconsistency due to human limitations; need for focusing the child's attention on the trial at hand; and disruptions on account of in-session data recording and analysis by the analysts while delivering the training. Developed in collaboration with HCI and behaviour analysis experts, our proposed solution is an interactive, tabletop software application that provides the consistency and integrity that DTT aims to achieve, while engaging the child's attention on the interactive interface, and seamlessly collecting and analysing in-session data in the background. Upcoming usability evaluations of the prototype promise to provide insight into the potential effectiveness of our prototype.

Keywords: *Autism Spectrum Disorder (ASD), Discrete Trial Training (DTT), Child-Computer Interaction (CCI) Education, Children, Learning*

1 Introduction

Autism Spectrum Disorder is a neuro-developmental condition and Autism is one of three recognised disorders on this spectrum, and affects the cognitive development of the child. In the last 40 years, the rate of incidence of Autism Spectrum Disorders has risen from 4 in 10,000 to approximately 62 in 10,000, as of 2007 (Rutter 2005). Part of this increase can be attributed to better diagnoses and more awareness of ASD, but this still means that there are much greater numbers of people who require or will benefit from specialized teaching. Targeted methods must be used to teach basic behaviour and interaction skills to learners with developmental and learning disabilities.

Discrete Trial Training (DTT) is a teaching procedure based on the principles of Applied Behaviour Analysis to teach academic and other skills to children with neuro-developmental disorders, such as Autism (Smith 2001). DTT has four distinct parts: the trainer's presentation, the child's response (which may be prompted), the consequence, and a short interval between the consequence and the next instruction (Smith 2001). Performing trials in a consistent way is a large contributor to the effectiveness of DTT sessions. Traditionally, DTT sessions are conducted using physical materials while the analyst and child-learner sit around a physical table.

DTT is meant to carried out according to the scientific guidelines in a highly consistent format (Smith 2001). While this may sound feasible in theory, the reality is that DTT conducted in traditional settings is error-prone as the analysts conducting it are liable to be inconsistent and make mistakes on account of human limitations. Additionally, in-session data recording (e.g. correct/incorrect responses) and analysis by analysts is often inaccurate and disruptive to the primary aim of DTT (Smith 2001).

2 Assistive Technology

Technology can provide the consistency and integrity that DTT aims to achieve, while engaging the child's attention and thus immersing them in therapy for longer periods of time than they would otherwise sustain.

It has been suggested that the reason children with autism gravitate towards technology is because of its consistency, predictability, logic and freedom from social demands (Chen 2012). The concern that the use of computers may inhibit spontaneous language and social interaction in children with autism has not been conclusively evidenced by research. In fact, the children were found to have increased their motivation, use of spontaneous comments, eye gazes towards the parent, and positive affects while on the computer and decreased their inappropriate language and behaviour compared to baseline play sessions (Whalen et. al 2006). Touch-based applications are particularly suited for this purpose because of their popularity, especially among the younger demographic who have emerged as digital natives (Palfrey and Urs 2008).

A significant number of existing software applications, related to the DTT intervention were found on software app markets. However, many of these applications were designed by domain experts in the field of behavioural analysis alone and therefore did not harness technical expertise or the benefits of good HCI design. Despite their portability benefits, DDT applications on mobile touch devices such as on iPhone, iPad and Android have the disadvantage of limited screen real estate and non-collaborative allowance (Artoni et. al 2012). They also lack the support of multiple simultaneous touch gestures, which is crucial for instances when both the child and therapist could be interacting with the application at the same time. Given DTT sessions are typically conducted with the analyst and child-learner sitting around a table, a tabletop seems well-suited for our interactive software.

3 Project Overview

The hypothesis of our study is that multi-touch table-top software has the potential to engage children in autism therapy while allowing therapists/behaviour analysts to conduct and monitor DTT far more easily and accurately.

The aim of this project is to develop an interactive, multi-touch based software application that mimics the research-based intervention of DTT. In particular, this software focuses on teaching basic colour and shapes recognition. The software application aims to act as a novel learning/teaching tool to allow therapists/behaviour analysts to conduct and track sessions with children diagnosed with Autism, in a far more effective and fluid manner, whilst still having the flexibility to customise sessions for each child. Additionally, the software application should also engage the child in the learning process without any sensory overload.

This cross-disciplinary project harnesses the expertise of software engineering and HCI researchers and software developers, certified behaviour analysts and psychology researchers. Our team is working collaboratively to create a solution that leverages the technological affordances offered by touch-based platforms in order to enhance the effectiveness of DTT while preserving its pedagogical requirements and original structure.

As part of the design process, both pen-and-paper and electronic user interface prototypes were created to give a basic vision of the structure of the application. The paper prototype was rapidly created as a very basic view of the application, and discussed at meetings with the subject-matter experts (certified behaviour analysts.)

Using the Scrum software development method allowed us to develop this iteratively and incrementally weaving in feedback from collaborating behaviour analysts. The tabletop used is the Samsung SUR40 with Microsoft PixelSense which supports 50 simultaneous touch points and object recognition.

4 Design Features

The application is aimed at providing a holistic approach to DTT and consists of the following main features:

- Setting up trials customized to individuals
- Conducting trials (train for colours and shapes)
- Tracking and reviewing trials

The portion of the UI that is viewed by the learner is as simple and minimal as possible in an effort to reduce any distractions that can lead to the learner not attending to task instructions and to maximize the probability that the learner emits a correct response (Koegel et. al 1977). To use the app, a facilitator will log in, choose a child profile from those they are teaching, and either "train" the user (conduct a trial) or "track" them (review progress data). A trial setup can be customized to select: colours or shapes, number of trials in a session, and type of prompts. Analysts can also select to focus on certain colours by setting them as the correct response and mixing them with a number of other colours of their choice. As children with neuro-developmental disorders can be easily overstimulated or overwhelmed, the portion of the UI that will be presented and viewed by the child has been made as simple and minimal as possible to reduce any distracting factors and allow the child to focus on the task at hand. Data recorded includes correct and incorrect responses for each of the trials in order that trends over the latest DTT session and over time can be reviewed by the analysts.

5 Future Work

Human Ethics approval for conducting usability evaluations is in place. Relevant institutions are currently being invited to participate in our evaluations for usability and usefulness. We expect to discover any remaining usability issues which will then be rectified; and identify the application's effectiveness in assisting with DTT in the aforementioned areas (providing integrity and consistency, increased child engagement, and seamless data recording and analysis.)

6 References

Rutter, M. (2005): Incidence of autism spectrum disorders: Changes over time and their meaning, *Acta Pædiatrica*, **94**: 2-15.

Smith, T. (2001): Discrete Trial Training in the Treatment of Autism, *Focus on Autism and Other Developmental Disabilities*, **16**(2): 86-92.

Chen, W. (2012): Multi-touch Tabletop Technology for People with Autism Spectrum Disorder: A Review of the Literature, *Proceedings of the 4th International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion*, **14**: 198-207.

Whalen, C. et. al (2006): Behavioral Improvements Associated with Computer-Assisted Instruction for Children with Developmental Disabilities, *Journal of Speech and Language Pathology and Applied Behavior Analysis*, **1**(1): 17.

Palfrey, J. G., and Urs Gasser. (2008): Born digital: Understanding the first generation of digital natives. Basic Books.

Artoni, S. et. al. (2012): Designing a Mobile Application to Record ABA Data. *In Computers Helping People with Special Needs*. (1st Edition).

Koegel, R. L et. al (1977): Assessing and training teachers in the generalized use of behavior modification with autistic children. *Journal of Applied Behavior Analysis*, 10, 197 - 205.

Effects of 3D Display Technologies on Spatial Memory

Mostafa Mehrabi **Christof Lutteroth** **Burkhard C. Wünsche**

Department of Computer Science, University of Auckland, New Zealand
mmeh012@aucklanduni.ac.nz, {christof, burkhard}@cs.auckland.ac.nz

Abstract

Spatial memory has been recognized as an important factor in efficient human-computer interaction. However, most previous studies are limited to very simple depth cues and 2D displays. We present a study investigating the effects of head-coupled perspective, stereoscopy and simple landmarks on spatial memory by measuring performance and accuracy in a memory game. Our results indicate that head-coupled perspective affects spatial memory positively and should be investigated further. The polarized stereoscopic display and the landmarks used in this study had a significant negative effect, suggesting that they should be used with care. Users' perceived efficiency of a 3D display technology turned out to be a bad indicator of its actual efficiency.

Keywords: Spatial memory, 3D display technology

1 Introduction

Several studies show that users' spatial cognition is an important indicator of their performance when operating a user interface (Vicente et al. 1987). So it comes as no surprise that researchers have tried again and again to harness the power of spatial memory in user interfaces (Jones & Dumais 1986, Robertson et al. 1998). It is not clear whether adding depth cues to a 2D user interface actually enhances spatial memory: while some studies demonstrated improvements, e.g. Robertson et al. (1998), these improvements could be attributed to factors other than the depth cues, e.g. (Cockburn & McKenzie 2001). Moreover, the range of 3D display technologies that has been investigated with regard to spatial memory is rather narrow, using mostly primitive depth cues such as 3D perspective projection, size gradient and shadows.

In this study, investigate the effects of stereoscopy and head-coupled perspective (HCP) on spatial memory in a memory game (Figure 1). We incorporate 3D objects representing the data elements and simple landmarks consisting of a 3D room with distinctive wall, ceiling and floor patterns.

Figure 1: Screenshot of the 3D memory game with all objects uncovered

2 Methodology

The memory game task involved finding matching pairs of objects. In each round of the memory game, 20 objects (i.e. 10 matching pairs) were shown uncovered for five seconds, and then they got covered with boxes. In each move a player could uncover two objects by clicking on them, and if the two objects formed a matching pair, they stayed uncovered. Otherwise, they were covered again and the move was counted as a mistake to measure recall accuracy. Performance was measured as task completion time.

We measured user performance (task duration) and user accuracy (no. of mistakes made) in display conditions formed from different combinations of 3D perspective projection (3D), stereoscopy (S) using a polarized screen and polarized glasses, HCP (H) using a webcam and tracking software, and landmarks (L) involving flower wallpaper, a floor and a ceiling (Figure 1). Using a within-subject design, each participant played the game under nine display conditions: 2D, 3D 3DH, 3DL, 3DHL, 3DS, 3DSH, 3DSL and 3DSHL. The order of test conditions was permuted between participants to distribute any order effects. Object positions were chosen randomly from a set of pre-defined arrangements (without recurrence). Participants would play the memory game for five rounds in each test condition, of which the first two were training and did not count towards the experimental results. Participants had the option to rest between rounds. At the end of the experiment, each participant was asked to fill out a questionnaire.

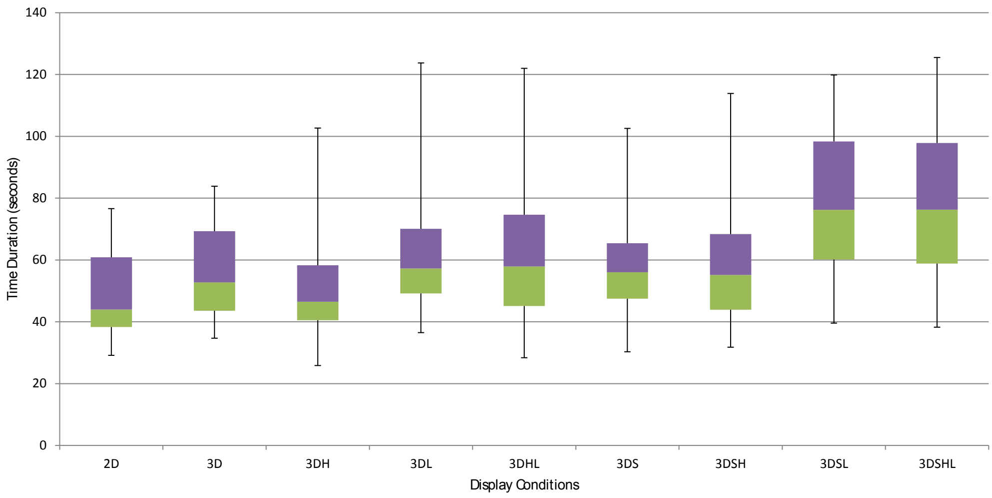


Figure 2: Task durations under different conditions (median and quartiles)

3 Results

We performed the experiment with 29 participants, which were chosen from different genders, age groups and occupations. Most of them were university students from various disciplines.

A boxplot of the task durations under the different conditions is shown in Figure 2. The results show a significant advantage of 2D and 3DH (3D projection with HCP) over all other display conditions, with no significant difference between the two. Landmarks and stereoscopy had significant negative effects, with 3DSL and 3DSHL performing worst.

There was not as much variation in the number of mistakes as there is for task duration. Noteworthy is only that simple 3D had significantly more mistakes than all other conditions except 3DSH and 3DSL. The only display technology that had a significant positive effect on the number of mistakes was HCP.

The questionnaire asked the participants to rank the display conditions with regard to their perceived efficiency and enjoyment. The results indicate that users' perceived efficiency was correlated with perceived enjoyment, but inversely correlated with the actual efficiency of display technologies as measured by task duration and number of mistakes.

Users were asked to think aloud during the task. Judging from that, the majority of participants believed that HCP was helpful because of adding more realism to the environment (i.e. the ability to see the environment from different view points) without imposing significant strain. The stereoscopic display used in our study caused fatigue and consequently deteriorated the performance. The landmarks added visual clutter and had therefore a negative effect on user performance. Other features such as the spatial arrangement of objects, their colors and a user's familiarity with them seemed to affect recall significantly.

4 Discussion

Unlike the 3D condition where the arrangements were random, the 2D condition always arranged all objects in a regular grid. Furthermore, the objects in the 2D condition were generally bigger than in the 3D conditions and had a clearer contrast to the background. Consequently, layout, object size and contrast were confounding variables that have likely made the 2D condition easier.

Technical problems in the head tracking system caused random jitter and occasional malfunction. Participants found this distracting, and it is a likely reason why participants mostly gave low rankings for HCP. This makes the positive effects of HCP the more surprising, and suggests HCP may perform even better if a more stable technology were used and the participants were more experienced in its use.

5 Conclusion

HCP seems to be a promising technology wrt. spatial memory and should be investigated further. Apparently stereoscopic displays can affect spatial memory negatively, likely because of fatigue. Landmarks need to be designed carefully, as they can have a significant negative effect on spatial memory by adding visual clutter to an environment. Finally, the perceived efficiency of a display technology is not necessarily an indicator of the actual efficiency, but may be more related to enjoyment. As future work, we will modify and replicate the study to remove confounding variables, especially in the 2D condition, and investigate the impact of other stereoscopic displays and types of landmarks.

References

Cockburn, A. & McKenzie, B. (2001), 3D or not 3D?: evaluating the effect of the third dimension in a document management system, *in* 'Proceedings of the ACM CHI Conference on Human Factors in Computing Systems', ACM, pp. 434–441.

Jones, W. P. & Dumais, S. T. (1986), 'The spatial metaphor for user interfaces: experimental tests of reference by location versus name', *ACM Transactions on Information Systems (TOIS)* **4**(1), 42–63.

Robertson, G., Czerwinski, M., Larson, K., Robbins, D. C., Thiel, D. & Van Dantzich, M. (1998), Data mountain: using spatial memory for document management, *in* 'Proceedings of the 11th ACM Symposium on User Interface Software and Technology', ACM, pp. 153–162.

Vicente, K. J., Hayes, B. C. & Williges, R. C. (1987), 'Assaying and isolating individual differences in searching a hierarchical file system', *Human Factors: The Journal of the Human Factors and Ergonomics Society* **29**(3), 349–359.

Scribbler – Drawing Models in a Creative and Collaborative Environment: from Hand-Drawn Sketches to Domain Specific Models and Vice Versa

Martin Vogel, Tim Warnecke, Christian Bartelt, Andreas Rausch
Clausthal University of Technology
Department of Informatics - Software Systems Engineering
Julius-Albert-Str. 4
38678 Clausthal-Zellerfeld, Germany

{m.vogel, tim.warnecke, christian.bartelt, andreas.rausch}@tu-clausthal.de

Abstract

In the early phases, software engineers use whiteboards and flip charts to create and discuss their ideas and later they transform manually the hand drawn pictures into machine readable models. During this transformation important sketch information, like the history of origin or some elements, will be lost. To solve this problem, we present a new approach using digital whiteboards to elaborate in a creative and collaborative environment hand drawn pictures and transform them into domain specific models and vice versa. This poster outlines the process of the automatic transformation from sketch models to models based on well-defined notations and vice versa in the early creative phases of software development.

Video: https://www.youtube.com/watch?v=0i3M9djPrRM

Keywords: Sketch Recognition, Model-Based Software Development, Collaborative Software Engineering

1 Introduction and motivation

These days software development is a distributed team process. The commercial success of products is increasingly important and the quality of the software has an important role in this context(Friedewald et al. 2001). However, less than 30% of all software projects are successful (Hartmann 2006). Especially critical are the early phases of the requirements analysis and the architectural design. Studies show, that the correction of errors from these early phases will cost up to 14 times as much as in case of errors in later phases (Westland 2002). Requirements analysis and architectural design are particularly important for the success of development projects. These early phases are also the creative phases. It is crucial for the success of a project, that teams work closely together – occasionally at different locations - and have strong cooperation. Thereby, handwritten sketches on whiteboards have been generally accepted in practice as the medium of communication. About 75% of developers like to use whiteboards during analysis and design (Cherubini et al. 2007) – even and especially in team workshops.

Normally software engineers do not use modeling tools like MagicDraw UML (MagicDraw 2013) in this early phase, because of their cumbersome handling. Modeling tools are made for precise model documentation, and not for creative sketching. Hence software engineers use whiteboards and flip charts in team meetings to create and discuss their ideas in form of various hand drawn pictures using well defined notations, like for instance, UML diagrams or Flow Chart diagrams. The existing (and partly expensive) modelling tools are used later on, to manually transform the hand drawn pictures into machine readable models for the next development tasks, like for instance, code generation. In follow-up meetings software engineers use the old hand drawn pictures and print-outs of the afterwards documented and further developed models to discuss and further improve their solutions. Thereby, they again produce additional hand-drawn pictures which again have to be later on manually integrated into the existing machine-readable models. Another topic is that some team members will not recognize the shown model, because in the transformation some important information is lost or a team member might try to optimize the resulting sketch model in the modeling tool.

2 Scribbler: A model sketching environment

To support software engineers in the early phases, we developed a collaborative and creative modeling tool called Scribbler. Scribbler uses digital whiteboards to transform free hand sketches in models based on well-defined notations and back again, saving the history of origins, recognizing objects and handwriting.

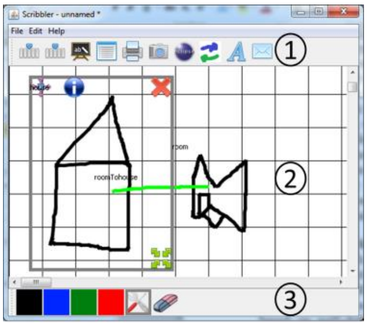


Figure 1: The Scribbler tool

Scribbler is portable for every device which uses java and supports ad-hoc meetings between geographically distributed locations. The transformation is completely independent from a pre-defined modeling language. The tool provides a training mode to learn new graphical syntax elements and map these to formal meta models entities. Furthermore developers can quickly create new pluggable components for their own requirements (Bartelt et al. 2013). A screenshot of the tool is shown in Figure 1. At the top (1), all current loaded plugins (e.g. learning) are represented with an icon, in the center (2) the canvas and last but not least the toolbar is located at the bottom (3), which consists of four colors, an edit button and a rubber.

3 Transforming sketch models to models based on well-defined notations and back again

Transforming a hand drawn sketch to a formal based model, describes a process, of mapping sketches to elements of the given meta model. Figure 2 shows a small Ecore (EMF) meta model.

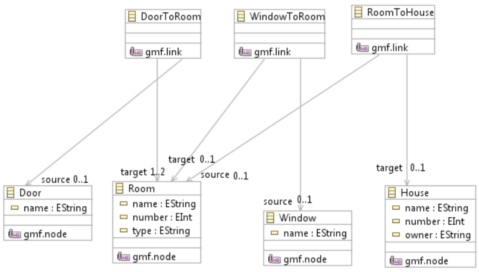


Figure 2: Ecore meta model

This meta model consists of entities like houses, rooms, windows and doors, and connections between this elements. In this example DoorToRoom, WindowToRoom and RoomToHouse are connections represented as lines. Room, Door, Window and House are represented by meaningful icons.

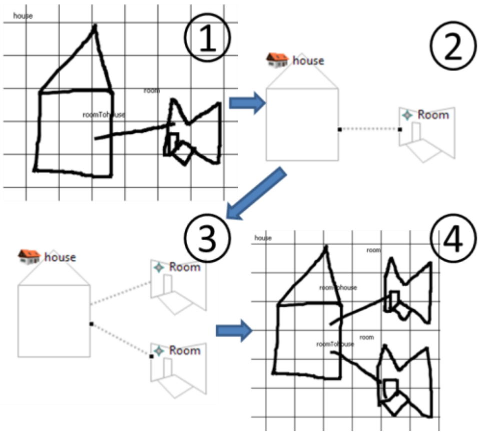


Figure 3: Transforming sketches into a model and back again

As mentioned above, the process of the transformation from sketch models to models based on well-defined notations is error-prone. Almost always the sketch information, like the history of origin, is lost and the formal model looks different to the sketch. To fix this problem, Scribbler can export sketched models to models based on well-defined notations which are generated with EMF – the Eclipse Modeling Framework - without losing sketch information. Furthermore, Scribbler can import these models from the EMF-Editor. Figure 3 shows an example of the import/export process of a sketched model. In **phase 1,** the team created a sketch model. After the meeting Scribbler can transform automatically the sketch model to a model based on well-defined notations. Thereto, Scribbler detects collision and inclusion, maps the recognized sketch objects to concrete elements of the Ecore meta model, stores the sketch model into a special tag and stores it into a file. **Phase 2** displays the formal model which is the result of the transformation. It is easy to see that the elements are at the same position and have roughly the same size like in the sketch model in. After the transformation of a sketch model to a formal model, the user is able to work on the model in the editor. In this example a user changes the position of the room and adds a new room. The modifications in the editor can be seen in **phase 3**. After that, the formal model can be imported in Scribbler. During the import Scribbler verifies the information of the formal model file. If there are valid sketch information Scribbler draws these elements on the canvas. If the position or size has changed, Scribbler adapts the coordinates. In this example a new room was added in the GMF-Editor. The application checks to which element in the knowledge base room is mapped and draws the room with this information. **Phase 4** shows the result of the import from the formal model file. Scribbler allows distributed, parallel (collaborative) sketching of models on digital whiteboards and tablet pcs, the transformation of those sketches in model based on well-defined notations and back again, recognize objects of every domain, recognize hand writing, an easy and interactive learning of new domain specific syntax elements and a recording/playback of the modeling/sketching history.

4 References

Bartelt, C., Vogel, M. & Warnecke, T., 2013. Collaborative Creativity: From Hand Drawn Sketches to Formal Domain Specific Models and Back Again. In A. Nolte et al., eds. *MoRoCo@ECSCW*. CEUR Workshop Proceedings. CEUR-WS.org, pp. 25–32. dblp.uni-trier.de/db/conf/ecscw/ moroco2013.html#BarteltVW13.

Cherubini, M. et al., 2007. Let's go to the whiteboard: how and why software developers use drawings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. New York, NY, USA: ACM, pp. 557–566.

Friedewald, M. et al., 2001. Softwareentwicklung in Deutschland: Eine Bestandsaufnahme. *Informatik Spektrum*, 24(2), pp.81–90.

Hartmann, D., 2006. Interview: Jim Johnson of the Standish Group. Available at: http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS [Acc. Nov 3, 2013].

MagicDraw, 2013. NoMagic - MagicDraw. Available at: www.nomagic.com/products/magicdraw.html [Acc. Nov 3, 2013].

Westland, J.C., 2002. The cost of errors in software development: evidence from industry. *J. Syst. Softw.*, 62(1), pp.1–9.

Casual Mobile Screen Sharing

Joris Suppers & Mark Apperley
Department of Computer Science, University of Waikato
Hamilton, New Zealand
jorissuppers@hotmail.com

Abstract

The concept of casual screen sharing is that multiple users can cast screen images from their personal hand-held devices on to a large shared local screen. It has applications in personal and business domains where documents or images need to be discussed in a shared environment. The 'casual' qualifier implies that the overheads of this sharing should be minimal. Implementation of casual screen sharing poses two general problems: sending content from multiple devices with minimal or no authentication/authorisation, and displaying this content on the larger screen. This paper proposes a solution and describes the development of a prototype, CasualShare.

Keywords: screen-sharing, public display, casual sharing

1 Introduction

Screen sharing is an effective method to share what one person sees on their device's screen with the screen of another device. The technique typically involves a host sharing its screen to a client's screen through an appropriate communication channel. The potential of this technique for a mobile device is significant, as it enables viewing on a larger screen which is more convenient for an audience. There are some particular problems that arise with existing screen sharing techniques:

The first problem is how the host and client will connect to each other. This can be a connection via cord, Bluetooth, wireless local network *(wifi)* or through the internet. Many existing techniques require long-winded authentication or synchronisation between the devices; this affects the usability of these techniques and can be frustrating to the users.

The second problem is that of providing for multiple connections. Many existing techniques allow only a single screen sharing session, because when more than one device is sharing their screen to a single server a variety of new problems occur. These include (i) the need to handle more connections which can cause delays, and (ii) the allocation of the real-estate on the large screen.

2 CasualShare

This paper describes a MAC OS X program called CasualShare which runs as a server and receives content from client(s) within a local area network (LAN), through Airplay™ (Apple, a) and a VNC connection (Fleishman, 2010). Both VNC and Airplay are used because they support wireless transfer, and require minimal authentication. The VNC connection is used for mirroring the whole of a client's screen, while Airplay is used only for sending images and videos. Once thiscontent is sent to the server, it is subject to manipulation by CasualShare to be displayed on a large screen. This overcomes both problems stated earlier: by allowing clients to share content with no authentication once connected to the same local area network (LAN) – it assumes whoever is connected to the network is trustworthy – and displaying content appropriately on a large screen.

3 Airplay

Airplay comprises a mix of protocols created by Apple, to view media from an iOS device or from a computer, using iTunes™ (Apple, b), to an Airplay receiver. Airplay supports images, videos, audio and also screen mirroring which requires special encoding hardware that does not use too much CPU power. iPhone 4S, iPad2 and Macs with Sandy Bridge CPU's include this hardware. Airplay locates compatible devices automatically by using Apple's Bonjour software (Apple, c), without requiring any configuration. This is achieved by using a DNS-based service discovery (Krochmal & Cheshire, 2011), and allows an Airplay server to be found from an iOS device simply by clicking the Airplay icon (see Figure 1).

With CasualShare, to send an image via Airplay once connected is simple; whatever image is displayed on the device is automatically sent to the server. A slideshow session is started by going to the first image of the slideshow then pressing the play button, located on the bottom of the image. Then every three seconds a new image is sent to the server. The duration can be changed in the device's settings, and the slideshow can be stopped at any time by pressing the pause button.

Videos are sent to the server in the same manner as slideshows. Pressing the play button while dislaying a video will send that video at the current play position to the server; the device keeps control over the video so it can still pause and scrub. The video is shown with a custom controller, placed in the centre bottom; this controller has four controls: play/pause, scrubbing scroll bar, full screen button and a volume scroll bar, so the server also has control over the video. These buttons are transparent and fade out when the mouse is inactive; this makes it more aesthetically pleasing and less distracting for the user(s) to view the images and videos.

4 VNC Connection

CasualShare uses FlashLight-VNC (Fucci, 2010), a lightweight open source VNC Client written in Adobe Flash. It is used to connect to a VNC Server (Richardson, *et al*, 1998) like Veency (Simon, 2008) which will be running on the mobile device to mirror its screen and also allow control from a mouse and keyboard.

To locate VNC Servers running on the network this project uses Nmap (Lyon, n.d.) ("Network Mapper") which is a free and open source utility for network discovery and security auditing enabling fast port scanning within an IP range. This allows the software to scan all addresses on the network to see which addresses have the VNC port open, allowing a user to easily connect to a VNC server without manually entering an IP address.

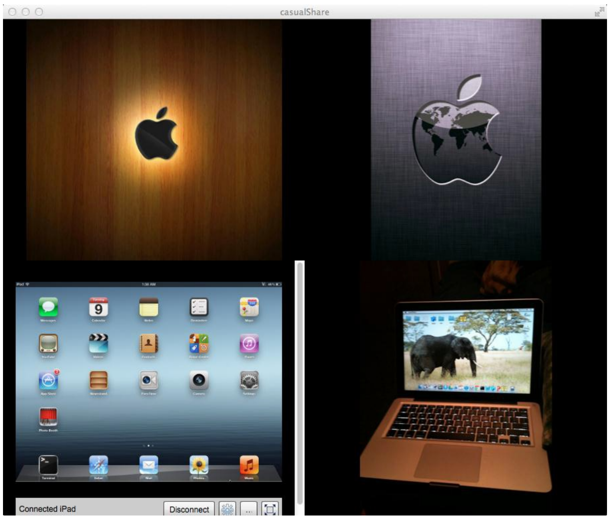


Figure 1: Multiple devices connected (images top and bottom right, mirroring session on bottom left)

5 Display Content

This server is also capable of displaying media simultaneously with multiple devices. This means several devices can send images, videos or mirror their screens to the server at the same time. The server partitions the screen appropriately so that each device receives the same proportion of the screen (see Figures 1).

There is currently no limitation on how many devices can connect. When a new device connects, the server simply checks to see if the screen needs splitting to create a new space, and if it does, it will be split vertically before horizontally, as screens are typically wider.

6 Preliminary Results

While extensive user testing has not been done, we have conducted a usability study on CasualShare with five participants. The goal of this study was to gather and analyse findings about the perceived strengths and weaknesses of CasualShare.

The results obtained through this study show that using Airplay to send images and videos to CasualShare is the favoured feature, as it is straight forward and easy for users to control. The mirroring feature proved to be the hardest and most undesirable feature of CasualShare, as it required the user to navigate to the mirroring tab, scan and connect to a specific IP address.

The study revealed that CasualShare was overall effective and easy to use, and that all participants found the software useful and would use the system again for personal and/or business use.

7 Conclusion and Further work

The aim of this project was to produce a casual but efficient technique for screen sharing from a mobile device to a larger screen. This has been achieved by utilising existing techniques and addressing their weaknesses. CasualShare, the screen sharing software created by this project runs on a Macintosh computer and displays images, videos and a mirroring session, sent wirelessly from a mobile device. The improvements achieved by CasualShare regarding screen sharing include: wireless transfer of media through a local area network (LAN); casual approach with no authentication (once connected to LAN); multiple connections from clients to the server; the display of media from multiple devices with appropriate partitioning; caching media to reduce latency; and full control of media from server and client side;

CasualShare is freely available through the website www.joris.co.nz. At the time of writing there had been some 1500 downloads over the 10 months it had been available.

CasualShare could be taken along several different paths to further improve its functionality. Screen mirroring is a capability of the latest Airplay protocol. This is achieved through an encoded video stream which is packetized holding a 128-byte header. One improvement for future work would be to include screen mirroring by decoding the video bit stream. This would allow mirroring without the need of a VNC connection.

8 References

Apple (a): Play content from your iPhone, iPad, or iPod touch on your HDTV. Apple Inc. http://www.apple.com/airplay/. Accessed September 2012.

Apple (b): Entertain all the possibilities. Apple Inc. http://www.apple.com/itunes/. Accessed September 2012.

Apple (c): Bonjour. Apple Inc. http://www.apple.com /support/bonjour/. Accessed September 2012.

Fleishman, G. (2010): VNC. *Macworld,* **10**(27): 52.

Fucci, M. (2010): FlashLight-VNC Open Source VNC Client written in Flash. http://flashlight-vnc.sourceforge.net/. Accessed September 2012.

Krochmal, S. & Cheshire, M. (2011): DNS-Based Service Discovery. December 2011. http://www.ietf.org/id/draft-cheshire-dnsext-dns-sd-11.txt.

Lyon, G. (n.d.): Nmap. http://nmap.org/. Accessed September 2012.

Richardson, T., Stafford-Fraser, Q., Wood, K.R. & and Hopper, A. (1998): Virtual Network Computing. *Internet Computing,* **2**(1): 33-38.

Simon, N. (2008): How To Use Veency to Remote Control your iPhone. http://www.simonblog.com/2008/10/19/how-to-use-veency-to-remote-control-your-iphone/. Accessed October 2012.

Author Index

Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website `http://crpit.com`.

Volume 124 - Database Technologies 2012
Edited by Rui Zhang, The University of Melbourne, Australia and Yanchun Zhang, Victoria University, Australia. January 2012. 978-1-920682-95-8.
Contains the proceedings of the Twenty-Third Australasian Database Conference (ADC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 125 - Information Security 2012
Edited by Josef Pieprzyk, Macquarie University, Australia and Clark Thomborson, The University of Auckland, New Zealand. January 2012. 978-1-921770-06-7.
Contains the proceedings of the Tenth Australasian Information Security Conference (AISC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 126 - User Interfaces 2012
Edited by Haifeng Shen, Flinders University, Australia and Ross T. Smith, University of South Australia, Australia. January 2012. 978-1-921770-07-4.
Contains the proceedings of the Thirteenth Australasian User Interface Conference (AUIC2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 127 - Parallel and Distributed Computing 2012
Edited by Jinjun Chen, University of Technology, Sydney, Australia and Rajiv Ranjan, CSIRO ICT Centre, Australia. January 2012. 978-1-921770-08-1.
Contains the proceedings of the Tenth Australasian Symposium on Parallel and Distributed Computing (AusPDC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 128 - Theory of Computing 2012
Edited by Julián Mestre, University of Sydney, Australia. January 2012. 978-1-921770-09-8.
Contains the proceedings of the Eighteenth Computing: The Australasian Theory Symposium (CATS 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 129 - Health Informatics and Knowledge Management 2012
Edited by Kerryn Butler-Henderson, Curtin University, Australia and Kathleen Gray, University of Melbourne, Australia. January 2012. 978-1-921770-10-4.
Contains the proceedings of the Fifth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 130 - Conceptual Modelling 2012
Edited by Aditya Ghose, University of Wollongong, Australia and Flavio Ferrarotti, Victoria University of Wellington, New Zealand. January 2012. 978-1-921770-11-1.
Contains the proceedings of the Eighth Asia-Pacific Conference on Conceptual Modelling (APCCM 2012), Melbourne, Australia, 31 January – 3 February 2012.

Volume 133 - Australian System Safety Conference 2011
Edited by Tony Cant, Defence Science and Technology Organisation, Australia. April 2012. 978-1-921770-13-5.
Contains the proceedings of the Australian System Safety Conference (ASSC 2011), Melbourne, Australia, 25th – 27th May 2011.

Volume 134 - Data Mining and Analytics 2012
Edited by Yanchang Zhao, Department of Immigration and Citizenship, Australia, Jiuyong Li, University of South Australia, Paul J. Kennedy, University of Technology, Sydney, Australia and Peter Christen, Australian National University, Australia. December 2012. 978-1-921770-14-2.
Contains the proceedings of the Tenth Australasian Data Mining Conference (AusDM'12), Sydney, Australia, 5–7 December 2012.

Volume 135 - Computer Science 2013
Edited by Bruce Thomas, University of South Australia, Australia. January 2013. 978-1-921770-20-3.
Contains the proceedings of the Thirty-Sixth Australasian Computer Science Conference (ACSC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 136 - Computing Education 2013
Edited by Angela Carbone, Monash University, Australia and Jacqueline Whalley, AUT University, New Zealand. January 2013. 978-1-921770-21-0.
Contains the proceedings of the Fifteenth Australasian Computing Education Conference (ACE 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 137 - Database Technologies 2013
Edited by Hua Wang, University of Southern Queensland, Australia and Rui Zhang, University of Melbourne, Australia. January 2013. 978-1-921770-22-7.
Contains the proceedings of the Twenty-Fourth Australasian Database Conference (ADC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 138 - Information Security 2013
Edited by Clark Thomborson, University of Auckland, New Zealand and Udaya Parampalli, University of Melbourne, Australia. January 2013. 978-1-921770-23-4.
Contains the proceedings of the Eleventh Australasian Information Security Conference (AISC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 139 - User Interfaces 2013
Edited by Ross T. Smith, University of South Australia, Australia and Burkhard C. Wünsche, University of Auckland, New Zealand. January 2013. 978-1-921770-24-1.
Contains the proceedings of the Fourteenth Australasian User Interface Conference (AUIC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 140 - Parallel and Distributed Computing 2013
Edited by Bahman Javadi, University of Western Sydney, Australia and Saurabh Kumar Garg, IBM Research, Australia. January 2013. 978-1-921770-25-8.
Contains the proceedings of the Eleventh Australasian Symposium on Parallel and Distributed Computing (AusPDC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 141 - Theory of Computing 2013
Edited by Anthony Wirth, University of Melbourne, Australia. January 2013. 978-1-921770-26-5.
Contains the proceedings of the Nineteenth Computing: The Australasian Theory Symposium (CATS 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 142 - Health Informatics and Knowledge Management 2013
Edited by Kathleen Gray, University of Melbourne, Australia and Andy Koronios, University of South Australia, Australia. January 2013. 978-1-921770-27-2.
Contains the proceedings of the Sixth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 143 - Conceptual Modelling 2013
Edited by Flavio Ferrarotti, Victoria University of Wellington, New Zealand and Georg Grossmann, University of South Australia, Australia. January 2013. 978-1-921770-28-9.
Contains the proceedings of the Ninth Asia-Pacific Conference on Conceptual Modelling (APCCM 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 144 - The Web 2013
Edited by Helen Ashman, University of South Australia, Australia, Quan Z. Sheng, University of Adelaide, Australia and Andrew Trotman, University of Otago, New Zealand. January 2013. 978-1-921770-15-9.
Contains the proceedings of the First Australasian Web Conference (AWC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 145 - Australian System Safety Conference 2012
Edited by Tony Cant, Defence Science and Technology Organisation, Australia. April 2013. 978-1-921770-13-5.
Contains the proceedings of the Australian System Safety Conference (ASSC 2012), Brisbane, Australia, 23rd – 25th May 2012.