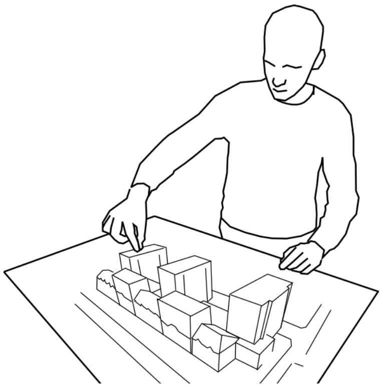
UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO



Direct Interactive 3D Modeling in a Semi-Immersive Environment

Bruno Rodrigues De Araújo

Supervisor: Doctor Joaquim Armando Pires Jorge
Co-Supervisor: Doctor João António Madeiras Pereira

Thesis approved in public session to obtain the PhD Degree in Information Systems and Computer Engineering

JURY FINAL CLASSIFICATION: **Pass With Distinction**

Jury

Chairperson: Chairman of the IST Scientific Board
Members of the Committee:
Doctor Karan Sher Singh
Doctor Nuno Manuel de Carvalho Ferreira Guimarães
Doctor José Manuel Pinto Duarte
Doctor Joaquim Armando Pires Jorge
Doctor João António Madeiras Pereira
Doctor Carlos António Roque Martinho

2013

UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Direct Interactive 3D Modeling in a Semi-Immersive Environment

Bruno Rodrigues De Araújo

Supervisor: Doctor Joaquim Armando Pires Jorge

Co-Supervisor: Doctor João António Madeiras Pereira

Thesis approved in public session to obtain the PhD Degree in Information Systems and Computer Engineering

JURY FINAL CLASSIFICATION: **Pass With Distinction**

Jury

Chairperson: Chairman of the IST Scientific Board

Members of the Committee:

Doctor Karan Sher Singh , Professor Catedrático, University of Toronto, Canada

Doctor Nuno Manuel de Carvalho Ferreira Guimarães , Professor Catedrático do Instituto Superior de Ciênciais do Trabalho e da Empresa - Instituto Universitário de Lisboa

Doctor José Manuel Pinto Duarte, Professor Catedrático da Faculdade de Arquitectura, da Universidade Técnica de Lisboa

Doctor Joaquim Armando Pires Jorge, Professor Catedrático do Insituto Superior Técnico, da Universidade Técnica de Lisboa

Doctor João António Madeiras Pereira, Professor Associado do Insituto Superior Técnico, da Universidade Técnica de Lisboa

Doctor Carlos António Roque Martinho, Professor Auxiliar do Insituto Superior Técnico, da Universidade Técnica de Lisboa

Funding Institutions

Fundação para a Ciência e a Tecnologia (SFRH/BD/31020/2006)

2013

http://vimmi.inesc-id.pt/~brar

brar@vimmi.inesc-id.pt

bdearaujo@gmail.com

Resumo

Apesar da crescente popularidade dos Ambientes Virtuais, estes ainda não oferecem uma alternativa válida para modelar cenarios tridimensionais quando comparados ás tradicionais ferramentas de Concepção Assistida por Computador. Os diálogos utilizados pela Realidade Virtual permanecem demasiados influenciados pelas metáforas utilizadas pelos ambientes de trabalho 2D, nomeadamente a metáfora *Windows, Icons, Menus, Pointing*. Esta abordagem não mapeia da melhor forma como as pessoas concebem, pensam e manipulam formas tridimensionais. O aparecimento de novas tecnologias, tais como câmara de profundidade, superfícies interactivas e novos dispositivos de entrada 3D, motiva um novo olhar nas interfaces de modelação 3D recorrendo a visualização estereoscópica para propor uma interface utilizador que tire maior partido da manipulação directa.

Este trabalho propõe a utilização de ambientes virtuais para a concepção de formas tridimensionais baseada em esboços e interfaces gestuais. Combinamos a técnica de interacção "sobre e por cima de superfícies interactivas" com o modelo de interacção bimanual de Guiard de forma a oferecer um espaço contínuo de interacção dedicado a modelação 3D. A nossa abordagem combina uma superfície multitoque com visualização estereoscópica e dispositivos de seguimento das mãos e dedos no espaço acima desta. Os resultados confirmaram que esta solução oferece um ambiente de modelação alternativo tirando maior partido de ambos os espaços de interacção.

Palavras Chave

Modelação 3D

Interfaces Utilizador 3D

Ambientes Virtuais

Modelação baseada em Esboços

Abstract

Despite the growing popularity of Virtual Environments, they have not yet replaced desktop Computer Aided Design systems when it comes to modeling 3D scenes. Traditional Virtual Reality idioms are still umbilically connected to the desktop metaphor they aim to replace, by leveraging on the familiar "Windows, Icons, Menus, Pointing" metaphors. Worse, the command languages underlying many of these systems do not map well to the way people learn to conceive, reason about and manipulate three-dimensional shapes. New and affordable technologies such as depth cameras, multi-touch surfaces and multi-sensor devices motivate a fresh look at semi-immersive interfaces to develop modeling interfaces that better support direct interaction.

We explore semi-immersive environments for conceptual design where virtual mockups are obtained from sketches and gestures. We applied on-and-above-the-surface interaction techniques following Guiard's asymmetric bimanual model to take advantage of the continuous interaction space for creating and editing 3D models in a stereoscopic environment. To allow for more expressive interactions, our approach continuously combines multi-touch on the stereoscopic surface with hand and finger tracking in the space above it. Results confirm that this combination produces an alternative modeling environment where users can seamlessly switch both interaction spaces to leverage the benefit of both interaction spaces.

Keywords

3D Modeling

3D User Interfaces

Virtual Environments

Sketch Based Modeling

Acknowledgements

After five years, I would like first to thank my closest family: Papa, Maman, Madrinha, Padrinho, my brother Gabriel, my almost sisters Rachel and Tania, Ricardo, André and my cousins Carlos, Marta and Pedro. This thesis is dedicated to them for their valuable support, patience and understanding, they always believed in me and balanced my days during all these years.

I would like to thank Professor Joaquim Jorge as my adviser for its support, motivation and recommendations along all these years. He started to share with me his enthusiasm for Computer Graphics as a teacher fifteen years ago and accepted to advise me during the last twelve years closely following my academic studies. I am deeply grateful to you for your availability, advices, patient and faith which inspired me and always make me feel in my research that independently of the challenges they could be overcome. I would like also to thank my co-adviser Professor João Madeiras Pereira for its support and advice. During all these years, he never doubt in my capabilities, always supported me and allowed me to keep up other research topics which interested me. I would like also to thanks the other members of my thesis committee: Professor José Pinto Duarte and Professor Carlos Martinho for their valuable comments remarks,advise and dedication which I admire and helped me to focus along all my workplan. Finally, I would like to thank Professor Karan Singh and Professor Nuno Guimarães who accepted the invitation to be part of my jury and helped in reviewing and improving my dissertation. Their research work and personality both inspired and motivated me and I am glad they accepted to be part of my achievement.

All my academic course was done at Instituto Superior Técnico, I would like to thanks this institution which shape my education in particular the Computer Graphics and Multimedia teachers which my advisers are part of as well as Professor Mario Rui Gomes and Professor João Brisson Lopes. I would like also to thank the research institution INESC-ID who hosted me during the last twelve years. In particular the VIMMI group where I collaborated with several people and institutions along several National and European research projects. First, I would like to thanks my colleagues in special Ricardo Jota and João Oliveira with who I shared the office, passed several good moments and had great discussions. Tiago Guerreiro, Luis Carlos Bruno and Hugo Nicolau also participated in this adventure and they also shared the same quest. We helped each other and their successes were a strong motivation to end my own quest. These five guys has been always available in and out of our lab walls. A special thanks also to Alfredo Ferreira, Manuel João Fonseca, Daniel Gonçalves who complete this team from my beginning at the VIMMI group and share their valuable experience on the same quest. I would like to thanks all my VIMMI colleagues Filipe Dias, Vasco Gervasio, Vasco Costa, Claudia Ribeiro, Ferran Naya, Pauline Jepp, Gabriel Barata, Ricardo Dias, Micael Carreira, João Guerreiro and in particular Diogo Mariano, Pedro Lopes, João Fernandes, Daniel Mendes, Fernando Fonseca, Mauricio Sousa, Sergio Azevedo, Luis Lopes and José Pedro Dias with who I worked closely at key steps of my thesis. During this thesis, I did two internships at INRIA Bordeaux and INRIA Lille Nord Europe which allowed me to focus and provided a valuable boost to fulfill the goals of my thesis. I would like to thank Martin Hachet for accepting me at his lab during two months and a special thanks to Géry Casiez with who I closely collaborate during my four month staying at the MINT group.

I would like also to thanks my architectural colleagues and friends which I collaborated along this five years: Eduardo Castro e Costa, Filipe Coutinho, Bruno Figueiredo, Luis Sousa and my good friend António Esteves. I had great discussions and exchange of ideas which clearly influenced and inspired my work and allowed to have a closer look to the architectural domain visiting educational and professional installations namely at the Architectural Faculty of Lisbon and the Architectural Faculty of Coimbra. They participated on key steps of my work from its conception to its evaluation. I would like to thanks the Architectural Department of Instituto Superior Técnico, in particular Professor Helena Rua and Professor Ana Tomé for providing me users to evaluate my approach. I thank all the users that participated to my user test and both Mauricio Sousa and Daniel Mendes for their valuable support on its execution or finding more users.

Finally, I would like to thank the institutions and projects which financially supported and allowed me to perform the work done in this thesis. In particular Portuguese Foundation for Science and Technology (FCT) for giving me the 4 year scholarship that allowed me to focus on my research work: this work was supported by FCT doctoral grant SFRH/BD/31020/2006. I would like to thank the host institution INESC-ID under contract Pest-OE/EEI/LA0021/2013. Part of this work was also partially supported by the Portuguese Foundation for Science and Technology (FCT) through the projects Alberti Digital PTDC/AUR-AQI/108274/2008, MIVIS PTDC/EIAEIA/104031/2008, CEDAR PTDC/EIA-EIA/116070/2009 and by other institutions through the projects ANR InSTInCT project ANR-09-CORD-013 and the Inter-reg IV-A 2 seas SHIVA project.

Contents

List of Figures

List of Tables

Acronyms

API Application programming interface. 134, 136

BRep Boundary Representation. 59, 69, 134

C-BRep Combined Boundary Representation. 25–27, 58

CAD Computer Aided Design. 1, 4–11, 14–17, 26, 36–38, 40, 43, 46, 47, 56, 57, 59–62, 66, 68, 70, 72, 73, 77, 78, 82, 90, 97, 134

CG Computer Graphics. 4, 10, 15, 16, 22, 79

CGA Computer-Generated Architecture. 21, 24, 25, 30, 31, 69

CSG Constructive Solid Geometry. 37–39, 66–68, 71–73

DH dominant hand. 92–96, 98–102, 106, 108, 124, 142

DoF Degrees of freedom. 55, 59, 62

DPI dots per inch. 113, 114

FL-System Functional Lindenmayer System. 28

GIS Geographic Information System. 23, 59, 73

GML Generative Modeling Language. 20, 21, 26, 27, 31, 69

GUI Graphical User Interface. 25, 47, 82

HCI Human Computer Interaction. 4, 9, 10

HMD head mounted display. 1, 2, 60–64, 68

IR Infrared. 54, 113

L-System Lindenmayer System. 22, 23, 28, 33, 34

NDH non dominant hand. 92–94, 96, 98–104, 106, 108, 124, 142

NURBS Non–uniform rational B–spline. 59

OSC Open Sound Control. 116

RF Radio Frequency. 54

RFID Radio-frequency identification. 52

SBM Sketch Based Modeling. 36, 37, 43, 46

STHMD see-through head mounted display. 60, 73

TUIO tangible user interface objects. 113, 116, 117

VE Virtual Environment. 2, 4–8, 11

VR Virtual Reality. 55, 56, 59, 73

VRML97 Virtual Reality Modeling Language 97. 28

WIMP windows, icons, menus, and a pointing device. 1–4, 6, 7, 10, 36, 47, 55, 67

XML Extensible Markup Language. 18, 62

1

Introduction

Stereoscopic Visualization brought new possibilities for 3D modeling with the emergence of affordable 3D devices such as head mounted displays (HMDs), 3D screens and 3D projection systems. These provide better perception of shapes than traditional 2D desktop based displays, while offering a better perception of the spatial relationships, proportions and interference between 3D elements. However, current modeling applications still rely on complex 2D graphical user interfaces based on windows, icons, menus, and a pointing device (WIMP) The increase in geometrical functionality has lead these interfaces to require extensive training to become too complex and unnatural to operate for most designer tasks. This has resulted on a lack of productivity when using such tools for design review and added important cost penalties to the design process since more effort is needed to perform changes. Furthermore, the additional cognitive costs constrain the creativity needed in conceptual phases. In addition, low level manipulation should be more accessible and easier to be defined. Higher level modeling representations, closer to the user domain and interaction metaphors are needed on top of existing techniques to turn 3D Virtual Environments (VEs) into a reliable solution for design and not just for visualization or exploration as it is done nowadays.

Past research has tried to overcome such problems following different approaches: complementing existing WIMP based applications, adapting them and creating new interaction metaphors for 3D environment and presenting more effective modeling approaches. The more promising 2D interfaces have relied on sketching combining gesture, suggestive mechanisms and constraint-based modeling. Several solutions have been proposed to enable the construction of 3D models based on 2D interfaces assisting the user or interpreting and reconstructing drawings. These approaches have been possible thanks to the extension of the geometric representation in order to propose modeling operators more adapted to new interaction metaphors. While these interfaces take advantage of user drawing skills, existing navigation methods still do not offer a good perception of the 3D space compared to physical representations of the model.

Virtual environments can overcome these issues and bridge the gap between both digital and physical representations even in conceptual design phases. In particular, semi-immersive environments enable to take advantage of stereoscopic displays to co-localize the physical interaction space with the virtual representation. As opposed to immersive solutions such as those based on HMDs, semi-immersive scenarios allow the user to fully perceive the relationship between its body, its actions and virtual objects fostering the usage of gestural interfaces and enabling more direct modeling methods.

Our approach can be summarized by the following hypothesis:

> **Performing 3D Modeling tasks in a semi-immersive environment can be as effective as using existing 2D graphical user interfaces while offering new design perspectives by providing adequate modeling operators combined with novel interaction techniques.**

This thesis contributes to creation new interaction techniques based on two hand gestures and on a new modeling environment taking advantage of stereoscopic visualization and user sketching skills.

To demonstrate our hypothesis, we devise a new modeling tool proposing an innovative 3D user interface to create 3D models such as 3D manufactured objects and architectural scenes during conceptual phases. Our scenario uses a stereoscopic visualization display combined with an interactive surface where one can interact using both hands and model 3D shapes. Combining several 3D input devices, we are able to track user position, including head, arms, hands and its fingers on a surface and in the air using a reduce set of physical artifacts. In this scenario, users are able to create 3D shapes from scratch in a stereoscopic environment. The main modality of interaction are user hands and fingers allowing the user to sketch and perform 3D gestures on an interactive surface or in the air for what they are best suited for. This scenario enables to create, manipulate, edit and visualize 3D content interacting directly with virtual content as it was a malleable physical scale model.

1.1 Scope

Along the last 50 years, Computer Aided Design (CAD) systems have become an indispensable tool for any design process independently of the engineering or architectural domain. This dissertation focuses on 3D Modeling User Interfaces using VE displays. Despite the growing popularity of VEs, they have not yet replaced desktop CAD systems when it comes to modeling 3D scenes. Traditional Virtual Reality idioms are still umbilically connected to the desktop metaphor they aim to replace, by leveraging on the familiar WIMP metaphors. The underlying command language used by existing modeling system does not always map well to the way people learn to conceive, reason about and manipulate three-dimensional shapes.

3D Modeling User Interface is an important topic of the Computer Graphics (CG) research community due to its strong relationship with both the visualization approach and the geometric representation. This work explores Sketch Based Modeling interfaces taking advantage of sketching skills from designers, modelers and architects. Drawings are the more fundamental way to describe 3D shapes and is transversal to any conceptual design phase. In the field of Human Computer Interaction (HCI), this work proposes new interaction techniques using 3D User Interfaces and applying 3D Direct Manipulation to 3D modeling. To design a new user interface, I applied a user centered design approach. I started by studying how users currently create 3D shapes using not only existing CAD tools but also physical representations such as scale models. This was done in the follow-up of my prior research under the scope of several research project at the INESC-ID Visualization and Intelligent Multimodal Interfaces research group. Along 10 years, I contributed in several systems exploring 3D modeling, for example a sketch based interface for free form surfacing BlobMaker [de Araújo and Jorge, 2003] and a sketch based design review tool in large scale display named IMMIView [Jota et al., 2010].

Finally, I was involved along this dissertation in two research projects (Satin and Maximus) funded by the European Commission and three projects (Digital Alberti, MIVIS and CEDAR) funded by the Portuguese Science and Technology Foundation. All these projects addressed different topics related to 3D Modeling in different fields such as Industrial Design, Engineering and Architecture. The Satin project proposes an immersive environment for both the visualization and quality assessment of automotive class A surfaces combined with an haptic device [de Araújo et al., 2010]. While, the Maximus project proposed tools for design review of virtual models using a multi-touch interface for architectural plans manipulation and visualization and a large scale display environment coupled with multi-sensor devices to control the presentation of virtual automotive prototypes. MIVIS focused on modeling approaches based on sketching and multi-touch interfaces for conceptual design phases. CEDAR addressed the problem of collaboration when reviewing 3D models related to engineering projects. Finally, Digital Alberti explored the usage of innovative modeling techniques and virtual environments for the study of Cultural Heritage such as buildings and monuments. The resulting prototypes of these projects enable to experiment several technologies and design tools at the João Lourenço Fernandes Laboratory [de Araújo et al., 2005] at Instituto Superior Técnico (TagusPark Campus). Such experience enabled to work with several professional designers and modelers from several industries as well as architects to better understand conceptual design phases and how nowadays CAD tools are used. To devise the approach presented in this thesis, I collaborated with the Architectural Faculty of Lisbon and the Architectural course at Instituto Superior Técnico analyzing recent advances on using computing tools for procedural modeling and trying to better understand the benefit that VE could bring to support modeling tasks. It also enables to get access to possible final users of our approach which have experimented our system along its implementation and at its final user evaluation.

1.2 Motivation for our Approach

CAD modeling tools are indispensable for any design process and still mainly rely on both 2D Direct Manipulation and WIMP metaphor. Even if such approach presents several drawbacks as mentioned before, why should VEs be explored as an alternative to current interfaces using traditional 2D input devices and what issues should be addressed in order to propose a reliable modeling alternative?

VE displays provide a new dimension compared to traditional desktop displays improving depth perception and mimicking our natural stereoscopic visual perception. This is particularly important for design tasks where the user is looking for describing a 3D shape. Virtual Reality systems are nowadays affordable and do not require costly simulation environment or large scale displays which used to be only accessible to Automotive, Aerospatial or Military Industries or Research Institutes. As presented by [Kasik, 2011], both Computer Graphics and Virtual Reality community agree that we are now starting the Third Wave of Virtual Reality. In particular thanks to the film-making industry, stereoscopic technology are well known to users and even consumer 3D television are now affordable and common. Our approach searches on using the benefits of stereoscopic visualization using a semi-immersive environment to support 3D modeling tasks. Following the Milgram Reality-virtuality continuum definition [Milgram et al., 1995], we refer to Semi-Immersive Environment as part of the Mixed Reality to denominate a visualization scenario where virtual stereoscopic content can be visualized co-located with physical content. Such scenario would enforce 3D Direct Manipulation allowing the user to view its real hands over virtual representations of 3D models.

Compared to 3D Visualization or most of existing 3D Gaming, 3D Modeling is a highly dynamic task without any predefined narrative. The typical design process of aesthetic products itself is an iterative loop mixing physical prototypes and digital models until the shape is fully satisfactory or the time available to design expires, as presented in our prior work [de Araújo et al., 2010]. This loop involves the user to conceive, to model, to both visually and physically evaluate. Then he should define a modification strategy and repeat the process several times.

This dissertation proposes to adapt and extend existing 3D User Interfaces for 3D Modeling. While several interaction metaphors and input devices exist to assist simple tasks as model exploration and navigation, more accessible metaphors are needed to support modeling tasks allowing to take advantage of user gestures without requiring cumbersome input devices.

Existing modeling interfaces mainly rely on WIMP metaphor and both primitive based instantiation and customization. Such approach have been adapted to VE without leveraging the cost of using such dialogs. Our approach aims on proposing a Direct Modeling method to maximize the interaction with the shape representation instead of having to deal with the graphical user interface interruptively. To do so, we will favor mechanism to reduce the need of graphical user interface while leaving the user to spend more time on its design. Frequent operations such as 3D manipulations or content creation or simple editing operations should be easier to access. We believe that such problem could be minored taking more advantage of bimanual interaction gestures to give clues of the intention of the user as suggested by our initial study [Lopes et al., 2011].

Sketching have been explored by several 2D modeling based interfaces taking advantage of user skills and its expressiveness to convey shape. Such benefits should be explored also in a stereoscopic environment using it as a communication tool between the user and the modeling system. However, it should be done mimicking the natural way to sketch on a surface instead of favor 3D sketching in the air. By using sketching, we intend to provide a more accessible modeling tool to users non familiar with traditional CAD interfaces without sacrificing rigor wished by experienced users. Existing sketch based recognition techniques should be extended to be applicable in a 3D environment and adapted to its input devices.

Finally, our approach will rely on freehand gestures to interact and define 3D shapes. Our idea is to take advantage of a physical three dimensional interaction space to interact with 3D content using a stereoscopic visualization. Hands are the main human structures for physically manipulate the environment allowing both fine and gross motor skills. Existing tracking technologies provide non intrusive and accurate solutions to track user hands. While multi-touch has become a well know modality on the surface to most of the users thanks to existing mobile phones, it could be also applied in part in 3D space to take advantage of hand motor skills and its gestural expressiveness.

1.3 Research Goals

Our research goal is to propose a 3D modeling environment using a semi-immersive display as reliable as traditional 2D user interface while offering more accessible way to conceive, edit and manipulate 3D shapes. We will combine on-and-above surface interaction techniques to take advantages of both spaces seamlessly while enforcing plausible gestures to mimic the interaction with physical objects such as scale models.

We will start by analyzing manual conception of scale models while better understanding how current modeling tools are used along the design workflow. Such analysis will enable to better define how to use both hand gestures and sketching in a three-dimensional interaction space. We will also analyze existing interaction metaphors used in 3D VE to define the metaphors which can complement 3D Direct Manipulation techniques to support modeling tasks. Working closely with designers and architects and presenting them existing Virtual Reality technology, we expect to devise a sketching based modeling environment complemented with gestures in 3D space.

Our modeling approach should be accessible to both expert and non expert users to model 3D shapes interactively using a reduce set of operations and fostering direct interaction with 3D models. By expert users, we refer to users which use existing CAD systems on a daily basis to conceive new manufactured objects or architectural scenes. While non experts are users which have to deal with virtual models but are not familiar with CAD modeling in a user perspective such as customers discussing with designers on an existing design project during its presentation. To take advantage of basic skills such as sketching, we will extend sketch based modeling techniques with constraints and recognition techniques to improve and easy the creation of shapes. Such approach should also enable the user to create regular primitives easily such a squares, rectangles, circles and ellipses. Such solution should enable to create complex 3D shapes similar to the one achievable by nowadays CAD modeling systems. Our system will propose simple operators based on face splitting and extrusion similar to the operators presented by Push-and-Pull modeling metaphor combined with sketching recognition algorithms. Constraints should be also explored to improve 3D modeling tasks in space, allowing to easily place information in space or reuse existing content to add further details on 3D shapes. Since the user is interacting directly with 3D model, complex sketching on a face in the air might be difficult beyond drawing straight lines. At any time the user should be able to use the surface as a support to create 2D content. In addition, we should favor the benefits of interacting in 3D space easing spatial manipulations and using interactive surfaces for what they are best suited for.

Regarding the graphical user interface, our goal is to reduce its usage at a minimal level to not distract the user from the 3D model. In addition, we expect to avoid the need of modes such as creating, editing and spatial manipulation modes taking advantage of both hands and interpreting gestures using contextual information from its location on the shape.

Our results will be assessed through the execution of a user evaluations following HCI methodology. While users will be involved during the design and implementation of our approach, we will proceed with questionnaires and the completion of modeling tasks of different difficulty using our approach and an existing modeling system. The evaluation will focus on the usability of semi-immersive environments and its suitability for modeling tasks in particular the creation and placement of 3D objects. This evaluation will be done on top of our prototype which will propose a modeling system with at least similar functionality to existing face based modelers (such as 3DVia Shape[1] or Google Sketchup[2]). In order to assess the naturalness of our modeling environment, the evaluation will be performed with users having a different modeling experience such as students from Computer Science Gaming course or Architecture courses, to experimented users such as professional Architects or Designers. The completion

[1]3DVIA Shape : `http://www.3dvia.com/products/3dvia-shape/`
[2]Sketchup 3D for Everyone: `http://www.sketchup.com/`

of modeling tasks on both systems will enable to evaluate our semi-immersive modeling approach as well as our interaction proposal using both hands. Beyond analyzing the advantages and limitations of interacting in a 3D environment, it will be possible to assess the 3D perception and visualization quality and its adequacy to the task and the scenario. Finally, this method will enable to asses our sketching based solution and the benefits of constraint based modeling approach compare to existing CAD systems. The analysis of this formal evaluation will be used as the basis to validate our hypothesis and assess the interaction metaphors proposed in this thesis.

1.4 Contributions

The research conducted in this dissertation led to the following contribution related with HCI, CG and 3D Modeling:

- *A novel two-hand interaction technique combining the Guiard Bimanual Asymmetric model with On and Above surface interaction applied to semi-immersive environment*. Such technique allows users to interact with an interactive stereoscopic surface and takes advantage of the above interaction space seamlessly to foster adequate gesture based interactions (described in Section 3.2 and 3.3). Our evaluation (Chapter 5) reveals the validity of this approach to reduce the need of graphical user interface and providing a more accessible spatial manipulation to the user than conventional WIMP based interfaces.

- *A novel Direct Modeling technique mixing a Sketch based Modeling interface with a Push and Pull modeling metaphor in a semi-immersive environment* (presented along Chapter 3). Thanks to a reduce set of operator, the evaluation demonstrates that modeling in semi-immersive environment can perform as well as conventional desktop modeling systems. Our approach mimics the physical interaction with scale models without its physical constraints. The Sketch based Modeling approach proposes an on-fly interactive smoothing algorithm to generate 3D piecewise Cubic Bézier curves (Section 4.4). In addition, we propose a beautification algorithm detecting geometric constraints and solving it by energy minimization. Our solution allows to overcome limitations and imprecisions typical of human gestures using constraints based modeling techniques to correct and help the user. Such approach allows to take advantage of sketching as an alternative to primitive instantiation based modeling.

- *A 3D modeling shape representation* allowing to represent complex 3D object representation while supporting interactive free-form shapes combining seamlessly topology, geometry and mesh information. Such representation, described in Section 4.5, extends existing boundary representations proposing interactive modeling ability without any performance penalty and allowing to create shapes from sketches and gestures. It also supports the dynamic generation of curvilinear extrusion presented in Section 4.5.

- *A 3D modeling environment combining a multi-touch surface with a stereoscopic display and 3D input devices* to gather information from user head, arms, hands and fingers on the surface and the space above it. We describe how such combination of 3D input devices can be performed seamlessly in a stereoscopic visualization environment (Section 4.1 and 4.2). While we illustrate the usefulness of such environment for 3D modeling, this semi-environment allows using VEs to be used not only for visualization as it is nowadays but also offering an alternative to traditional CAD systems for design review. Such environment make virtual models more real, inviting the user into an environment with a better 3D perception and allowing him to interact ”physically” with virtual objects such as it is done with physical mock ups.

1.5 Publications

The research performed for this dissertation yielded in several original publications accepted in peer-reviewed scientific conferences and journal, which are listed in a chronological order by date of publication.

1. De Araújo B., Guerreiro T., Fonseca M., Jorge J., Pereira J., Bordegoni M., Ferrise F., Covarrubias M. and Antolini M. *An haptic-based immersive environment for shape analysis and modelling* In JORNAL OF REAL-TIME IMAGE PROCESSING,"Special Issue Improving Display and Rendering Technologies for Virtual Environments", Volume 5, Number 2 (2010),pp. 73-90, DOI: 10.1007/s11554-009-0139-8, 2010.
   Main topic: Haptic based stereoscopic modeling prototype from the Satin project.

2. Jota R., De Araújo B., Bruno L., Pereira J. and Jorge J. *IMMIView: a multi-user solution for design review in real-time*. In JORNAL OF REAL-TIME IMAGE PROCESSING,"Special Issue Improving Display and Rendering Technologies for Virtual Environments", Volume 5, Number 2 (2010),pp. 91-107, DOI: 10.1007/s11554-009-0141-1, 2010.
   Main topic: Sketch based design review application using a large scale display from the Improve project.

3. Lopes P., Mendes D., De Araújo B. and Jorge J. *Combining bimanual manipulation and pen-based input for 3D modelling*. In Proceedings of Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM 2011), pp. 15-22, 2011.
   Main topic: 3D bimanual modeling application using touch and pen devices.

4. De Araújo B., Casiez G. and Jorge J. *Mockup Builder: Direct 3D Modeling On and Above the Surface in a Continuous Interaction Space*. In Proceedings of Graphics Interface (GI'2012), Toronto, Ontario, Canada, May 28 - May 30, 2012.
   Main topic: User interface and interaction techniques presented in Chapter 3.

5. De Araújo B., Casiez G., Jorge J. and Hachet M. *Modeling On and Above a Stereoscopic Multitouch Display*. In ACM CHI Workshop on The 3rd Dimension of CHI (3DCHI): Touching and Designing 3D User Interfaces, Austin, United States, May, 2012.
   Main topic: 3D Modeling techniques and environment presented in both Chapter 3 and 4.

6. De Araújo B., Jorge J. and Duarte J. *Combining Virtual Environments and Direct Manipulation for Architectural Modeling*. In Proceedings of the 30th International eCAADe Conference 2012, Prague, Czech Republic, September 12 - 14, 2012. Awarded with the **Ivan Petrovic prize** for the best presentation by a young researcher.
   Main topic: 3D Modeling techniques applied to Architectural Scenes and combined with procedural techniques.

7. De Araújo B., Casiez G., Jorge J. and Hachet M. *Mockup Builder: 3D Modeling On and Above the Surface*. International Journal of Systems & Applications in Computers & Graphics (C&G), Special Section on Touching the 3rd Dimension, Elsevier B.V., volume 37(3), pp. 165-178, May 2013. *Main topic:* Extended version of Publication 5 including the user evaluation presented in Chapter 5.

1.6 Dissertation Outline

The dissertation is organized into six chapters. Following this introductory chapter, the outline of the dissertation is structured as following.

Chapter 2 presents an overview of existing techniques to represent and generate 3D shapes for 3D modeling applications. We particularly focus on generative and procedural models proposed by the Computer Graphics research field to represent complex architectural 3D scenes. Then we survey the interaction techniques and metaphors applied to 3D Modeling which propose alternatives to traditional mouse and keyboard interfaces. We define a set of criteria to compared sketching based modeling interface and their ability to represent and edit 3D models. We then focus on two hand based interaction techniques before comparing and discussing virtual reality modeling interfaces.

Chapter 3 describes our approach using a semi immersive virtual environment for 3D modeling. Starting from an overview of our direct modeling approach , we describe the different interaction techniques combining the bimanual asymmetric model with on and above surface interaction technique. Finally, we focus on both

3D manipulation methods for 3D scenes and modeling operations provided in our visualization scenario.

Chapter 4 presents our system describing in details our semi-immersive modeling environment from its input devices to the stereoscopic visualization. We describe how 3D gestures and sketch processing are supported fusing user inputs from several input sensors. We present our graphical user interface and how all modalities are combined to offer a consistent user interface. Then we explain how our sketch based interface is devised processing sketches and gestural inputs. Finally, we present our modeling architecture, our internal shape representation and how it is used to generate 3D shapes.

Chapter 5 presents the results achieved by our approach. We performed a formal user evaluation comparing our system with an existing commercial CAD system to assess the benefits and limitations of our approach. We presents the analysis of the this user test and discuss the area of improvements.

Chapter 6 discusses the contributions of our approach and how it suites current 3D modeling work flow fostering the usage of 3D user interface and taking advantages of both innovative visualization scenarios and interaction techniques. Finally, we present our conclusions and propose directions for future works to improve the usage of 3D user interfaces for 3D modeling.

2

Related Work

This Chapter presents an overview of the work related with the topic of this dissertation.

CAD is nowadays an essential tool of any process to manufacture products. However, due to strong basis on mathematical definitions prior to computer systems, most existing modeling tools require a long learning curve and are only accessible to expert users. Several designers can review a project proposal and discuss conceptual ideas. However, it will depend on their CAD modeling skills to realize such changes. The CG community tried to handle 3D authoring problems by introducing new technologies to improve the creation and manipulation of virtual content.

These improvements have mainly focused on two aspects. The first is providing modeling operators with a better understanding of the geometric representation allowing more efficient way to manipulate and create 3D models. The second is improving modeling interfaces in order to propose interaction metaphors more natural and accessible to the user.

In the following sections, we start by surveying alternative modeling techniques such as procedural techniques and new geometrical representations which have been designed to speed-up the authoring of 3D virtual content.Then, we present the state of the art regarding modeling interfaces focusing on sketch based modeling and how virtual reality have been used to support modeling tasks. Finally, we discuss existing 3D modeling interfaces defining comparative criteria regarding 3D modeling capabilities, interaction techniques and visualization scenarios.

2.1 3D Modeling Representation and Generation

The two main classes of geometric representation supported by existing CAD systems are Boundary Representation and Parametric Objects. While boundary representation describes the model as a solid and is usually coupled with constructive solid geometry operations such as Boolean operators,parametric objects define shapes based a set of parameters defining attributes or its domain. Parametric surfaces are an example of this class and can be edited by control point manipulation or trimming. Regarding CG, most of visualization systems rely on polygonal meshes to describe 3D models due to existing graphic hardware. However other representations such point based surfaces or voxels are also used and traditional CAD representation like any mathematical geometric definition can be converted to be visualized by existing computer graphic techniques. A description of most common geometric representation used in CG can be found in [Foley et al., 1990]. In this section, we present alternative geometric representations which do not only describe geometry but how to generate or construct geometry by describing a modeling process. These methods are called as procedural modeling techniques and have been used to create complex structured models such as buildings or trees. We also present other representation and methods used to describe architectural geometry based on domain related primitives (i.e. predefined parametric objects such as windows, doors and walls) or extending existing mesh based representations.

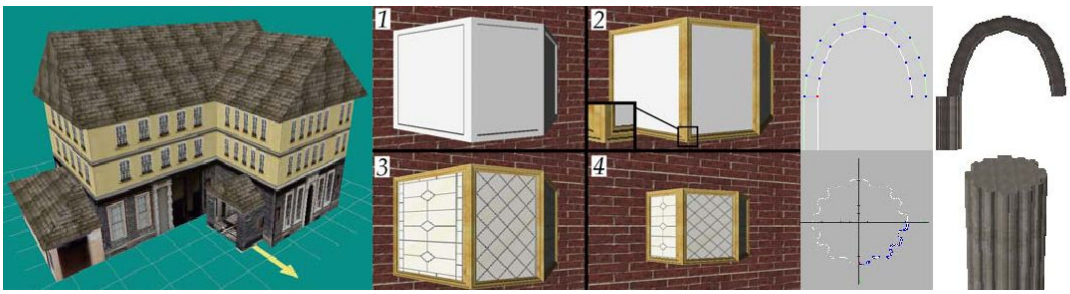


Figure 2.1: Customized Plug-ins for 3D Studio Max implemented under the scope of the CHARISMATIC project [Birch et al., 2001a; Birch et al., 2001b]

2.1.1 Generation of Architectural 3D Scenes

The following research works present procedural modeling tools dedicated to the creation of buildings and architectural scenes. However, instead of using a grammar based definition and allowing the user to script, they present dedicated interfaces which can be customized by the user. The procedural techniques is already embedded in the modeling application knowing domain related elements such as wall, doors, windows and roofs. The CHARISMATIC research project [Browne et al., 2001] is one of the first to handle the problem of both the generation and the visualization of buildings. This project focused on the usage of virtual reality to create cultural heritage attractions. To overcome the time consuming task of content generation, they propose a set of tool based on procedural modeling to complement existing CAD modeling and animation tools such as 3D MAX. In the scope of this project, [Birch et al., 2001a; Birch et al., 2001b] proposed a set of operators based on procedural modeling customized to support architectural modeling. The operators are implemented using a plug-in based architecture into a modeling tool for fast prototyping. Houses and churches can be modeled based on the generation of architectural elements such as windows, doors, columns and arches. For each element, a 2D widget based interface is proposed to the user allowing to customize its different properties. For example, roof of several types can be generated and added to a 3D mass model, windows can be created splitting panels, splines can be used to generate arches and columns and churches can be defined specifying the important structural characteristics. Each operator creates the geometry procedurally and can be applied in the scene or stored to be re-used by future models. Regarding the scene composition and landscape creation, [Flack et al., 2001] present a scene assembler package to configure the terrain, the sky, roads and trees using textures. They focus on methods to specify how the different elements are placed in the 3D place. The user can place procedurally generated objects on the terrain using a 2D map view and rotate them along the vertical axis or dragging the object directly on a 3D view. Houses are generated automatically along roads and trees are placed by defining regions. This project also focus on rendering occlusion techniques and level of detail to enable an interactive visualization of architectural complex scenes [Willmott et al., 2001].

[Laycock and Day, 2003] presents an automatic process to create large urban environment using footprints and LIDAR information. Set of lines are extracted from the footprint creating polygons that are extracted according to the height retrieved from the LIDAR data. Then a straight skeleton extraction is performed on top faces extracting the adequate roof topology. Several roof styles are presented and can be applied to each building according to the skeleton. A different approach is presented also to generate more plausible roofs using rules on top of rectilinear polygons. This algorithm generates building geometry automatically. However, no details or textures are presented on the facades. [Greuter et al., 2003] presents a procedural method to generate pseudo infinite cities in real time. The cities contain geometrically varied buildings that are generated as needed. Only buildings contained in the view frustum are generated and a caching method allows updating the geometry during the visualization. Buildings are generated in a pseudo random way. The city is viewed as a 2D grid and for each cell a seed random number is defined which is used to generate the building. Buildings are created by a random top-down iterative process. At each step a random polygon is inserted to the floor plan definition with a random orientation. Then it is extruded at a random height. This process creates office skyscraper from top to bottom and use a set of ten textures to generate a random city. [Larive and Gaildrat, 2006] describe a wall grammar alternative to generic procedural languages such as split grammars [Wonka et al., 2003]. Instead of using shapes, they use walls as terminal elements and a limited set of rules which reduce the learning time compared to more complex and complete grammar representation as surveyed in [Larive, Dupuy, and Gaildrat, 2005]. Five rules are presented to model building: wall panels representing flat empty walls, bordered walls to put windows, extruded walls to offset wall elements and composed wall layouts such

Figure 2.2: Random generated building using floor definition [Greuter et al., 2003]

Figure 2.3: Wall grammars interface and result proposed by [Larive and Gaildrat, 2006]

as lists or grids. Using Extensible Markup Language (XML) and viewing a tree representation of the grammar, the user can create a building easily. Beyond the rules, three ground templates are proposed to define the basis of the building on oblique terrains. Regarding roofs, a set of styles is proposed to the user which can be adapted to each building topology such as presented in [Laycock and Day, 2003].

[Hahn, Bose, and Whitehead, 2006] present a real-time procedural generation of building interiors. They use a random generation method to define the rooms starting from a predefined seed such as [Greuter et al., 2003]. This is done following several stages using a set of rules. They start by specifying where stairwells and elevators should be located and global attributes such as textures. Then they proceed with the floor division of the region creating uniformly spaced floors. Each floor region is divided creating hallways using a 2D subdivision process, then rooms are created with portals. Only visible regions are visualized and generated on demand. The process is based on random generation. However it perfectly deterministic starting from the same seed point which avoid the need to

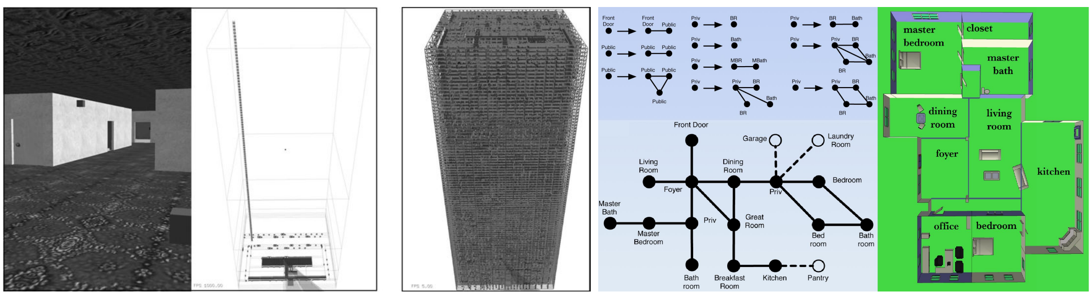


Figure 2.4: Generating Building Interiors: Left) Randomly [Hahn, Bose, and Whitehead, 2006] or Right) using graph based rules [Martin, 2006]

Figure 2.5: Computer-Generated Residential Building Layout [Merrell, Schkufza, and Koltun, 2010]: bubble diagram, floor plans and 3D Model

store the complete building structure. A cache system is used to store previously created built regions to speed up the visualization. Local changes are limited but possible and stored externally. [Martin, 2006] propose a procedural method to generate floor plants of house mimicking average American houses. The process is divided into three steps. First a graph is created using a statistical model on top of a context free grammar and a user defined rule set. The graph starts from the front door and adds public rooms and private rooms representing nodes of the graph and storing the connection of the rooms as edges of the graph. Then rooms are placed in the footprint and a growing process based on Monte Carlo semi deformable growth defines the size of each room thanks to a pressure value assigned to each room. The graph is the basis of the footprint layout and use statistical values to determine the existence and the connection between public and private rooms. These statistical values can be edited by the user, and new rules can be added to support the graph generation. Recently [Merrell, Schkufza, and Koltun, 2010] present a method for automated generation of building layouts using a Bayesian network trained on real world data. The user starts by defining a set of requirements such as the number of bedrooms, bathrooms and approx-

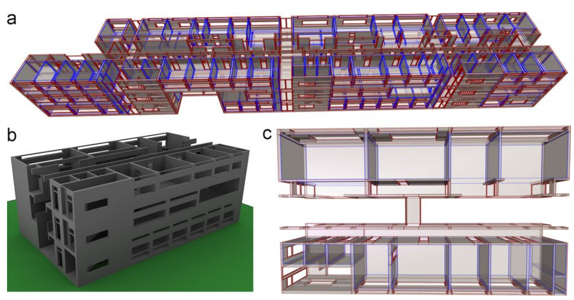


Figure 2.6: Building Interior generated using GML based toolkit with CGA-like grammar operators [Hohmann et al., 2010]

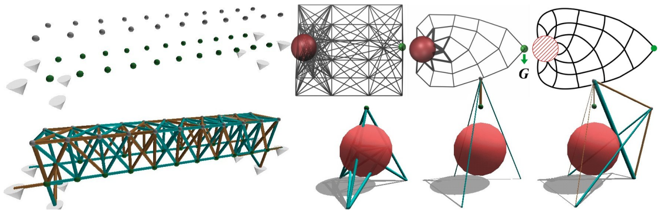


Figure 2.7: Truss structures generated by graph optimization [Smith et al., 2002]

imate square footage. Then thanks to the Bayesian network the architectural program is computed. The Bayesian network was trained using 120 architectural programs defining the adjacency between rooms into a floor distribution. Then an optimization process defines the detailed floor plan for each floor. Finally, a realistic 3D model is created using style templates. Other approaches have use grammars to create interiors. Using this approach, realistic residential building layout are generated considering knowledge of real houses and user requirements. In [Hohmann et al., 2010], a generation of an enriched 3D building model with interiors is presented using Generative Modeling Language (GML). Starting from the manual analysis of a real building footprint, the authors present a method to create a flexible building representation based on GML. Since GML is a imperative scripting language, they propose a set of useful functions which can be used as a toolkit to generate interiors.To better suite the modeling process, an adaptation of concepts such as scope and split familiar to declarative grammars such as Computer-Generated Architecture (CGA) is proposed. Finally several guidelines are presented in order to generate a parametrized building with interiors introducing additional function to create and populate floors with rooms. GML and CGA are discussed in Section 2.1.3 with other general procedural formalisms.

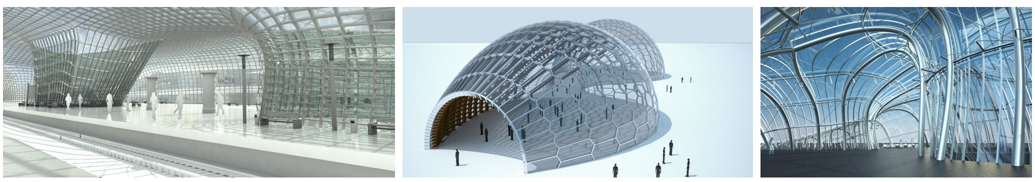


Figure 2.8: Glass strutures generated using conical meshes [Liu et al., 2006], edge offset meshes [Pottmann et al., 2008] and combining booth for curved panekls [Pottmann et al., 2008]

2.1.2 Extending Meshes for Architectural Models

[Smith et al., 2002] presents a process to created structured architectural elements such as bridges and towers. The representation relies on a set of points corresponding to load and anchor points of the structure and a set of obstacle constraints such as spherical or half plane volumes. Using theses points and constraints, a physical based optimization process places free joints and connects the points with beams in order to create a truss like structure. The physical model consider mass distribution and stability to create realistic structures presented in the paper such as different types of truss bridges or the Eiffel tower. [Liu et al., 2006] presents a mesh based representation to design free-form glass structures. The basis of this representation is the conical mesh representation which is sub-type of planar quad meshes. These meshes can be constructed using an optimization process on any kind of mesh and present interesting properties to construct glass structured such as easing the computation of offsets which is the major particularity of multi layer glass and steel structures. This concept was further investigated by [Pottmann et al., 2007] introducing the concept of parallel meshes using edge offsets, i. e. Edge offset meshes. This representation allows mixing quad, pentagonal and hexagonal meshes. It enables to create smooth glass structures. However not all the curvature can be reconstructed using this method. This constraints was partially relaxed by the semi-discrete surface representation [Pottmann et al., 2008]. This last method is an extension of the previous conical and edge offset meshes allowing to represent curved panels. By doing so, they can approximate smooth surfaces by developable strips with smooth boundaries. This research line has shown the reliability of meshes as a geometric representation for architectural free-form surfaces. Several examples of these techniques applied to building exteriors are presented in [Pottmann and Wallner, 2008; Pottmann, Grohs, and B., 2009; Pottmann, Schiftner, and Wallner, 2008].

2.1.3 Procedural Modelling Languages

While previous methods are based on procedural techniques or extend existing representations to handle architectural elements, they rely on customizable procedures or predefined structures. However, several approaches propose new formalisms allowing the user to define the procedural process using a scripting language referred as grammars. Several grammar based procedural modeling techniques have been proposed customized to the architectural domain. However they can be used to represent other types of 3D objects. Existing procedural techniques used in CG to generate geometry are influenced by two main grammar based approaches: Lindenmayer Systems (L-Systems) [Lindenmayer, 1968; Prusinkiewicz and Lindenmayer, 1990] and Shape Grammars [Stiny and Gips, 1972; Stiny, 1975]. While glsplLSYSTEM are a variant of formal grammar to describe the growth process of plant development and the morphology of a variety of organisms, Shape Grammar is a computational approach to the generation of designs mainly used in Architecture.

[Parish and Müller, 2001] presents City Engine an extension of L-Systems to generate cities with streets and buildings. Their extended L-System relies on ideal successor selection to define the correct rule of the grammar to be used, on the definition of global goals from rules and on local constraints from the environment. Thanks to this extended L-System definition, streets can be generated using a growing process based on 2D input data (population, elevation and water areas) and three road patterns (raster, radial and branching). Local constraints are used to process and correct street intersection during the growing process. Then blocks and lots are generated based on the street layout. Lots are then extruded to generate stochastically building geometry based on an L-System definition. Finally facades are textured using procedurally generated layering grids. This work was one of the first to generate realistic cities automatically using procedural modelling. [Wonka et al., 2003] propose split grammars as an alternative to L-System more adapted to building description. Instead of relying on a growing process on top of lines, these grammars use shapes and rules describing how to

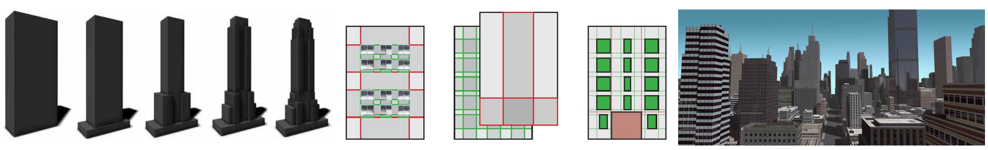


Figure 2.9: CityEngine [Parish and Müller, 2001]: left, using L-Systems to define buildings, middle, layered grids for facade texture, right, generated city example

Figure 2.10: left: using split grammars to define a facade , right: example showing several buildings [Wonka et al., 2003]

subdivide (splits) these shapes into new ones. The derivation process, selects the rule to be applied and propagates attributes between derivations. Attributes can be set manually, copy from the parent to children or defined by a control grammar. The control grammar avoids chaotic design, specifies symmetry, coherence or controls the randomness. It can not only control the attributes but also the rule selection. Split grammars can be used to generate facade buildings and to reuse several styles using a rule database. [Müller et al., 2005] presents an automatic reconstruction method applied for culture heritage. This work combines [Parish and Müller, 2001] and [Wonka et al., 2003] works reconstructing the roman city of Pompei. Using several Geographic Information System (GIS) data as input (building footprint, population, land usage and age area), three pipelines are defined to generate streets, buildings and vegetation. A set of grammars are used to generate the building probabilistically using GIS information. [Müller et al., 2006b] presents an extension of the split grammar named CGA shapes enriched with several operators. Beyond the split and repeat command to subdivide shapes, they

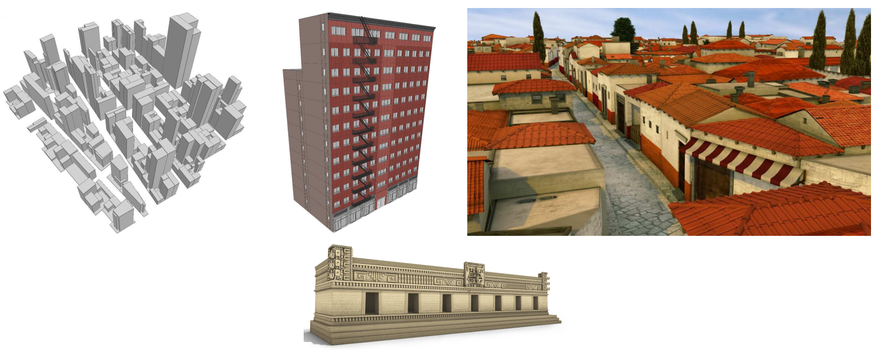


Figure 2.11: Procedural generated examples using CGA grammars [Müller et al., 2005; Müller et al., 2006b; Müller et al., 2006a]

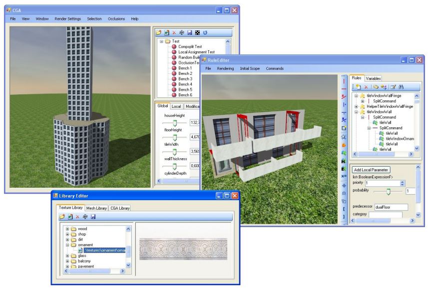


Figure 2.12: Interactive CGA grammar editor using 2D GUI [Lipp, Wonka, and Wimmer, 2008]

introduce component splits which enable to manipulate faces, edges and vertices from a mass model. Geometrical transformations (translation, rotation, scale) can be expressed directly on the grammar and take advantage of the scope definition of a shape which represents the volume of the shape. Derivation is performed based on the concept of priority levels specified to a set of rules. Constraints can be applied thanks to a set of grammar commands such as occlusion queries to invalidate some part or snapping lines to readjust the generation. A commercial tool named CityEngine have been released based on this work. [Müller et al., 2006a] describes the usage of CGA shape to reconstruct a Mayan site using existing building footprints, elevation and archaeological information. They illustrate the flexibility of CGA shape to present how to construct realistic Mayan doors, windows and house variants. A less complete approach was followed by [Huang, Mann, and Cowan, 2009] to create Inca Buildings using an interface which allows the user to select building footprints from a plan and reconstruct 3D buildings using a predefined CGA grammar. [Lipp, Wonka, and Wimmer, 2008] presents a visual editing workflow to create CGA shape grammars without using text. This work proposes a set of 2D Graphical User Interface (GUI) (Building, Rules and Texture attribute editors) combined with a 3D view of the model and a 2D tree where drag and drop of features is supported and rules can be edited by selecting them directly on the 3D model. To discard the need of scripting to use grammar, the authors introduce several concepts enabling direct visual editing, local modifications, semantic and geometric selection and persistence of local modifications. This is done by introducing a set of locators representing variable assignments to model local modifications. These modifications are stored externally to the grammar definition and use to change geometric properties. The locator consists on storing the location of the change in the grammar derivation tree. Three types of locators are introduced: exact instance locator, hierarchical locators and semantic locators. Rules can be reused between grammars thanks to a mapping mechanism where the user is invited to identify the matching non terminal symbols. All CGA grammar operators can be selected from a toolbar and the CGA language was extended to support UV mapping for texturing. The system is easier than text scripting thanks to direct manipulations of geometric features to control splits and crossing 3D selection with 2D tree to identify the rules to be edited. However, the editor is too oriented to CGA grammars and building generation which reduces but does not avoid that users need to be familiar with the CGA grammar writing.

[Havemann and Fellner, 2001] introduces the Combined Boundary Representation (C-BRep) representation for the virtual reconstruction of detailed building geometry. This is an extension of the boundary representation to describe control meshes of subdivision surfaces. Edges can be defined as sharp or smooth changing the behavior of the Catmull subdivision allowing this representation to mix smooth subdivision surfaces with boundary representations. It make also possible to generate models with different levels of detail. Examples are presented using this representation to model several types of building ornaments and windows with oriental and gothic styles. While the conversion of CAD model is done manually, the authors present the more frequent CAD operations of these models (sweeping, extrusion, intersection and measuring) in order to define the basis of a generative modeling technique. The follow-up of this work has resulted on the GML described in [Havemann and Fellner, 2003]. This language is based on the Postscript language and generates geometry using a stack based interpreter. The authors provide a set of mechanism to access and manipulate the C-BRep elements in particular vertexes, edges and faces. They introduce Euler Operators into grammars to give a well-defined access to the mesh and to modify both its connectivity and geometry. These operators update the half-edge data structure which is the basis of C-BReps. Special operators are provided to support the navigation inside the mesh structure to select specific vertexes, edges or faces. Additional implementation details and complex gothic reconstructions were re-

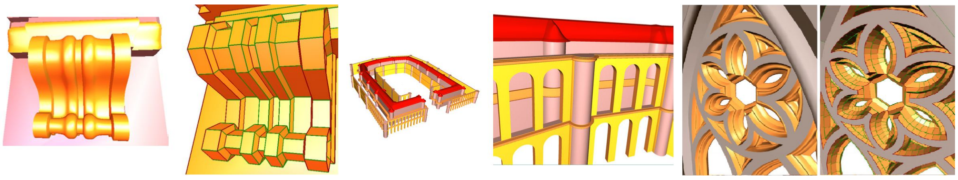


Figure 2.13: Examples of ornaments and models generated using GML language [Havemann and Fellner, 2001; Havemann and Fellner, 2008]

cently presented in [Havemann and Fellner, 2008]. Using the GML language realistic gothic elements such as arch and rosettes are reproduced in [Havemann and Fellner, 2004] and to reconstruct cultural heritage artifacts under the scope of the EPOCH European project. The notation is enriched with other operators such as extrusions, creation and manipulation of circles. While the notation is complex due to the stack based nature of the interpreter, prototypes and macros foster the re-usage of elements allowing to easily model scenes with repetitive components and easy its extension in a modular way. This work was extended by [Berndt, Fellner, and Havemann, 2005] to model houses and buildings introducing new macros to represent building floors and taking advantage of the mix between sharp and smooth surfaces. GML was also integrated with other building generation techniques under the scope of the Charismatic project [Browne et al., 2001; Birch et al., 2001a] to model and render large urban environments of historic and cultural interest sites. In [Day et al., 2003], C-BReps are mixed with a shell based definition to represent building structures such as wall, floors and openings. In [Bein et al., 2009] a sketch based modeling system is presented for subdivision surfaces relying on the GML representation. The system enables the user to sketch stokes and to create 3D models by extrusion and lofting. Then it is possible to edit the underlying boundary representation creating and removing vertexes, edges and faces using a gizmo approach to access the available operators. All user interactions are mapped into GML scripts which are interpreted and executed using C++ macros of the GML operation. A similar approach was used to provide a 3D modeling system of medieval castles for non expert users in [Gerth et al., 2005]. In this paper, the GML is extended in order to be used for the modeling but also to update the scenegraph representation based on OpenSG[1]. Using the same gizmo approach, the C-BRep can be easily edited directly in the 3D environment allow-

[1]Open Source portable scenegraph:`http://www.opensg.org/`

ing to create a libraries of castle parts. This approach follows the construction kit metaphor for modeling. On the other hand, the flexible GML definition of the objects allows the user to customize the existing template in order to create new primitives.

[Finkenzeller, Bender, and Schmitt, 2005] presents a floor plan representation that permits arbitrary floor plan outline to represent building using a hierarchical decomposition of the facade. Floor plan are represented by joining convex polygons using a edge based representation with adjacency information. Floors can be connected representing the several levels of the building. Corners and walls are represented as extrusions on the line strips. They can be subdivided horizontally or vertically. Doors and windows are represented as holes in the walls. For each element, attributes might exist such as the wall or frame thickness or the existence of mortar in the window. Finally several types of roof can be generated using the convex polygons to define its skeleton (flat, pent and gamble roof). All this building components are linked representing the building in a hierarchical way. The authors propose a scripting based approach to create the geometry and a 2D user interface to edit the attributes of each node of the graph. The geometry can be generated on the fly, allowing the editing of the node attributes and building structures. Finally the geometry is represented using a representation suitable for Maya[2] or RenderMan[3] applications. This work was extended in [Finkenzeller and Schmitt, 2006a] to include door, windows and cornices using a LOGO like language to describe the profiles. In [Finkenzeller and Schmitt, 2006b] a more flexible definition of the floor plan is presented allowing representing buildings similar to the Petronas towers. The convex polygons can be subdivided in order to have a more compact evolution of the building outline between floors. The system was complemented in order to provide several styles which can be configured by the user in [Finkenzeller, 2008]. Recently, [Finkenzeller and Bender, 2008] proposes a more complete definition for the semantic modeling of buildings. This method is based on typed graphs defining the semantic representation of the coarse building outline independent of the style. Following this approach the

[2]Autodesk Maya 3D Animation Software: `http://www.autodesk.com/products/autodesk-maya/`

[3]Pixar RenderMan: `http://renderman.pixar.com/`

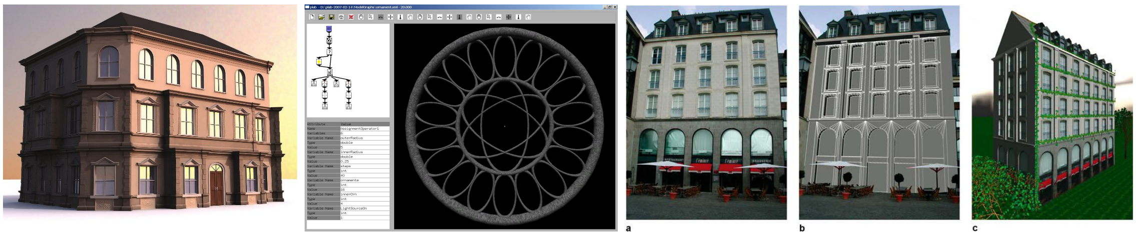


Figure 2.14: left: Realistic building using ornaments by [Finkenzeller, 2008], middle: rosace example using a modeling graph language [Ganster and Klein, 2007] and right: building generated using the FL-System [Marvie, Perret, and Bouatouch, 2005]

user can easily change the building and its style on a semantic level. It allows the designer to create more complex buildings faster than with usual modeling tools. [Marvie, Perret, and Bouatouch, 2005] presents the Functional Lindenmayer System (FL-System) which is an extension of the L-System using functions instead of strings. This representation is suitable to represent buildings, roads or plants where functions act as instantiation of generic objects. They complement the L-System notation with a "for" instruction for iterative generation and a synchronization symbol "!" to work as a barrier for parallel processing. The derivation mechanism to generate the geometry is simple and based on a stack queue. Models are created using Virtual Reality Modeling Language 97 (VRML97) syntax and can be visualized on the fly. [Ganster and Klein, 2007] presents a modeling graph language to generate cities and architectural elements. The modeling is offered through a 2D interface where the user can construct a 2D graph using several operator nodes and configure their attributes. Geometrical primitives and transformation nodes are proposed as well as programming nodes to control the flow execution of the graph. This work is extensible and presents a proof of concept of a visual modeling language easier to use than textual grammars or generative modeling languages.

2.1.3.1 Image based Semi Automatic Reconstruction of Building

[Bekins and Aliaga, 2005] presents a semi automatic method to generate building models from photographs. The user starts by providing a set of 2D images and creates a 3D mass model combining extruded simple 3D shapes. Then us-

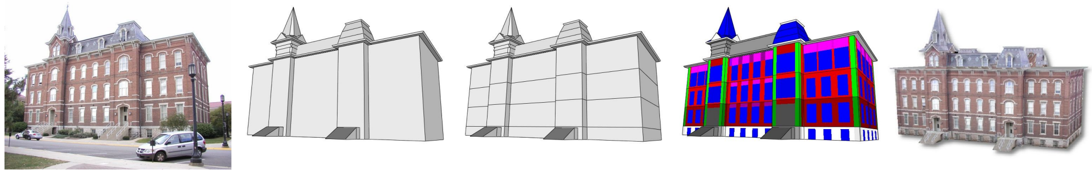


Figure 2.15: The Build-by-Number system [Bekins and Aliaga, 2005]: starting from photographs and a coarse mass model, the building is automatically subdivided into floors with doors and windows and can be redefined to generate alternative building styles

ing a graphical user interface, the user marks occluded faces in each image and edge correspondences between the model and photos. The system automatically matches and aligns the photograph to the 3D coarse model. The user then subdivides the model into features and the system is able to infer regular expression rules automatically using a set of design schemas. Build-by-Number detects faces, floors and it structures the entire model into a set of rules. Then a view dependent texture mapping techniques textures the model homogeneously combining occlusion-free renderings with colour and shading equalization. Thanks to the grammar based reconstruction, the 3D model can be extended and edited easily to create building variants or new buildings. This work generates a rule based description of the model by analysing the user subdivision of the coarse model. However, assumptions on the building structures do not allow reconstructing all kind of building. [Sinha et al., 2008] presents an interactive modeling system using a set of images as input. The system starts by processing the collection of images creating a 3D point cloud, detecting existing vanishing points and lines and estimating the camera pose of each image. Then the user sketches polygons on top of a selected images snapping to existing geometrical information of each image (i.e. points and vanishing lines). These polygons are projected in 3D and are visible on all images according to the camera definition creating a 3D model. Polygonal and modeling operations are proposed such as extrusion, welding, fillet and mirroring updating the mesh of the 3D model. Finally the model is textured according to the images. [Müller et al., 2007] propose an automatic generation of facade using procedural modeling starting from images of facades. This approach performs image analysis to detect existing symmetries (vertical and horizontal) and generates an irreducible facade with the minimal elements of the facade. Then they subdivide faces defining the best split combining the information of all similar tiles. The grammar description of the facade is extracted following [Bekins and Aliaga, 2005] approach and represented using CGA grammars. A 2D matching is performed using the image information to retrieve 3D models of windows and doors from a library previously specified by the user. The method is semi-automatic and the image is re-used to texture the 3D reconstructed model. Depth information of elements of the facade is defined manually by the user. This method is not suitable if there is too much noise in the images due to vegetation or reflections and if a limited symmetry is inferred from images. Only regular buildings can be reconstructed. An alternative image processing techniques is proposed by [Ricard, Royan, and Aubault, 2008] based on the merging of histograms resulting from image analysis algorithms such as edge, background, Hough detectors. [Reznik and Mayer, 2007] presents an automatic reconstruction of building facades from terrestrial image sequence. Using a learning machine, windows are detected and their configuration is analyzed checking if a window is organized into row or column elements or groups. Finally a plane sweeping is performed assigning depth to each facade. Only almost regular buildings are presented and several images from the same building are required.[Van Gool et al., 2007] presents a similar method using a single image. Using a three stage approach, they take advantage of the perspective distortion and the window and balcony regularities existing in buildings to automatically define the depth of each component of the facade. First, features are extracted and grouped, then they proceed with a camera calibration. Finally, the facade is reconstructed solving a minimization problem iteratively and refining the result due to shape a priori. This method only works with strong perspective distortion. An approach similar to [Müller et al., 2007] is followed when the image presents a weak perspective. [Hohmann et al., 2009] presents the status of the CityFit project which aims on reconstruction 80% of the building of Graz automatically. They use LIDAR point clouds and road side photographs as input data to generate grammars and define depth using point cloud information. Currently they generate template grammars using CGA . However they detect limitations of the representation to deal with overlapping elements, split order, unknown neighbors and exception of the grammars. These limitations difficult the usage

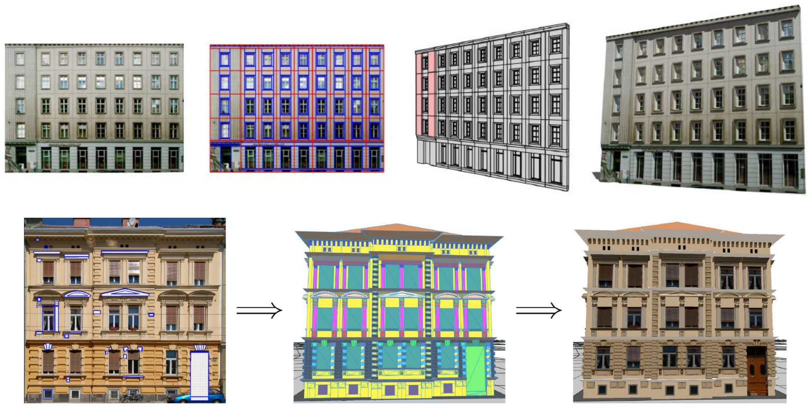


Figure 2.16: Facade reconstruction from image analysis with CGA grammars: top) [Müller et al., 2007] and bottom) [Hohmann et al., 2009]

of CGA and the design of an automatic tool. They propose to adapt the method to use GML in the future, preferring convex polyhedron manipulation instead of the CGA scope definition.

2.1.4 Inverse Procedural Modeling

Procedural Modeling techniques provide expressive and efficient ways to describe and edit geometry. However the key challenge is the definition of the rules requiring in-depth knowledge of the grammar formalism and abstraction capability similar to code programming. Recently, a new research area named inverse procedural modeling is proposing new tools to handle such problem. These tools try to generate automatically procedural rules from an existing 3D model. The advantages of such automatic system are diverse. For example, generation of classes of similar objects would be straight forward by editing its procedural rules. Models could be extended by modifying the internal structure and stored in more compressed way (its rules). Finally, the procedural description could be used as syntactic representation for other algorithms such as image analysis.

[Aliaga, Rosen, and Bekins, 2007] presents and extension of Build-by-Number [Bekins and Aliaga, 2005] which creates more complete grammars (block, face, column). The grammar description relies on the subdivision of the model into ground, a set of floors and a roof. Then each floor is subdivided into faces and faces into columns. Finally columns are represented by terminal symbols representing walls of different materials, windows, doors, trims and ornaments. To define how each top level rule is subdivided, they use predefined regular expressions which are adapted thanks to the detection of patterns on the model images for each face. Regarding the floor definition, the subdivision into faces is based on the topology of the mass model i.e. the existing inner and outer corners. Then faces are matched between them to recognize patterns based on corner orientation, size and resolution. The reconstructed grammar corresponding to the building is presented to the user through graphical user interface using a 2D tree view. By manipulating the tree, the grammar can be edited and can be applied to other mass models allowing to create building with similar styles. Several rendering effects are proposed to stylize the rendering of the model using non photo realistic techniques. An airbrush tool allows to paint the model blending these effects or add elements to the landscape such as trees.

[Yeh and Mech, 2009] presents a method to detect symmetries and curvilinear arrangements in 2D vector art. Thanks to the symmetry detection, the user is able to edit scanned artwork by sliding elements along the detected arrangement path as a group, by changing the distance between elements or by scaling the elements further away from existing ones. Additionally, the user can brush the art work to copy the existing elements and the arrangement between them and create new patterns. Such as other inverse procedural method, the main challenge is the detection of similar and partial symmetric parts of the model. In this paper, the authors follow the approach proposed by [Mitra, Guibas, and Pauly, 2006] which detect symmetries and regularities in point cloud data. The method is based on

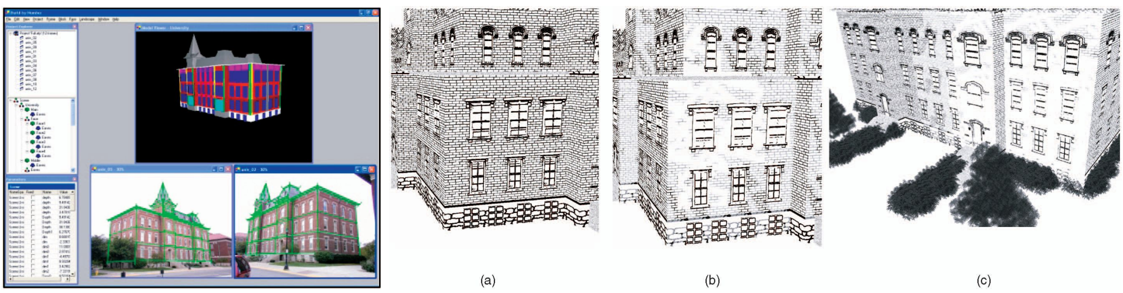


Figure 2.17: Graphical User Interface o f the [Aliaga, Rosen, and Bekins, 2007] system and stylized renderings of the final model

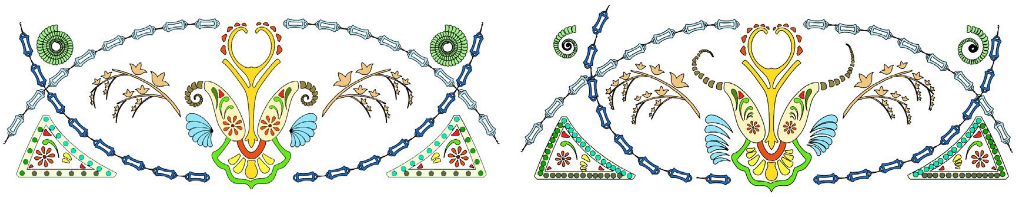


Figure 2.18: left: 2D vector drawing automatically coded as an L-System, right: edited drawing by the user changing L-System parameters [Stava et al., 2010]

clustering in a transformation space. First, for each point a signature is computed to match points between them. Then a transformation is defined for each similar pair of points and valid symmetries are added to the transformation space. Using the mean-shift clustering method, clusters representing most probable symmetries are detected using the transformation space as a voting mechanism. This approach is followed by [Yeh and Mech, 2009] sampling points according to the curvature along the curves of the artwork. Then arrangements, mirror symmetries and scaling are detected along curved paths based on the transformation space values. Finally curvilinear arrangements are un-warped to find more symmetries not only between elements but also between arrangements combining several elements. Thanks to this technique, the user can extend drawings reusing existing symmetries and curvilinear arrangements.

While the previous approach uses an internal representation to represent the arrangement and do not present any rule based grammar to the user, [Stava et al., 2010] proposes a 2D vector editing tool generating a customizable L-System grammar. Based on the same ideas of [Mitra, Guibas, and Pauly, 2006] and [Yeh and Mech, 2009], they use transformation space to detect symmetries in the vector artwork. First they start by finding similar elements which are atomic structures from sequence of connected lines or curves. This is done by sampling each element following [Yeh and Mech, 2009] to quantify the similarity between elements thanks to the space transformation voting system. The resulting elements are grouped into similar groups with their transformation and used as terminal symbols to compose the L-System alphabet. Then using multiple transformation space between pairs of elements they detect similar transformations using [Mitra, Guibas, and Pauly, 2006] approach. Finally procedural rules are generated based on two atomic rules representing sequences and branching. Based on [Pauly et

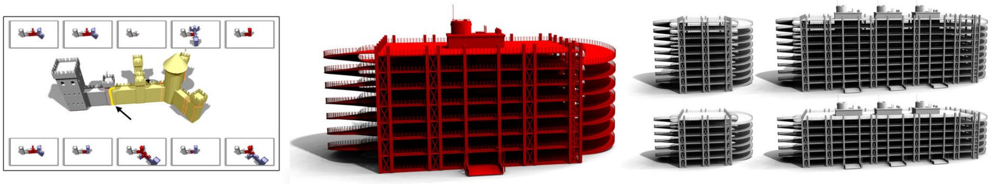


Figure 2.19: left: Incremental Modelling Interface with possible extensions and right: original 3D model and variants obtained by editing the procedural reconstrution [Bokeloh, Wand, and Seidel, 2010]

al., 2008] findings, they know that repetitive structures present themselves as regularly spaced sets of clusters in the transformation space allowing them to search repeated and grid patterns and generate higher level rules by re-clustering the application of recognized rules. These rules are express using the L-System formalism and allow the user to extend and edit the artwork by changing the parameters of the L-Systems. Several 2D examples are presented generating different vector artworks and trees or symmetrizing an existing 2D vector drawing.

[Bokeloh, Wand, and Seidel, 2010] presents an inverse procedural modeling method which finds a set of rules given a 3D geometry. Based on the symmetry detection algorithm proposed by [Bokeloh et al., 2009], the input 3D model is cut into pieces along curves within symmetric areas. By doing so, shape operations are available to the user that maintains local similarity by construction. Such operations are analyzed to construct a shape grammar. Using shape matching, a basic grammar is computed following the Chomsky type-0 formalism. Then they compute a context-free subset of this grammar which is easier to handle by the application. Finally the grammar is improved adding non-context free grid-based replication rules to represent repeated patterns and grids useful for analyzing real-world objects such as buildings. Thanks to this technique, three example modeling tools are proposed. The first tool generates random shape variations of 3D models. The second option provides a semi-automatic modeling tool where the user can extend a partial 3D model into a more complex model interactively by reusing parts of the existing model. Using this tool, several possible partial extensions of the model are suggested to the user. After selecting an extension the model is analyzed and new extensions are proposed to user allowing an incremental modeling. Finally a tool for resizing 3D models is proposed refining a 3D model by editing regular and grid rules in the model. Several examples of buildings are presented based on meshes and point cloud data thanks to the feature line algorithm used for the symmetry detection [Bokeloh et al., 2009]. Compared to other symmetry algorithms based on space transformation voting [Mitra, Guibas, and Pauly, 2006], this algorithm is able to handle larger data sets and rely on a RANSAC (Random Sample Consensus) sub graph matching method to find all symmetric patterns on top of feature lines instead of using the complete data as done by [Mitra, Guibas, and Pauly, 2006].

The research works presented in this section have shown the benefits of 2D and 3D model analysis to generate procedural representation allowing to model new objects from existing ones. Other approaches such as [Gal et al., 2009] have presented the benefits of model analysis to propose more high level shape manipulations. However they do not use procedural rules which make impossible to represent regular patterns. In [Gal et al., 2009], the IWire system is proposed to create geometric variation of input mesh models while preserving its main features and characteristics based on deformation operations. First, input meshes are analyzed to retrieve space set of one-dimensional features named Wires and were initially introduced by [Singh and Fiume, 1998]. These are salient curves of the object that sufficiently describe a shape for editing purpose. Wires are extracted from analyzing sharp edges of the mesh based on sharp dihedral angles or if they are lying on the boundary. Then starting from a seed edge, wires are created chaining the edges by a tracing procedure. Then wires are characterized as planar or non planar, atomic (line, circle, ellipse or polynomial curves) or compound when they are divided into sub-wires due to internal sharp characteristics. Regarding compound wires, the relation between internal sub-wires is also ana-

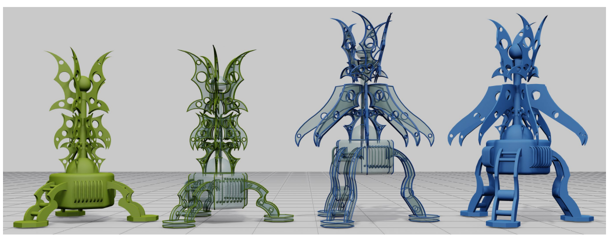


Figure 2.20: I-Wire system [Gal et al., 2009] from left to right: input 3D model, extracted wires, new wire configuration by edition and resulting model

lyze recording equal connection angles, parallel connections and equal lengths. Mutual relations between groups of wires such as symmetries are detected using [Mitra, Guibas, and Pauly, 2006] approach and similar wires are grouped if they exhibit similar properties. Finally the user can deform the model by editing wires, which can be done by dragging part of the surface, deforming an individual wire or by sketch based interaction. While editing, an optimization process propagates wire changes while trying to preserve main features of the shape. This solution presents a natural editing tool based on shape analysis providing high level modeling operators based on deformation.

2.2 3D Modeling Interfaces

2.2.1 2D Sketch Based Modeling Interfaces

Following the research done by Sutherland on interactive pen displays, Sketch Based Modeling (SBM) systems have tried to propose a natural modeling solution taking advantage of user drawing skills and the importance of sketches during the conceptual phase of any design process. In SketchPAD [Sutherland, 1963], Sutherland proposes to use a pen device to interact directly with the display creating 2D and 3D shapes by sketching. However, only during the 90s emerge the first interactive modeling systems targeted for pen device input instead of the traditional mouse and keyboard (such as SKETCH [Zeleznik, Herndon, and Hughes, 1996] and Teddy [Igarashi, Matsuoka, and Tanaka, 1999]). While traditional CAD systems rely on WIMP metaphor, SBMs use sketch to specify gesture commands or draw geometric primitives. At the turn of the millennium, with the fostering of affordable pen-based devices such as digitizing tablets, tabletPC computers and pen based displays, research on SBM proposed several approaches following Sutherland ideas. Recently, existing system have been surveyed by [Olsen et al., 2008] which presents a detailed taxonomy analyzing sketch acquisition, sketch filtering, pre-processing techniques and sketch interpretation issues of existing systems. A more detailed classification is presented in [Olsen et al., 2009] dividing existing systems by their creation method (Iconic, Template, Engineering, Free-form, Multi-View), the surface type (Parametric, Mesh, Implicit, Fair), possible editing operations (Surficial, Additive, Cut, Oversketch, Bend, Constructive Solid Geometry (CSG)) and their interface (Suggestive, Gestural). The remaining of this section will focus on main approaches designed to support the creation of manufactured objects and important mechanisms which might be useful in an interactive virtual reality modeling environment.

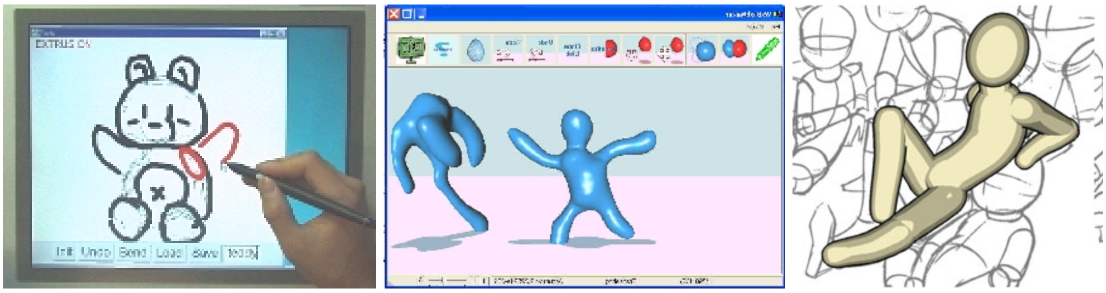


Figure 2.21: Blob-like Sketch based Modelling Systems [Igarashi, Matsuoka, and Tanaka, 1999; de Araújo and Jorge, 2003; Gingold, Igarashi, and Zorin, 2009]

Teddy [Igarashi, Matsuoka, and Tanaka, 1999] is one of the main impact research work on SBM interfaces and have been followed by several systems [Karpenko, Hughes, and Raskar, 2002; de Araújo and Jorge, 2003; Schmidt et al., 2005; Karpenko and Hughes, 2006]. This system allows the user to create free-form shapes by sketching its contour on a 3D perspective view. Then the input stroke is inflated creating a 3D blobby surface. The interface relies exclusively on sketched inputs without the need of any menu to perform the different editing operations. By over sketching the contour, the user can redefine any profile of the shape. Sketching two strokes can create appendices by extrusion. Drawing a straight line on top of the model slices the model into parts. Thanks to a set of gestures, several other operations can be performed: such as merging shapes, creating holes, bending or adding detail on top of the surface. Several systems [Karpenko, Hughes, and Raskar, 2002; de Araújo and Jorge, 2003; Schmidt et al., 2005; Karpenko and Hughes, 2006] have followed this work experimenting several geometric representation such as meshes or implicit surfaces, proposing additional gestures to perform more complex editing. However most of the resulting shapes are simple and based on blobby surfaces making these systems inadequate to create complex models such as a car or a building. While these gesture based approaches provide efficient access to modeling operations mimic natural drawing language, rigorous models such as CAD like shapes cannot be constructed. Recently [Gingold, Igarashi, and Zorin, 2009] try to overcome this lack of rigor using sketch to annotate 2D drawings with depth information order or adding constraints allowing a more controlled shape creation process. However, this solution makes the modeling process less interactive and it is not so well suited for incremental modeling. Finally, the system proposed by the authors only enables to create blobby like shapes.

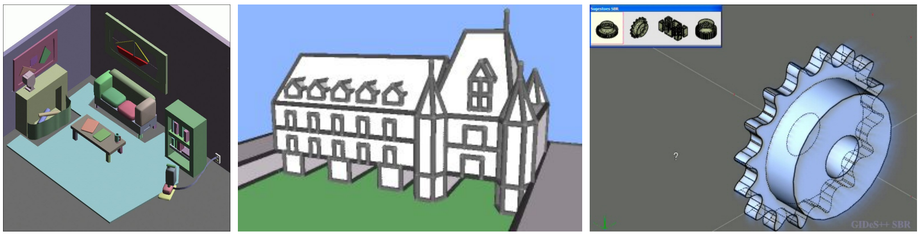


Figure 2.22: Dealing with Sketch Ambiguities: Gestures [Zeleznik, Herndon, and Hughes, 1996], Suggestions [Igarashi and Hughes, 2001] and Expectation Lists [Pereira et al., 2004]

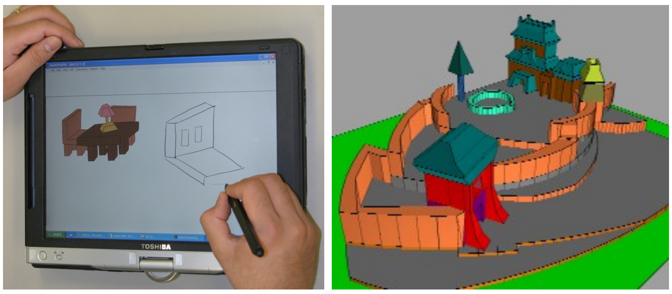


Figure 2.23: SmartPaper:Interpreting Line Drawings [Shesh and Chen, 2004] versus Sesame:Pushing and Pulling Metaphor [Oh, Stuerzlinger, and Danahy, 2006a]

A different approach have been followed to create non free-form shapes based on gestures such as SKETCH [Zeleznik, Herndon, and Hughes, 1996] or using sketch reconstruction such as the work done by [Shpitalni and Lipson, 1996]. Prior to Teddy, the SKETCH system maps CSG operations to gestures allowing the creation of simple manufactured objects such as furniture. Most of geometric primitives are created by extrusion or revolution, for example creating an ellipse in an isometric view and a perpendicular stroke will create a cylinder. Simple shapes such as parallelepipeds, cylinders, cones, pyramids can be easily constructed using line drawings and edited using Boolean operations specified by stroke gestures. This approach is better suited to create manufactured object or architectural elements than Teddy like systems and have been followed by several systems and integrated on some of the existing CAD systems such as Sketchup. Using several gestures, the user can specify a wide number of modeling operations. However, it might be difficult to remember all the commands. CHATEAU [Igarashi and Hughes, 2001] presents an elegant solution proposing suggestions while the user is sketching a line based drawing in a 3D view. Using the mouse, the user can select lines of the model and the system automatically suggests possible geometric extension using a prediction mechanism based on user hints. Depending of the configuration, several suggestions can be presented from a set of twenty suggestions allowing the user to create faces from line loops, drawing planes, rectangles or boxes from perpendicular lines, shape extrusions, resizing, cutting, duplicating or copying measures. The suggestion allows to speedup the modeling and avoids complex menus focusing on model editing. This solution allows creating regular models such as buildings. However, with the increase of the model complexity, the suggestion mechanism might not present the wished operation if the corrected hints are not selected. An alternative solution is followed by the Gides++ system [Pereira et al., 2004] introducing the concept of expectation list and the usage of construction lines to better define shape modification. Unlike previous suggestion mechanisms, the expectation list appears automatically while the user is drawing and ambiguities are detected by the gesture recognition. To avoid complex gestures, floating menus are used presenting additional modeling operations such as Booleans when selecting two objects. This system was extended in [Jorge et al., 2004] to better suite the rigorous needs of the mould making industry. Using simple gestures such as crossing an edge of a face, the user can redefine measures using hand-writing recognition. The system also enables the user to define specific constraints such as edges with equal length or give feedback when perpendiculars, parallelism or concentricity is detected on the model. To allow the creation of more complex shapes, SKETCH gesture like approaches have been combined with line drawing reconstruction methods. SmartPaper [Shesh and Chen, 2004] presents an interactive tool which proposes modeling operations similar to SKETCH. However, they allow the user to sketch more complex line drawings instead of simple primitives such as prisms or parallelepipeds. Using an isometric view, the user can sketch line based shapes which are reconstructed interactively using [Shpitalni and Lipson, 1996] method. This solution encourages natural sketching styles reducing the need of CSG operations to create more complex shapes. [Masry, Kang, and Lipson, 2005] proposes a similar system allowing to create more complex line based drawing such as mass building models in an isometric view. The reconstruction algorithm is more robust than previous work and the approach exclusively rely on sketching discarding other editing operations. This method also enables to use arcs instead of lines allowing the creation of conical shapes. Most of line based reconstruction mechanisms force the user to sketch all the line of the model even the hidden ones. While this approach is adequate for experimented designers, it is not so advantageous for users with limited drawing skills using isometric views. The Sesame [Oh, Stuerzlinger, and Danahy, 2006a] system proposes a drawing approach where sketching is used to define base profiles of shapes and extrusion along straight lines creates 3D shapes. This approach is similar to the Sketchup push-pull metaphor. However the system invites the user to draw the extrusion path instead of directly manipulate the face based representation. To create a mass model of a building, the user can sketch the footprint such as in the paper crossing lines. Sesame automatically analyzes the sketch and approximates it to a set of faces with edges formed by arcs or lines. Then the user can specify the height of the building for each face by drawing a straight line. Compared to other CAD systems such as 3D Max, SESAME has a shorter learning curve and reduces the need of object manipulation and editing operation for similar models.

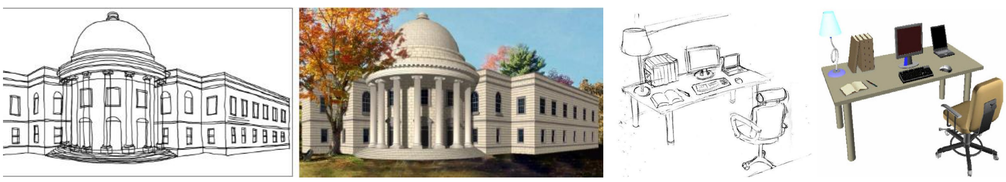


Figure 2.24: Combining Sketch with Retrieval Techniques to compose 3D scenes [Chen et al., 2008; Shin and Igarashi, 2007]

Previous line based reconstruction methods use isometric 3D view and do not scale interactively for complex drawings with fine details. This problem was partially overcome by the Sketching Reality system [Chen et al., 2008] which convert perspective free-hand drawings into realistic looking models. The approach mixes 2D line reconstruction with sketch based retrieval techniques. The perspective camera pose is defined by sketching horizon and vanishing lines. Then, the user sketches the scene while the system classifies if it corresponds to geometric primitives, detailed geometry or textures. If it is a geometric primitive, a reconstruction mechanism based on edge graph and junction classification creates a 2.5D model from reconstructed faces. Compared to other methods, this reconstruction takes advantage of vanishing points to interpret drawings. It allows reconstructing simple models with lines and curves. If a detailed geometry is sketched, the sketch is used to retrieve 3D models from a database. The retrieval is based on several histograms (shape, orientation, length) and is used for windows, doors, columns and ornaments. A similar technique is used to retrieve textures such as wood or rocks allowing to generate a detailed model such as a building or an office with furniture. [Shin and Igarashi, 2007] presents a similar usage of retrieval techniques to support 3D scene composition named Magic Canvas. On top of a well defined 3D view, the user can sketch a 3D scene. Then similar models are retrieved from a database. The models are automatically scaled, oriented and positioned according to the sketching. This is done comparing sketched elements using a centroid Fourier descriptors for global features and a inverse Fourier for local features. For each model stored in the database, descriptors are generated using 16 different views of each model allowing to correctly retrieve the model independently of the pose. A different workflow was followed by [Lee et al., 2008; Lee, Feng, and Gooch, 2008]. They start from real free-hand perspective architectural drawings which are scanned and vectorized creating a polygonal graph representation of the sketch. Then each face and line junction are classified and a 3D reconstruction is generated using an incremental process and the 3D camera definition defined by the user thanks to a bounding box. Compared to other line drawing reconstruction systems, it does not require the drawing of hidden lines, it supports perspective views and it is able to reconstruct curves. The reconstruction is based on hinging angle optimization and cost functions evaluating axis alignment, symmetry, parallelism, co linearity, orthogonality and isometry which better scale to complex drawings than previous approaches.

2D recognition symbol techniques have also been used to create 3D models of buildings. In [Do, 2002], the user is able to sketch the footprint of a building sketching walls in a 2D View. Additionally, symbolic representations are used to specify structural elements such as columns by sketching a circle or furniture elements using predefined symbols. Then the 2D sketch is elevated creating the walls and the structural elements of the house. Finally, recognized symbols are replaced by the corresponding 3D model populating the 3D scene. A similar approach was followed by [Brito, Fonseca, and Jorge, 2005] allowing the user to easily switch between 2D and 3D views. While navigating on the 3D view, it is possible to readjust the position, scale and orientation of the models interacting with 3D widget manipulators. Some attributes such as wall colors or textures can be edited from an existing catalog. Structural information regarding the height of windows, walls or doors can also be changed. While previous approaches focus on interiors and building floor configurations, [Yu and Zhang, 2007] uses a similar assisted approach to elevate 2D sketched footprints into 3D building mass models. First, the user sketches the 2D footprint as it is done on the paper. Then lines and Bezier curves are recognized presenting an overlapped layer with a clean version of the footprint. Finally using an extrusion operator similar to Sesame [Oh, Stuerzlinger, and Danahy, 2006a], the user can control the height of the different components in a 3D view. The system uses a mouse device as input and only requires one button to switch between 2D and 3D allowing creating realistic mass models with curved sections.

Projective drawing is an alternative approach to support conceptual design phases. Instead of aiming on creating or reconstructing a 3D model, this solution allows to create 3D sketches improving the spatial perception compared to traditional 2D sketches. [Kallio, 2005] presents a simple interface where users sketch directly on virtual planes or parabolic shapes located on a 3D view. The virtual plane is represented by a floating grid which can be controlled using the keyboard while 2D stroke inputs are projected using the perspective view definition. [Dorsey et al., 2007] presents the Mental Canvas, a projective drawing system allowing the user to sketch in a 3D view for conceptual architectural design similar to [Kallio, 2005]. However, this system allows creating several virtual planes which are used as canvases. Several drawing modes are available allowing the user to sketch on the 2D View and create a 3D canvas to support the drawing.

The canvas can be activated clicking on its corner and the user can navigate freely on the 3D scene controlling the camera with the keyboard. While this system does not rely on any sketch recognition or reconstruction technique, the user is able to create a 3D representation based on a set of drawings projected on 3D planes. To deal with the overlapping of several canvases, the transparency of strokes belonging to a canvas is modulated depending if the camera is facing the plane. Additionally, a 2D occlusion map can be defined by the user to identify which parts of the sketch should be visible from a designated view.

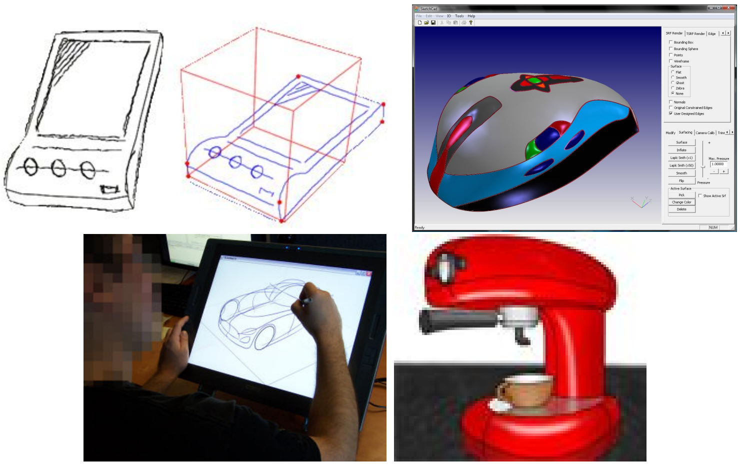


Figure 2.25: Creating free-form CAD surfaces using curve networks: Deforming an existing template [Mitani, Suzuki, and Kimura, 2002; Kara and Shimada, 2006] or inferring curves from sketches using planes [Bae, Balakrishnan, and Singh, 2008; Schmidt et al., 2009]

Line drawing based techniques enable to create face based models with a limited support for smooth surfaces. Using 2D curve fitting techniques, several SBM systems have been proposed to create smooth surface which better fit CAD based industries such as the automotive one. [Mitani, Suzuki, and Kimura, 2002] proposes a solution to create smooth surfaces using sketches from face based templates. Initially, the user creates a line drawing and the system infers the perspective view and reconstructs a 3D face based model using predefined templates. Then, it is possible to redefine the edges using curves by over tracing them. The system uses the perspective information and the template to create a curve network. Finally, each face bounded by curves is approximated by a smooth surface creating the final 3D model. Most of existing SBM interfaces to create smooth surface rely on sketching curve networks. The process of creating

3D planar or non-planar curves in 3D view using sketches is the main issue. [Bae, Kijima, and Kim, 2003] proposes a solution to create more regular planar curves using mirroring techniques. They present a bounding box defining a 3D view and a virtual plane which can be controlled by the user. Then input strokes are projected in order to create symmetric planar curves which are useful for car models. [Tsang et al., 2004] presents a suggestive interface using a cuboid view (orthogonal views) to create 3D curves. In this system the user is drawing directly on a 3D view where he can put blueprints on the three planes of a box. The user can create curves or lines sketching on planes using stylus and control the view using the keyboard. During the sketching, the system automatically suggests extensions of the curve, its closure or extrusions if a profile curve intersects a path curve. Suggestions are presented directly on the drawing unlike [Igarashi and Hughes, 2001] and can be confirmed deleted or ignored using a gesture. Several curve editing operations are available such as cutting and over sketching. The interface relies on gestures to access the different curve functionality depending of the pressure of the stylus. If these lines are on the preloaded blueprint images, they can be used to adapt globally (snap) or partially (glue) the curve sketched by the user using a snake mechanism. [Kara, D'Eramo, and Shimada, 2006] presents a more general version of [Mitani, Suzuki, and Kimura, 2002] approach using templates. The system enables the user to load a drawing and select a wire-frame template of a similar 3D model. Starting with the sketch of a bounding box, the system automatically aligned the template with the drawing calibrating the 3D camera. Then the user can sketch multi-stroke curves on the edges of the template which are interpreted by the system and converted into 3D curves minimizing the curvature energy and the distance to the target curve. By tracing the edges of a face from the template, surfaces can be created using an initial triangulation of the face starting from its centroid. Then using triangle edge swapping and a mesh optimization algorithm, a refined surface is created. The mesh optimization relies on a physical V-Spring based deformation. The surface curvature can be edited by inflating the surfaces, by using the pressure of the stylus to modulate it thanks to an iterative process similar to mesh deformation techniques. This approach has been extended in [Kara and Shimada, 2006] including gesture commands to speed up the selection of editing operations. While previous systems rely on templates, [Bae, Balakrishnan, and Singh, 2008] propose the ILoveSketch system allowing the user to create a network of 3D curves using 2D gestures and an axis widget in a 3D perspective view. Sketched curves are interpreted as 3D using the epipolar method and snapped to existing geometry (i.e. planes or curves). Curves are created on planes which can be controlled by interacting with the axis widget similar to projective drawing systems [Kallio, 2005; Dorsey et al., 2007]. The 3D view can be seen as a virtual sketchbook, which can be zoomed or rotated naturally by the user or automatically depending of the curve orientation for a more suited position. Curves can be created using multi strokes and the system controls its visibility fading oldest curves. [Schmidt et al., 2009] presents a similar drawing technique based on an inference system and the usage of constraints. Following an analytic drawing approach 3D scaffolds are inferred from linear segments. The system is able to deal with 2D drawing ambiguities solving a constrained problem which uses vanishing points and combines several snapping techniques. Complex curves can be reconstructed from a single view providing adequate guidelines and are visible to the user as tangentional points which can be removed by the user if the system is over-constrained. More recently , a similar approach have been proposed by [Orbay and Kara, 2011] allowing to create free form surfaces from curve networks.

2.2.2 Two handed based modeling interfaces

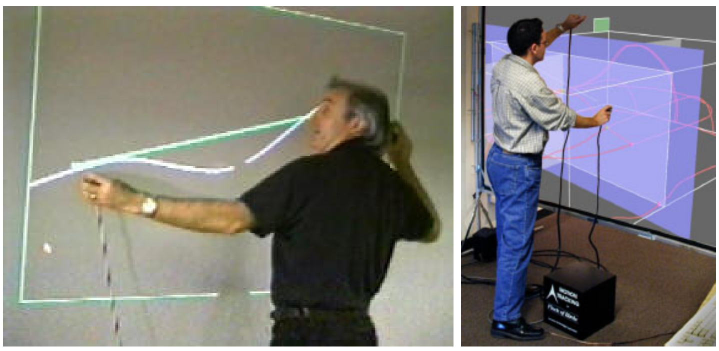


Figure 2.26: Digital Tape Drawing using both user hands [Balakrishnan et al., 1999; Grossman et al., 2002]

Two hand based interaction techniques have been proposed as a natural user interface for 3D modeling. Tracking user hands, these approaches have been mainly used with large mono view projection systems providing interaction metaphors with virtual objects similar to those with physical objects. Initially large projection based system were only affordable for the automotive industry. [Balakrishnan et al., 1999] presents the digital tape drawing which mimics the 1 to 1 scaling physical tape drawing technique used by the automotive industry. Using a rear projection visualization setup and a flock of bird, the user is able to create lines and curves using both hands. Thanks to a button on each tracked sensor, the user starts with the left hand and defines the direction of the curve with the right hand. Then, moving the left hand along the direction mimics the action of putting tape, creating the final 2D curve. The left button allows to stop the drawing while the right button enables to undo and invalidate parts of the curve. Finally, curves can be edited cutting parts of the created curve. This work also shows that the bimanual Guiard model is not fitted to analyze the taping process, since the concept of dominant hand for spatial references and the dominant hand for actions is not respected. While this system allows to create a set of 2D curves, [Grossman et al., 2001] adapt the taping technique to create 3D planar curves using a isometric 3D view and main axis planes to project sketched curves in the 3D view. However to be possible to interact with 3D views, it is necessary to provide some navigation to control the camera and the virtual planes. Following the bimanual model, they use the dominant hand to switch between active planes and the non dominant hand to control the camera. This work have been extend by [Grossman et al., 2002] to construct non planar curves using several planar curve sections. They also enrich the plane definition using parabolic shapes to project curves or using existing curves to define a basis shape. This mechanism provides an alternative way to create arbitrary curves on the 3D scene resulting in more complex wireframe models. Two hand based interaction has been also used to deform 3D models. In [Llamas et al., 2003; Llamas et al., 2005], two systems are proposed name Twister and Bender respectively. Instead of using a large projection display, these systems use traditional monitors combined with two 3D sensors from a flock of bird such as the previous systems. However, they focus on the modeling technique to specify 3D deformations such as twist and bending. Non instrument hands have been also used for 3D modeling using vision based recognition methods. In [Moustakas et al., 2006], the user is facing a projection system and both hands and head are tracked using a stereo camera. Using a multimodal interface, the user is able to sketch shapes using hands, to draw gesture symbols in the air and specify speech commands using a microphone. Several CAD operations are available and combining voice with gestures, the system is able to deal with sketching ambiguities. For example such as some SBM systems, when the user sketches a circle in the air, the recognition system proposes a cone, a cylinder or a sphere. The user can specify which is the correct option using speech commands. Speech can be also used to define actions, for example giving the scale command will allow the user to quantify the scaling using both hands. By saying the create curve command, user can sketch 3D curves in the air moving the hand. This system allows the user to model shapes by creating simple primitives thanks to the gesture recognition which retrieves the primitive. Then it is possible to edit the transformation and deform it. Finally, the user can assembly the primitives to create more complex models. Using both hands is a natural solution to describe shapes. However it is not trivial to fit into the traditional CAD modeling workflow. Recently, [Yi, Qin, and Kang, 2009] proposed to use tracked hands to capture 3D data. Putting several optical markers on the hand and on a stylus, the user can define 3D curves and surfaces. Using hand gestures, the user starts by defining what kind of architectural elements is being defined (wall, door, roof, column).Then the motion of the hand is captured and stored into a 3D file. Finally the data is imported into a CAD system and used to create 3D shapes. This system allows creating architectural models using hands. However this is not done interactively to complement existing CAD modeling tools.

Several approaches use multi-touch devices to provide a tabletop environment where users can interact directly with virtual content using both hands and manipulate 2D or 3D shapes. [Müller-Tomfelde et al., 2010] proposed different methods to use the space above the surface to provide ways of interacting with 2D tabletop content closer to reality. Multitouch surfaces have been complemented with pen devices or other artifacts allowing to take advantage of the Guiard bimanual asymmetric model [Guiard, 1987] to support 2D or 3D editing tasks. Such scenario allows to combine finger or hand gestures with pen devices. [Brandl et al., 2008] proposed a sketching system where the user selects options through touch using the non dominant hand on a WIMP–based graphical interface, while the dominant hand sketches using a pen device [Brandl et al., 2008]. Such a configuration allows to better explore hand gestures proposing richer interaction concepts to represent 2D editing operations such as demonstrated by [Hinckley et al., 2010]. Indeed, this makes switching between modalities easier and allows users to perform a wide range of 2D editing tasks without relying on gestures or GUI invocations. The Conté system [Vogel and Casiez, 2011] extends such idea by proposing a tangible device acting not only as a pen but enabling to place menus and access a wide variety of 2D operations by lying the input device on top of the multitouch screen. The combination of touch and pen have been also used to model 3D content combined with bimanual interaction models.[Lopes et al., 2011] adapted the ShapeShop sketch based free-form modeler to use both pen and multi-touch simultaneously. They found out that the asymmetric bimanual model allows users to perform more manipulations in less time than conventional single interaction point interfaces, which increased the percentage of time spent on sketching and modeling tasks.

2.2.3 Tangible User Interfaces for 3D modeling

Tangible user interfaces allow users to interact with virtual content using physical objects instead of traditional input devices such as mouse and keyboard. While two hand interaction techniques have been the solution of choice to complement large scale displays suiting automotive industry with virtual tap-

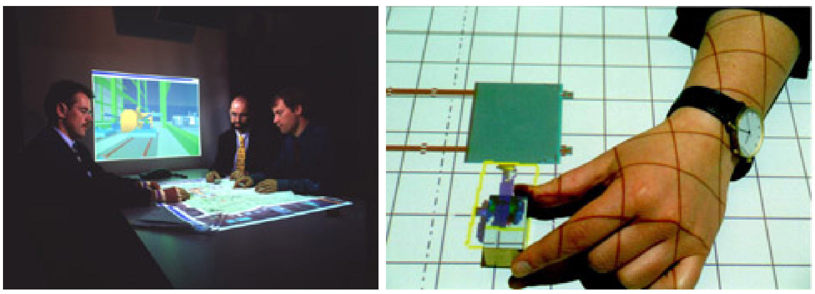


Figure 2.27: BUILT-IT:Interacting with a projected plan using tangible plastic bricks [Fjeld et al., 1999; Fjeld et al., 2002]

ing techniques, tangible user interfaces are ideally design for table top devices. Pioneer work on this area have proposed several solutions to support architectural design review tasks mimicking the natural interaction with building plants and physical mock-ups. Initially, tangible user interfaces were named Graspable User Interfaces and were introduce by [Fitzmaurice, Ishii, and Buxton, 1995]. This work presents the initial ideas of using a set of physical artifacts to interact with virtual content on top of tabletop devices. Several concepts are presented using bricks to move and rotate 2D objects or using two bricks to defines scale objects. Thanks to graspable objects, a rich vocabulary of expression for inputs can be defined showing the benefits of the usage of search approach for future applications such as floor planning, curve spline based definition or object deformation. Theses concepts are illustrated in a 2D drawing application prototype using an ActiveDesk tabletop. However the graspable objects are simulated using a flock of birds. [Fjeld et al., 1998; Fjeld et al., 1999] presents the BUILD-IT environment using a table augmented by projecting digital content on top of it and using physical objects to interact with the scene. This scenario allows one or several users around the table to review a building plan and taking measurements using brick objects. These bricks are transparent plastic objects used on top a projected 2D plan. Putting bricks on top of the table creates building footprints and moving the bricks allows to directly move virtual representations on the plan. Different tangible shapes are available proposing different functionalities, for example the user can use specific bricks to control the plan, moving all the scene or zooming on it using both hands [Fjeld, 2000]. This system is complemented with an additional projection system which presents a 3D view of the scene allowing the user to navigate through the edited scene manipulating another type of bricks. In [Fjeld et al., 2002], several tangible alternatives are proposed such as using physical blocks on a physical 3D mock-up or a paper cardboard representation of the scene. Usability evaluation shows that trial times are reduced using physical 3D mock-up compared to 2D tangible interfaces or interaction with virtual environments. The name of Tangible User Interface has been introduced by Ishii and its research lab at MIT instead of graspable interface. Since the beginning of the century, several research prototypes have been proposed to support architectural design tasks.

Figure 2.28: Tangible User Interfaces for Urban Planning [Ishii et al., 2002] and Lanscape Design [Piper, Ratti, and Ishii, 2002]

[Underkoffler and Ishii, 1999] introduced the URP system supporting the user in several urban planning tasks: building shadow and proximity analysis, light reflection and wind simulation. The setup is an augmented table with a set of I/O Bulb systems which integrate a projector and a camera. Using a set of artifacts with colored patterns, they are able to track them using the camera and project digital content on top of the table with the projector. Buildings are represented by tracked wireframe structure which can be moved by the user on top of the table. The projector augments the table projecting shadows, measuring information, wind simulation data and reflection lights. Using a set of artifact the user is able to control the lighting position and measure the distance between elements. This Luminous Planning Table is one of the first augmented reality environments using a tangible interface to support urban planning and design. This system was used to support MIT's Site and Urban Systems Planning class [Ben-Joseph et al., 2001] projecting digital maps on the table or combined with more elaborate building mockups. [Ishii et al., 2002] proposes an augmented urban planning workbench using the Luminous Table device. On top of a projected Table Top,

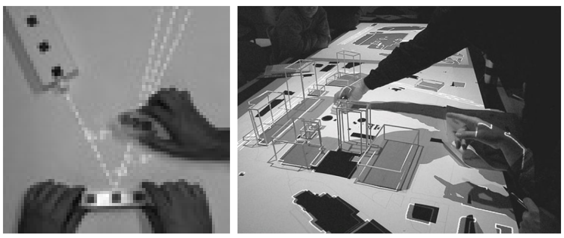


Figure 2.29: Physical objects to take measures [Underkoffler and Ishii, 1999] or to represent buildings [Ben-Joseph et al., 2001]

they can visualize 2D sketches such as building plans and augment them with 3D real object (physical building mock-up) or visualize 3D digital data (model or simulation data). This device was used as an educational tool to support an architectural design course. To interact, physical object representing buildings (Plexiglas model or wireframe) were tracked and used as a tangible user interface. The prototype enables users to control the sun position allowing to deal with building shadows, to visualize 2D simulation of wind flow, load and save 2D and 3D data or models. This is an evolution of the URP environment providing a better perception for key architectural aspects (shadow, shiny and windy areas) mixing 2D and 3D content in the same environment. Users note that technology can be a distracting factor for design and this environment only focus on some aspects of the architectural design. The setup uses TableTop projection and camera tracking, users prefer to use the mouse to interact since the tracking of the physical artifact was not robust enough. A variation of the Luminous Planning Table was created for landscape analysis [Piper, Ratti, and Ishii, 2002; Ishii et al., 2004] named the Illuminating Clay. This scenario augments a clay landscape model. Several thumbnails with different analysis functions and cross sections are placed around the clay model, allowing the user to visualize different types of digital data on top of the model. The system is also able to capture the clay model topology automatically thanks to a laser scanner coupled with the projector. Such scenario enables the user to analyze the model digitally while they change the physical representation by modeling the clay or placing objects on top of it. This scenario can be complemented with a vertical screen next to the working table showing different virtual views of the landscape or existing buildings.

Figure 2.30: Using Physical elements to create digital shapes by combining them [Anderson et al., 2000; Kitamura, Itoh, and Kishino, 2001; Jin, Kim, and Park, 2007]

The previous systems are fitted to be used with tabletop devices and the tangible user interfaces rely on the basis that physical devices are collocated the virtual representation by projecting digital data on them. Other strategies have been followed resulting on new physical objects which act as displays, alternative input devices or physical proxy objects in front of traditional or large scale displays. [Anderson et al., 2000] propose a set of physical blocks for 3D modeling similar to Lego elements. Each brick embed a circuit board which is able to detect which plugs of the Lego are connected to other elements. Such as using Legos, the users can physically assembly the elements creating a physical shape such as a house. Then the output signal from the bricks is processed by a computer detecting the configuration of the assembly and presenting a virtual reconstruction of the 3D shape represented by the blocks. The digital model can then be edited introducing more realistic windows, doors and roofs for example. This work also presents another natural interface based on physical clay modeling. The user can manually model the clay and a digital voxel based representation is created using a scanning device. Other type of block based devices such as the ActiveCube [Kitamura, Itoh, and Kishino, 2001] or Cognitive Cubes [Sharlin et al., 2002] have been presented to support the same virtual prototyping workflow. Recently, [Chen, Wang, and Wang, 2009] combines the usage of tangible user interfaces with procedural techniques for architectural design. [Jin, Kim, and Park, 2007] presents the ARMO system which also uses a set of predefined blocks. However instead of using a generic shape such as a LEGO block, each block has a different shape creating different objects by combining them. After assembling the pieces, the user can manipulate a digital reconstruction of the object in an augmented reality display. The reconstruction is based on vision based techniques which perform edge detection and try to match the model knowing existing available blocks. To easy the segmentation and know the orientation of the object while interacting with it, a marker is put on one of the piece of the object. [Hosokawa et al., 2008] proposes a physical interface to support building design. They present a tangible interface using Radio-frequency identification (RFID) technology which represents interior architectural elements such as walls, walls with doors or windows. These objects are similar to tiles and can be placed on top of a physical grid defining the interior layout of a building floor. Finally a 3D virtual environment is reconstructed from analyzing both the physical grid and tiles. Without the need of any modeling system, the user can define the footprint layout, place furniture putting physical objects on the mockup or even select the wall color or the floor type by putting physical labels with the corresponding material. [Smith, Thomas, and Piekarski, 2008] describes the Digital Foam, an input device for natural sculpting operations mimicking clay sculpting. Using a set of conductive sensors distributed spherically, users can deform a virtual shape by squeezing the device interactively. [Lapides et al., 2006] propose a hardware setup to create 3D curves using a TabletPC on top of an elevator like structure. By doing so, 3D curve creation is easier and more precise that sketching in the air without any support. The user can sketch on top of the tablet and move the TabletPC up and down placing the curve anywhere or defining the profile curve by sketching on the tablet while moving its height. However curves can only be created in the space reached by the tablet and the elevator limiting the user to a restrictive space where the device is located. This problem is solved by the Beyond system [Lee and Ishii, 2010] using a collapsible tool with tabletops. This system proposed tangible interfaces (Pen and Saw device) which are physical devices similar to a pen where both extremities are tracked. When the user sketches on the table, the pressure collapses the tip of the tool allowing the user to define the depth of any drawing input. The user head is tracked orienting the mono view of the tabletop to the user creating an illusion of 3D perception. Using the pen with the dominant hand, the user can sketch 3D curves and surfaces and simple primitives which can be adjusted by extrusion mimicking SketchUp direct manipulation. The primitives are created using the tracked non dominant hand defining a set of gestures. Using hand gestures, the user can specific if he wants

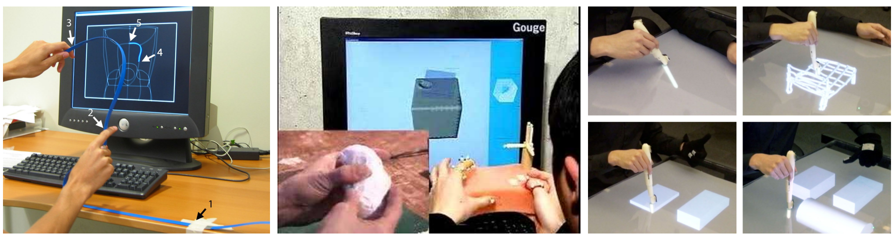


Figure 2.31: 3D modelling tangible tools: a physical tape for curves [Grossman, Balakrishnan, and Singh, 2003], a proxy object for deformations [Sheng, Balakrishnan, and Singh, 2006] and a collapsible pen [Lee and Ishii, 2010]

to draw straight lines, squares, ellipses, move a shape or perform an extrusion. Additionally, the Saw device can be used to trim and cut surfaces. This system proposes a natural user interface using a simple tangible interface dealing with the difficulties of specifying 3D information using a pen like device and allowing to create and move simple scenes. Other systems propose tangible interface to be use in front of a monitor or large scale display as a proxy of the virtual object. [Grossman, Balakrishnan, and Singh, 2003] proposes a tangible user interface based on a physical sensored tape. The user interacts with the tape through a set of gestures to create 3D curves resulting in 3D wireframe model. Additionally two buttons are available using a foot pedals, allowing the user to focus on the usage of the hand to model 3D objects. The tangible tape is used to define the profile of the curves which are projected on the 3D view creating planar and non planar 3D curves. This system combines previous two hand based curve editing techniques from large scale displays [Grossman et al., 2002] with gestures. Since the shape of the tape is tracked, it can be used not only for curve profile definition but also to recognize gesture done with both hand. For example, tapping with the tape on the screen will create the final curve, pinching the tape will define the valid part of the curve or cut it. This idea of using physical devices as proxy of the user interface have been also used by [Bae et al., 2004]. They propose two graspable transparent objects which can be used in front of a retro projected screen to create 2D curves. Unlike taping techniques,the two objects represent the hands of the user to define the curve and not to manipulate a virtual tape. This method is more similar to sketching; the first hand is used as pointing and the second to control. The user can switch between hands the role of each graspable object if wished. The graspable objects are cylinder like transparent shape with a Infrared (IR) reflective material on it. A camera behind the powerwall recognizes the position and orientation while the user is manipulating the object touching the screen. Speech commands are also available to scale objects and different gestures are proposed to edit the curve. [Sheng, Balakrishnan, and Singh, 2006] proposes an instrumented sponge as a proxy object for virtual sculpting tasks. The user is facing a monitor and interact using the proxy object which provides a physical feedback of the editing. The proxy object is tracked as well as the user finger using an optical tracking system. IR Reflective material simulates buttons on the proxy object to control editing options. Additionally, a tracked plastic knife is available proposing a natural metaphor to cut the virtual object. [Abdelmohsen and Do, 2007] presents an architectural modeling system using two physical cubes as a tangible interface. The cube contains a Radio Frequency (RF) transmitter sending which face of the cube is on the top. On each face of the cube, labels are written defining the main command for the left handed cube and the options on the right handed cube. This wireless device acts like a menu hierarchy navigator. The user can create building elements (wall, cube, slab, column, sphere, cylinder), select virtual entities, move and rotate then or control the camera and select predefined view. [Geiger and Rattay, 2008] presents the TubeMouse device to support routing cable design and wiring harnesses using a stereo projected virtual environment. The device is a flexible tube with a metallic spring and each extremity is tracked using an optical tracking system allowing to know the position and shape of the device. Using the device the user can deform the virtual representation of harnesses in a cabling configuration such as it is done in a real scenario. [Jota and Benko, 2011] presents the Stereoblocks system allowing to create 3D scenes using wooden tangible pieces. The interface presents a capture area where the user can stack wooden pieces as building blocks to create 3D shapes. The tangible artifacts are captured in real-time using a Microsoft Kinect depth camera. The live feed is renderer using a stereoscopic visualization mirroring the real scene. Using a handled device, the user can give the order to capture allowing to incrementally create a 3D object.

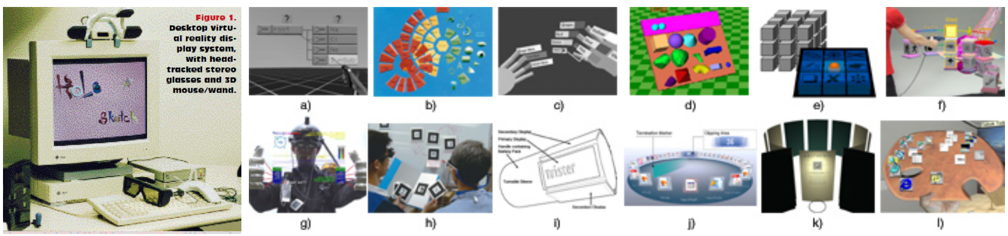


Figure 2.32: HoloSketch Modeling Scenario [Deering, 1996] and Examples of typical 3D Menu User Interfaces [Dachselt and Hübner, 2006]

2.2.4 Non Immersive and Semi-Immersive Virtual Reality Modeling

Non Immersive and Semi-immersive Virtual Reality (VR) environments take advantage of stereoscopic displays to support virtual prototyping tasks. By proposing a better 3D perception, these environments try to make virtual objects more real on top of 3D user interfaces based on 3D input devices. Regarding the visualization, projection based systems or stereo-monitors have been used into different configurations with one or more screen depending of the wished immersivity and on the number of users. The user is required to wear active shutter or passive glasses to correctly see the scene in stereo. Regarding the input devices, 6Degrees of freedom (DoF) flock of birds and gloves [Dipietro, Sabatini, and Dario, 2008] have been the preferred solution for the first VR systems mapping both user hands as a more natural way to interact compared to mouse and keyboards. Nowadays, less invasive input devices are preferred based on optical tracking technology or advanced 6DoF sensors. 3D-Draw[Sachs, Roberts, and Stoops, 1991] is one of the first VR modeling systems allowing the user to design 3D shapes using a stereo non-immersive display. User hands are instrumented with 6DoF sensors and an additional sensor is used to track a stylus to interact with a tablet. This system allows the user to sketch 3D curves and create a curve network as a wireframe view of the 3D model. Some curve operations are proposed to create more controlled curves using constraints and mirror planes. In this system, the modeling functionality is provided by direct manipulation or using the different buttons available on each sensor device. In order to propose more editing operations, [Deering, 1996] proposes an adaptation of traditional WIMP metaphor to 3D VR environments. By using a 3D mouse and a tracked wand, the user can interact with circular menus which are floating in the 3D space. This is one of the first work to handle the problem of 3D graphical user interfaces. By complementing direct manipulation with menus, several CAD editing operations and primitives can be created by navigating through menu hierarchies. Several 3D menus have been proposed along the years mapping the evolution of 2D graphical user interfaces and are presented on most of existing VR system which are surveyed by [Dachselt and Hübner, 2006]. [Kallmann and Thalmann, 1999] presents a virtual environ-

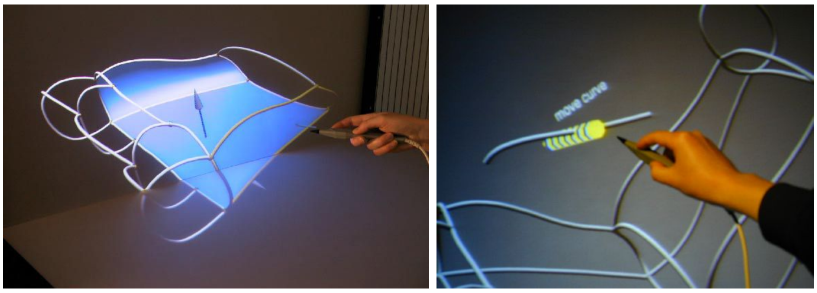


Figure 2.33: The Semi-Immersive FreeDrawer system using a pen with the Toolfinger metaphor to create 3D curves and surfaces [Wesche and Seidel, 2001]

ment where the user can interact with existing objects. Since not all the objects can be manipulated, they present the concept of Smart Objects which are actions that can be triggered for each 3D object depending of the hand user proximity to visual clues depicted on the scene. Instead of using menus,[Nishino et al., 1997; Nishino, Utsumiya, and Korida, 1998] proposes an interactive two-handed gesture based modeling system. The user is wearing gloves which hand motion is analyzed to recognize gestures mimicking clay sculpting. The user can create objects such vases by twisting and bending an initial shape using both hands. The user is standing in front of a large stereoscopic projection system.

[Wesche and Droske, 2000] presents a two handed 3D styling system for free form surfaces. Using a Responsive WorkBench, they offer a mixed reality visualization tracking the user head, glove and a 3D pen. Using such devices, the user is able to control the orientation of the scene with the non dominant hand and sketch curves in the space, creating a 3D network of curves. Finally, surfaces can be generated using loops of connected curves. This work was adapted resulting in the FreeDrawer system[Wesche and Seidel, 2001]. In this new prototype, a L-shaped

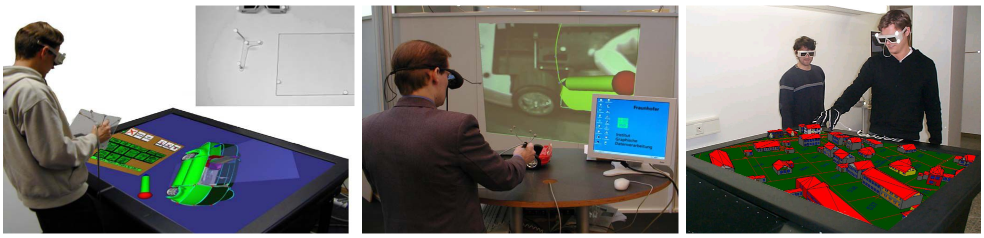


Figure 2.34: 3D Modelling using a WorkBench by sketching, interacting with menus and pointing in the air [Amicis, Conti, and Fiorentino, 2004; Santos et al., 2003; Steinicke et al., 2006]

display was used and more editing operators were proposed allowing the user to drag, select, smooth and sharp curves and surfaces. The freeform representation combined Catmull-Clark, Kuriyama and Bicubic spline surfaces depending of the number of curves used to define the surface. Both system were based on direct manipulation and 3D menus to access to the modeling functionality. In [Wesche, 2003; Wesche, 2004], a solution named ToolFinger was proposed to better support the selection and modification using a 3D pen instrumented with a button to interact with 3D menus. The operations are presented near to the position of the pen, providing an easy access to modeling and manipulation operations helping the user to focus on the editing task. This system was used in [Krause et al., 2004] to propose a three stage conceptual design process using virtual environments. In the first stage the user creates curves and surfaces using FreeDrawer, then surfaces were edited in an elaboration phase applying methods of virtual clay modeling. Finally a physical model can be generated using 3D printing technology. Several other systems [Fiorentino et al., 2002; Fiorentino et al., 2002; Santos et al., 2003; Fiorentino et al., 2004; Amicis, Conti, and Fiorentino, 2004; Fleisch et al., 2004a; Fleisch et al., 2004b; Santos et al., 2005; Fiorentino, Uva, and Monno, 2008] have followed Wesche ideas and propose semi-immersive environment on LShape screens or large stereoscopic walls. These works mix sketching and direct manipulation ability from a tracked pen or artifacts with graphical interfaces using 3D menus. Most of the system uses an adaptation of the Personal Interaction Panel introduced by [Szalavári and Gervautz, 1997] for augmented reality to semi-immersive environment. This is achieved using a transparent PIPSheet [Schmalstieg, Encarnação, and Szalavári, 1999]. By looking to the projection display through it, it creates the illusion that the menu interface is available on the PIPSheet. Using a tracked pen, the user can select CAD primitives and editing them. [Schkolne, Pruett, and Schröder, 2001] describes a different way to create 3D shapes. He proposes a surface drawing system allowing to create organic 3D shapes with the hand and tangible tools. Using a responsive workbench, the user hand position is captured using a tracked glove to sketch shapes floating in the air. The glove is instrumented with a button to start the 3D capture and create shape by moving the hand. The shape has a ribbon like shape which width and profile is defined by the posture of the hand. A set of tracked tangible tools allows the user to grab, move and scale the object using kitchen tongs. Parts of shapes can be deleted using a tracked squeezable object and a magnet tool is available to deform the shape. Thanks to this system the user is able to create complex organic shapes using his hands [Schkolne, 2006]. This approach is able to mix sketching with sculpting metaphors allowing to model objects such as furniture, human faces or body postures easily. Combined with other artifacts, this approach was also used to provide a immersive design environment of DNA molecules[Schkolne, Ishii, and Schroder, 2004]. This scenario allows having a better spatial perception of the shape and provides a natural way to interact with the object by directly manipulating them in space using the hands. 3D menus were also experimented to provide more modeling operations, proposing the tools near to the location where the user is interacting. It allows the user to focus more on modeling tasks. [Kim and Fellner, 2004] presents a hand gesture interaction system for a back-projection wall environment, supporting object manipulation (translation, orientation and scale) and selection tasks through a vision-tracking technique. They propose a stereoscopic passive visualization and they recognize four different hand gestures which are captured using thimble-shaped fingertip markers. These gestures enable the user to interact directly with the virtual object (grab and rotate) or with a virtual cursor (point and pause). In [Kim et al., 2005], this approach was coupled with C-BReps allowing more complex modeling operators and real-time deformation of virtual objects. The different operations mapping the Euler operators were accessible using a graphical menu as modeling modes in order to avoid a large number of gestures to learn. More recently, a similar approach have been proposed by [Perkunder, Israel, and Alexa, 2010] allowing the user to model free

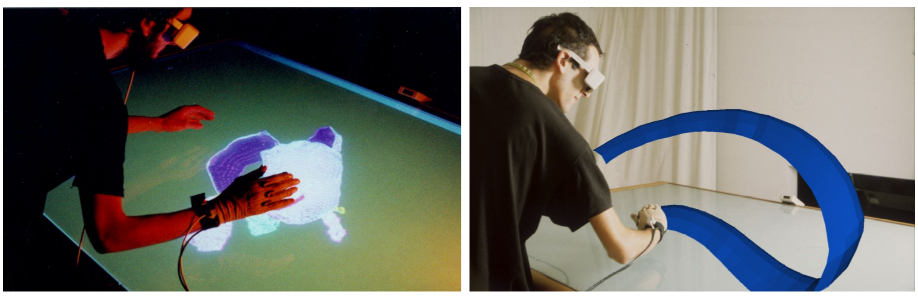


Figure 2.35: Creating 3D free-form shapes by moving the hand [Schkolne, Pruett, and Schröder, 2001]

form shapes by drawing their contour in the air. Using two vertical large screens, [Choi et al., 2005] propose a semi-immersive freeform-surface modeling environment. Using a small tracked wand, the user can sketch 3D curves which are automatically interpolate into Non–uniform rational B–spline (NURBS) curves. Then NURBS surfaces are created by the skinning method using existing curves as sections of the surface. Surface editing is available by deformation and fine details can be created on the surface thanks to a sculpting method. [Steinicke et al., 2006] presents a semi immersive urban city planner. On top of a projection system similar to the responsive workbench, the user is able to visualize GIS data as well as to manipulate 3D buildings using a data glove. This work evaluates several pointing metaphors based on ray casting methods proposing the Improved Virtual point to deal with the lack of precision. This method also includes a haptic vibration to provide feedback when the user hand is colliding with objects. The system is complemented with a 2D interface on a separate desktop allowing to modify some building attributes[Steinicke, Hinrichs, and Ropinski, 2006]. [Keefe, Zeleznik, and Laidlaw, 2007] introduces the DrawingOnAir system allowing to sketch strokes in 3D. Visualizing the scene in a 3D monitor, the user interacts with a phantom haptic devices which is tracked allowing to input hand motion and output forces. User can sketch using the phantom stylus or combining it with an additional 6DoF sensor to create curves by taping method. To activate the sketch, the tip of the stylus acts as a button. Haptic forces are used to create air friction helping the user while sketching in the air. [Bourdot et al., 2010] presents a multimodal immersive modeling environment attempting to integrate existing CAD systems with VR technologies. The user interacts manipulating a tracked artifact and wearing a data glove on the other hand. The multimodal interface allows combining direct manipulation with gesture commands recognized from the glove and speech command from a user microphone. Gestures are used to control the 3D scene and the wand to perform Boundary Representation (BRep) CAD editing such as extrusions. [De la Rivière et al., 2010] presents a stereoscopic tabletop environment combined with a multi-touch surface to visualize architectural scenes. This system allows to navigate and annotate the 3D scene which is rendered a being inside the table. The user interacts using its finger on the multitouch surface acting as proxies over the scene to control the navigation.

Toucheo [Hachet et al., 2011] proposed a fish-tank like setup using a multi-touch surface and a stereoscopic display to assemble 3D virtual archaeological artifacts. Such scenario enables to co-locate user hands and virtual content without occlusions. Using dialogs presented on the multi-touch surface, the user can perform 2D gestures and control the positioning of the objects above the surface.

All the systems presented try to adapt virtual world to create a real-world experiences thanks to 3D interaction techniques. Navigation and Manipulation are the main tasks to support these environments and should be designed in order to be natural for the user. These techniques are discussed by [Kulik, 2009] classifying them into two classes of interaction techniques: reality-based interaction and imagination-based interaction. Reality-based interaction refers to head tracking and natural locomotion techniques, 3D pointing, Direct Manipulation and Graspable input devices. While Imagination-based Interaction refers to concepts such as suspension of naive physics, geometric and motion scaling, automation, magic spells and mode changes. Design of 3D User interfaces should combine the advantages of each class while maintaining the model of interaction consistent and without losing functional expressiveness as mentioned by the authors.

2.2.5 Augmented Reality Modeling Interfaces

As defined by Milgram[Milgram et al., 1995], augmented reality uses HMDs or monitors with camera to enhance the perception of virtual models in real environment. Even with reduced field of view, the usage of HMDs has been followed by several systems to support several modeling tasks while allowing more than one user to interact unlike semi-immersive environments. [Kiyokawa, Takemura, and Yokoya, 2000] presents an augmented reality modeling interface to analyze user awareness in collaborative tasks. Two users are sited in front of a desk, face to face, wearing optical see-through head mounted displays (STHMDs) tracked using a Polhemus flock of bird. Using such environment users are able to create and transform CAD primitives interacting with 3D widgets using both hands. The 3D widgets propose constrained manipulations and alternative representations of the other user are tested to be aware of the focus point of each user. [Gausemeier, Fruend, and Matysczok, 2002] presents a similar solution named

Figure 2.36: The Arthur Project: using a wand to point on 3D models [Broll, Stoerring, and Mottran, 2003] or interact with floating [Penn et al., 2004], or using physical place-holders [Broll et al., 2004]. On the Right: a collaborative CAD learning tool [Kaufmann and Schmalstieg, 2006]

AR-Planning to constraint object placement for a factory planning application. Using a tangible wand tracked by ARToolkit [Kato and Billinghurst, 1999] markers, the user can select 3D objects from a catalog and snapped them in 3D space according to safety rules embedded in the system. Such as AR-Planning tool, several systems have use low cost ARToolkit solution to track artifacts and propose tangible interfaces to interact with 3D models. Using this solution, MagicMeeting [Regenbrecht, Wagner, and Baratoff, 2002] allows several users to interact and review a virtual prototype such as it is done in roundtable meetings. Using a set of tangible interfaces tracked using ARToolkit, users can control the orientation of an object, create section view manipulating a virtual clipping plane and add annotations to a 3D model using a tracked PDA. [Cheok, Edmund, and Eng, 2002] proposes an augmented reality modeling system similar to Schkolne[Schkolne, Pruett, and Schröder, 2001] surface creation method. Tracking the head and the hand in the same way as previous approaches, the user adds control points directly in the air to create parametric surfaces. Then surfaces can be edited by pushing and pulling control points mimicking the traditional CAD approach. [Kato et al., 2003] presents a simple tangible interface for object manipulation in an augmented reality scenario to support city planning tasks. The user wears a HMD display and interacts using a cup with a marker. Using a set of simple gestures the user can pick (covering with the cup), move (sliding the cup) and delete objects (shaking the cup). Other work such as DesignStation [Anderson, Esser, and Interrante, 2003], propose a virtual environment for conceptual design in architecture. Using an HMD and an optical tracking system, they provide an augmented desk mixing real content with digital information such as plans, 3D models and documents. While this system does not propose any 3D modeling functionality, it tries to simulate architectural working environment visualizing real and virtual content in the same environment. [Broll, Stoerring, and Mottran, 2003] presents the initial concept of the ARTHUR project which provides an augmented round table where several users wearing HMD can discuss around a 3D virtual scene. Within this project, several collaborative design scenarios using HMDs are tested as presented in [Broll et al., 2004]. The first scenario is a design review meeting where the user can select and manipulate virtual objects from an architectural scene using a 3D wand device as well as annotate by sketching and interact with 3D menus to select alternative models. The second scenario is a visualization of a simulation of pedestrians and the third presents CAD modeling functionalities using 3D pie menus. Beyond the 5DoF wand, ARTHUR proposes other interaction modalities such as the usage of gesture performed using fingers and the usage of tangible interfaces to interact with menus. The tangible interface is a set of placeholder objects which can be used to manipulate associated virtual objects or interact with menus. Regarding the gestures, static poses of the hand are recognized [Penn et al., 2004] using the head mounted cameras and used to easy the access to a set of commands, such as line drawing, menu opening, copy paste (depending of the number of visible fingertips). A detailed description of the software architecture is presented in [Broll et al., 2005]. While the visualization is an extension of the MORGAN framework targeted to augmented reality, they also present an XML based definition of the interaction techniques using both sensor and actor concepts. Several applications are presented demonstrating the flexibility of the framework. The evaluation of the system, discussed in [Fatah gen. Schieck et al., 2005], focuses initially on the limitations of the different interaction technologies. The user prefers the placeholder objects instead of the gesture input since it presents a more robust tracking solution. On the other hand, they notice drawbacks due to the usage of the HMD camera to track the physical object, since the user was not aware of the correct position when pointing with the place holder object. Finally, three design sessions were performed to evaluate the benefits of the approach on collaborative design review. While the approach seems not to be adequate for modeling task due to the lack of precision, the scenario proved to foster collaboration between users and had increase in the awareness and perception of the virtual scene in particular to better evaluate design options. [Kaufmann and Schmalstieg, 2006] proposes an immersive environment for Geometry Education. Tracking a set of artifacts such as a PIPSheet, used by several semi-immersive systems, two users wearing HMDs can create solid models and parametric surfaces using the Construct3D system. Thanks to a tracked stylus, users can select CAD operations and primitives from a menu presented in the transparent PIPSheet [Szalavári and Gervautz, 1997] and edit a 3D model by direct manipulation using a set of 3D widgets. A preliminary user evaluation was performed showing the educational benefits of such environment to understand 3D modeling operations. [Seichter, 2007] presents a user evaluation of augmented reality and tangible interfaces for architectural planning. In particular, they focus on analyzing its impact on the early stage of the design process and on assessing the communication ability of such tool for decision making. Proposing a scenario where the user is wearing HMD and using tangible tools such a 3D pen or a tangible cube, they evaluate the system with 28 users. The test reveals that augmented reality provides a better spatial perception than traditional media. However, the technology become distracting sometimes for the user. On the other hand the 3D pen generated more discussion due to be a low bandwidth device compared to the tangible cube.

While the previous systems focus on collaboration issues and use HMD, several works have been done expanding sketching modeling techniques in augmented reality scenarios. Most of these systems use monitor based augmented reality and try to present controlled solution to sketch directly in a 3D environment. [Xin, Sharlin, and Sousa, 2008] presents an handled mixed reality 3D sketching system named Napkin Sketch. Using a TabletPC with a camera, the user is able to see an augmented view of a physical napkin. Using the pen and looking through the display, the user can sketch directly on 3D virtual planes as they were real. Since the TabletPC is tracked, the user can control the camera view naturally changing its position regarding the physical napkin. Virtual planes are created with a one-stroke gesture then 2D strokes are projected on the 3D plane according to the perspective camera definition. This system is similar to [Kallio, 2005; Dorsey et al., 2007] 3D projective drawing systems. However thanks to aug-

Figure 2.37: Augmented Reality Sketching based Scenarios [Xin, Sharlin, and Sousa, 2008; Bergig et al., 2009; Hagbi et al., 2010]

mented reality, the navigation and placement of strokes in space is easier. [Bergig et al., 2009] propose a framework for authoring three dimensional virtual scenes for augmented reality based on hand sketching which can be used as a teaching tool for mechanical systems. First, the user sketch mechanical elements in the paper and presents it to a camera. The camera detects the drawing and scans it obtaining a vectorial representation of the line drawing. Using traditional line drawing reconstruction techniques for orthographic view combined with 2D/3D beautification techniques, a 3D model of each connected component of the scene is created. They obtain a scene compound of simple 3D primitives. Then they proceed with a 3D physical simulation using ODE physical engine of geometric elements on top of the location of the paper. Geometric primitives can be redefined, erasing, sketching on the paper and presenting it again to the system. The sketch can be complemented with additional physical property notes such as the mass of elements, existence of friction or object material characteristics which are recognized by the system. This work has been extended by [Hagbi et al., 2010] proposing a more interactive authoring solution for gaming named Sketchaser. Supporting several Augmented reality scenarios such as HMD, monitor or handheld displays, the user interacts with a real-time tracked paper, sketching on it to compose a 3D scene. The paper is tracked thanks to a black boundary such as the previous approach. However, using colored markers, the user can sketch symbols on the paper. A recognition system is used to identify symbols and retrieve corresponding 3D models. Painting an area in green defines a grassy area, coloring it in blue creates a lake and sketching a wavy line defines hills allowing the user to define the gaming landscape on a natural way using the paper as a map. Then buildings and palm trees can be positioned on the 3D scene sketching corresponding symbols. When loading a model, the recognition sketching system specifies the scale according to the size of the symbol. Using such tool, the user can create the 3D environment for an augmented reality racing game. After sketching the symbols for the initial position of two players and a finish line, two users can race against each other on the augmented designed scenario using a keyboard and a mouse. [Bunnun and Mayol-Cuevas, 2008] present the OutlinAR device which is wand with buttons and a camera. This device is used to create model buildings by outlining real boxes captures by its wide fov camera which can be viewed in a monitor based augmented reality environment. To create the building, the user point in the space on real space such as a virtual pointer. The button allows to mark vertexes and the camera pose is estimated from the selected vertexes. All 3D reconstruction is based on an epipolar line based algorithm. The user can then create points, lines, planes and simple box volumes directly in 3D by interacting with the augmented reality device. Planes are created drawing lines and volumes by pushing/pulling faces such as an extrusion. Some simple 3D modifications of the reconstructed objects are provided by moving the 3D model faces. More recently, an automatic method using a camera based vision techniques[Pan, Reitmayr, and Drummond, 2009] proposes a method to reconstruct interactively a simple 3D mockup-up using the ProFORMA system. This system presents an augmented reality scenario which can be used with handled camera and monitor or head mount displays to reconstruct 3D models from existing real objects.

2.3 Discussion

In this section, we present a comparison of the different modeling interfaces focusing on the type of objects that can be modeled by each system and its interaction metaphors. Table2.1 presents an overview of the different sketch based modeling systems. For each system, we define which modeling approach have been used, what kind of visualization is offered to the user and what kind of objects can be modeled using the modeling operator available by the application. Finally, we classify each system regarding the interaction metaphors. As it can be seen most of existing approaches use direct manipulation to complement sketch inputs allowing the user to interact directly with virtual objects over the view and manipulate them. We consider the following set of metaphors for the comparison: Gestures and Symbols representing all systems using a language based on sketch to create shapes, Over Sketch allowing to redefine curves or surfaces by over-tracing the contour or supporting multi-strokes to define curves, Push and Pull which is a special type of direct manipulation allowing to extrude shape such as it is done by SketchUp application, Menus to specify modeling operations, Constraints which can be defined by the user to create more rigorous objects, Projective Drawing which is the ability to project 2D sketch inputs into 3D using virtual planes, Reconstruction which defines systems based on sketch understanding and reconstruction methods, Retrieval which are use to add models from a database to the scene allowing to put re-use objects or define fine details and Templates which are used to constraint the reconstruction of 3D objects based on sketch regarding a specific shape domain. Finally, additional Notes are presented on the last column highlighting some particularity of each system. The existing systems allow representing four main classes of shapes: blobby objects, CSG CAD, 3D lines models and Curves and Surfaces. CSG is the richest modeling paradigm creating complex objects. However due to the large number of operators, they strongly rely on a large set of gestures or menus requiring a long training period. Blobby objects are usually generated by contour inflation focusing more on the over-sketch and gestures than menus. However, they can only represent simple objects and are not reliable to represent manufactured objects with regularities. Regarding 3D line drawing most of them use reconstruction mechanism which is less adapted for incremental modeling systems and with a limited ability of edition. However, they allow representing a large set of objects and are easily adapted to deal with architectural footprints. Finally, 3D curves and surfaces have presented solutions more related with the automotive industry and small manufactured objects thanks to advanced sketch understanding coupled with constraints and templates to infer 3D shapes from 2D data. Constraints, Projective Drawing, Reconstruction and Retrieval have been used to deal with sketching ambiguities making the sketch modeling process more rigorous. However, most of these methods do not scale for all kind of objects or adequate for interactive incremental modeling tasks. We should highlight the usage of constraints, suggestions and expectation lists as the more adequate methods to deal with sketch ambiguities and give feedback to the user of the system sketching ability. Finally gestures mixed with push and pull metaphor have proved to be more efficient methods to access a large number of editing operations instead of the traditional usage of menus and have demonstrated to be more accessible to users. While over-sketching seems to be a natural approach mimicking paper and pencil, it does not seem to be adequate for complex shapes with planar faces. Regarding reconstruction, it usually requires high sketching skills from the user, even more when drawing in perspective compared to gesture based languages or symbols.

Table 2.2 presents the comparison of two hand based modeling techniques and tangible interfaces. For each approach, we present the type of display environment and visualization projection used by the system. We denote what kind of modeling activities are offered to the user. Finally, we define the list of interaction metaphors and modalities as well as particular notes which should be highlighted from each system. Most of the proposed techniques are coupled with large scale displays or table tops fitting two main scenarios: the automotive industry with tapping techniques and architecture with design review tools. Tangible interfaces have proposed solutions to wide variety of scenarios making them specific to each application context. They can be seen as a natural alternative to menus by combining it with gestures. From the existing approach only one system still rely on menus in order to propose CSG operations. Tangible interfaces are adequate to be coupled with tabletops creating physical tools customized for a particular task augmenting the virtual interaction space. We should note that when used as physical displays, tangible interface can only represent a limited class of objects making then unsuitable to general modeling task. However, for design review they have proved to me more accessible to the user than traditional WIMP interfaces increasing the communication between users and helping them to focus on the performed task. These interfaces are good alternative to avoid the excess of menus on a modeling scenario. The more general approach usually instruments the hand of the user proposing new types of pen devices more adequate for sketch based modeling techniques. The main advantage of these interfaces is making virtual content more physical and suitable to naturally map direct manipulation concepts. Finally the physical representation increases the user awareness regarding the status of the scene.

Finally the last set of modeling interfaces uses virtual reality scenarios with stereoscopic visualization or augmented reality. In order to compare these systems, Table 2.3 describe the modeling approach followed by each system, the type of visualization environment to increase the immersivity of the user and its 3D perception of virtual content. Then we classify them regarding what kind of user instrumentation is used in particular head and hands, if the system relies on other devices or artifacts, which tracking technology have been used and finally which interaction metaphor and modalities have been followed and used by the system. All the systems presented track the head of the user to present a correct stereoscopic visualization with high depth perception. This is done tracking the user head on the physical environments or defining its relative position to the virtual scene. Most of the systems are aware of the position of user hands by tracking them directly, by using glove devices or by providing a set of tools such as pen, wands or tracked artifacts. When the system is able to track the hands or the position of gloves, the modeling approaches usually take advantage of gestures and all of them propose direct manipulation to interact with virtual objects. Such as it was verified before, menus are only used when the application propose a large number of editing operations in particular when coupled with CSG modeling techniques. Pens and Wands allow adapting sketching modeling techniques to these semi-immersive environments. However, these are only used for curve sketching and other solutions are preferred to edit and transform virtual content. HMDs propose approaches more adapted to multi-users interacting with the scene and prefer low cost solutions based on markers and vision techniques to track the user and physical objects. We should note that compare to previous approaches, a large number of applications proposes generic modeling systems based on CSG or Surfacing paradigm. However, they usually exhibit the functionality and complexity of existing CAD 2D interfaces with a limited adaptation to the visualization environment. Most adapted modeling functionality take advantage of sketching or tangible interface to propose more natural modeling approaches. While scene manipulation and transformation are correctly handled thanks to direct manipulation techniques, 3D modeling techniques proposed by these immersive environment still do not scale for complex 3D modeling tasks.

Regarding 3D modeling representations and generation methods alternative to existing CAD systems, we have classified existing approaches into five categories. The first represents specific approaches to architectural domain and consists to a set of applications or extensions allowing to model 3D buildings. These approaches have presented dedicated interface to generate buildings which can be used to complement existing modeling applications. They usually focus on particular characteristics of the building layout such as mass models, facades, windows, ornaments and roofs. They propose user interfaces where different parameters can be configured by the user specific to the shape domain. Then based on these parameters a generative process creates the resulting 3D model. Following the same approach semi-automatic methods have been proposed to generate realistic interior layouts based on the analysis of real examples or using a parameterized procedural process. The second category surveyed is extensions to current mesh representations which can be created using any modeling tool and be optimized to meet a set of properties. These solution have been used to represent specific structures such truss structures and freeform glass structures. They do not rely on interactive tools but on an optimization problem starting from an existing mesh or a set of construction constraints. However, they illustrate the scalability of mesh representations as a starting point to define freeform structures. The third category is procedural modeling languages which have been applied to the generation of complex building geometry. These methods rely on textual scripting and the user needs to model the construction process with a set of rules. Several solutions have been proposed, we can highlight the CGA grammars and GML as the more adequate methods to represent buildings. Thanks to the splitting operators and scope manipulation, CGA grammars have present a flexible solution allowing a wide variety of building styles. However since the main paradigm is the split and the extrusion, this solution is not adequate to represent freeform shapes. On the other hand, GML proposes a more functional process which is similar to code programming instead of using rules. This solution requires more programming skills from the user but allows creating a library of components to be re-used as toolkit. One of the main advantages of GML is that all the operations can be map into simple Euler operators allowing generating any kind of manipulation on top of BReps. In addition, this method can handle curves and freeform surfaces since the model is based on subdivision surfaces. Procedural modeling languages are flexible and can be used to represent any kind of object, however they are difficult to define needed scripting and abstraction skills from the user. A long learning period is needed from the user to use grammars as a modeling paradigm.

The fourth category is image based semi automatic reconstruction methods which allow to model 3D buildings based on images. This approach is only suitable to create a virtual model from an existing building and change some parameters. However it cannot be used to model a building from scratch similar to modeling by example techniques. Finally inverse procedural modeling tries to create procedural representations from existing 3D models allowing to capture style and perform complex extensions to a 3D model. Such as the previous method, this solution can only be used to edit an existing model. However, the two last category have illustrated the ability from the system to better understand the 3D model by analyzing its structure to create an internal high level representation to present more meaningful editing operators that traditional modeling CAD. These solutions demonstrate that using procedural techniques traditional 3D models can be more productively edited and changes are closer to the user domain. We believe that this mechanism could be coupled with existing incremental modeling techniques to create high level modeling operations and generate more complex models.

References	Modeling Approach	Viewing Type	3D Object Type	Direct Manipulation	Over Sketch	Gestures Symbols	Push Pull	Menus	Constraints	Projective Drawing	Reconstruction	Retrieval	Templates	Notes
[Igarashi, Matsuoka, and Tanaka, 1999]	Contour Inflation	Perspective	Blobby	x	x	x				x				
[Karpenko, Hughes, and Raskar, 2002; de Araújo and Jorge, 2003; Schmidt et al., 2005; Karpenko and Hughes, 2006]	Contour Inflation	Perspective	Blobby	x	x	x		x		x				
[Gingold, Igarashi, and Zorin, 2009]	Contour Inflation	Perspective	Blobby	x	x	x			x	x				Annotations
[Zeleznik, Herndon, and Hughes, 1996]	3D Lines and Symbolic Gestures	3D Orthogonal	CSG	x		x		x		x				
[Pereira et al., 2004; Jorge et al., 2004]	3D Lines and Symbolic Gestures	3D Orthogonal	CSG	x		x	x	x	x	x		x		Expectation List
[Igarashi and Hughes, 2001]	3D Line Drawing	Perspective	3D lines, polygons	x					x	x				Suggestions
[Shesh and Chen, 2004]	3D Line Drawing	3D Orthogonal	3D lines, polygons					x			x			
[Masry, Kang, and Lipson, 2005]	3D Line Drawing	3D Orthogonal	3D lines, arcs, conics, polygons								x			No editing
[Oh, Stuerzlinger, and Danahy, 2006a]	3D Lines and Symbolic Gestures	Perspective	3D lines, polygons, volumes	x			x	x		x				
[Chen et al., 2008]	3D Line Drawing	Perspective	3D lines, polygons, models	x							x	x		No editing
[Shin and Igarashi, 2007]	3D Line Drawing	Perspective	3D models	x								x		No editing
[Lee et al., 2008; Lee, Feng, and Gooch, 2008]	3D Free Hand Sketch	Perspective	3D lines, polygons						x		x			
[Do, 2002]	2D Line Drawing	2D Orthogonal	3D lines, polygons, volumes, models				x	x						Architectural Footprint
[Brito, Fonseca, and Jorge, 2005]	2D Line Drawing	2D Orthogonal	3D lines, polygons, models					x				x		Architectural Footprint
[Yu and Zhang, 2007]	2D Free Hand Sketch	2D Orthogonal	3D lines, arcs, conics, polygons			x	x	x						Architectural Footprint
[Kallio, 2005]	3D Free Hand Sketch	Perspective	3D lines, curves							x				Virtual Planes
[Dorsey et al., 2007]	3D Free Hand Sketch	Perspective	3D lines, curves	x						x				Virtual Planes
[Mitani, Suzuki, and Kimura, 2002]	Curves Sketching	Perspective	3D curves and surfaces		x					x			x	Planar Curves
[Bae, Kijima, and Kim, 2003]	Curves Sketching	Perspective	3D curves		x				x	x				Planar Curves
[Tsang et al., 2004]	Curves Sketching	3D Orthogonal	3D curves		x				x	x				Suggestions
[Kara, D'Eramo, and Shimada, 2006; Kara and Shimada, 2006]	Curves Sketching	Perspective	3D curves and surfaces		x	x				x	x		x	Non Planar Curves
[Bae, Balakrishnan, and Singh, 2008]	Curves Sketching	Perspective	3D curves		x	x		x	x	x	x			Non Planar Curves
[Schmidt et al., 2009]	Curves Sketching	Perspective	3D curves and surfaces		x	x			x	x	x			Non Planar Curves

Table 2.1: Sketch Based Modeling interface comparison regarding the modeling technique, the viewing mode, the resulting modeled object and the interaction metaphors used by each approach.

References	Viewing Setup	Viewing Type	Modelling App.	Two hand Interaction	Buttons	Tangible UI	Gestures	Sketching	Menus	Notes
[Balakrishnan et al., 1999]	2D Wall	2D Orthogonal	2D Curves	x	x					Taping
[Grossman et al., 2001; Grossman et al., 2002]	2D Wall	3D Orthogonal	3D Planar Curves	x	x			x		Taping
[Llamas et al., 2003; Llamas et al., 2005]	Desktop	Perspective	3D Deformation	x	x					
[Moustakas et al., 2006]	2D Wall	Perspective	3D CSG, Curves, Deformation	x						Voice
[Yi, Qin, and Kang, 2009]	3D Wall	Perspective	Captured 3D curves	x			x			Architectural Domain and CAD integration
[Fitzmaurice, Ishii, and Buxton, 1995]	2D TableTop	2D Orthogonal	2D Plan Review			x	x			Architectural Domain
[Fjeld et al., 1998; Fjeld et al., 1999; Fjeld, 2000; Fjeld et al., 2002]	2D TableTop and Wall	2D Orthogonal and Perspective	2D Plan Review			x	x			Architectural Domain
[Underkoffler and Ishii, 1999; Ben-Joseph et al., 2001; Ishii et al., 2002]	2D TableTop and Wall	2D Orthogonal and Perspective	2D Plan Review			x				Urban Planning
[Piper, Ratti, and Ishii, 2002; Ishii et al., 2004]	3D landscape and Wall	Augmented and Perspective	3D Terrain modelling			x	x			LanscapeDesign and Analysis
[Anderson et al., 2000; Kitamura, Itoh, and Kishino, 2001; Sharlin et al., 2002; Chen, Wang, and Wang, 2009]	Physical Device and Desktop	Perspective	3D Mockup			x				Bricks
[Jin, Kim, and Park, 2007]	Physical Device and Desktop	AR Perspective	3D Mockup			x				Components
[Hosokawa et al., 2008]	Physical Device and Desktop	Perspective	3D Mockup			x				Architectural Mockup
[Smith, Thomas, and Piekarski, 2008]	Physical Device and Desktop	Perspective	3D Deformation			x				Spherical Sensor
[Lapides et al., 2006]	2D TabletPC with Elevator	2D Orthogonal	3D Curves					x		Depth using HW
[Lee and Ishii, 2010]	2D Tabletop	Perspective	3D Curves and CSG			x	x	x		Push Pull and Depth using HW
[Grossman, Balakrishnan, and Singh, 2003]	Desktop	Perspective	3D Curves			x	x			
[Bae et al., 2004]	2D Wall	3D Orthogonal	2D Curves			x	x	x		Voice
[Sheng, Balakrishnan, and Singh, 2006]	Desktop	Perspective	3D Sculpting		x	x	x			Proxy Object
[Abdelmohsen and Do, 2007]	Desktop	Perspective	CSG			x	x		x	Architectural Domain
[Geiger and Rattay, 2008]	3D Wall	Perspective	3D Curves			x	x			Architectural Domain

Table 2.2: Two Hand Based and Tangible Modeling interface comparison regarding the visualization environment, purpose and interaction metaphors used by each approach.

References	Modeling Approach	Viewing Type	Head Tracking	Hand Tracking	Pen or Wand	Artifacts	Other PUI	Magnetic Tracking	Gloves	Optical Tracking	Vision Based	Direct Manipulation	Tangible UI	Gestures	Buttons	Menus	Sketching	Notes
[Sachs, Roberts, and Stoops, 1991]	3D Curves	3D VR Desktop	x	x	x		x	x				x			x			
[Deering, 1996]	CSG CAD	3D VR Desktop	x		x		x					x				x		
[Kallmann and Thalmann, 1999]	Scene Composition	3D VR Desktop	x	x				x				x			x			Smart Objects
[Nishino et al., 1997; Nishino, Utsumiya, and Korida, 1998]	Scultping	stereo wall	x	x				x	x			x		x	x			
[Wesche and Droske, 2000] and others	Curve Sketching and Surfacing	stereo wall	x	x	x				x	x		x				x	x	
[Fiorentino et al., 2002] and SMARTSKETCHES	Curve Sketching and Surfacing	stereo wall	x		x	x	x			x		x				x	x	Contraints
[Schkolne, Pruett, and Schröder, 2001; Schkolne, 2006]	Free form shapes by hand motion	stereo wall	x	x		x		x	x			x	x		x			
[Kim and Fellner, 2004; Kim et al., 2005]	CBREP modelling	stereo wall	x		x					x		x		x				
[Choi et al., 2005]	Nurbs and deformation	stereo wall	x		x					x		x		x			x	
[Steinicke et al., 2006; Steinicke, Hinrichs, and Ropinski, 2006]	GIS Scene Manipulation	stereo wall	x	x		x			x	x		x					x	
[Keefe, Zeleznik, and Laidlaw, 2007]	3D sketching	3D VR Desktop	x		x		x	x									x	Haptic
[Bourdot et al., 2010]	CSG CAD	stereo wall	x	x					x	x		x		x				Multimodal Voice
[Kiyokawa, Takemura, and Yokoya, 2000]	CSG CAD	STHMD	x	x				x				x			x	x		
[Gausemeier, Fruend, and Matysczok, 2002]	Scene Composition	STHMD	x		x						x	x						Contraints
[Regenbrecht, Wagner, and Baratoff, 2002]	Design Review and CSG CAD	STHMD	x			x	x				x	x	x			x		
[Cheok, Edmund, and Eng, 2002]	Nurbs and Control Point Editing	STHMD	x	x		x					x	x	x	x			x	
[Kato et al., 2003]	Scene Composition	STHMD	x								x	x	x	x				
[Anderson, Esser, and Interrante, 2003]	Architectural Conceptual Design	STHMD	x			x				x		x						
[Broll, Stoerring, and Mottran, 2003] and ARTHUR	Design Review and CSG CAD	STHMD	x		x						x	x	x			x		
[Kaufmann and Schmalstieg, 2006]	CSG CAD	STHMD	x			x						x	x			x		
[Seichter, 2007]	Design Review and Architural Planning	STHMD	x		x						x	x	x				x	
[Xin, Sharlin, and Sousa, 2008]	3D sketching	Handheld AR	x		x		x			x				x			x	Projective Sketch
[Bergig et al., 2009]	Scene Authoring	Handheld AR	x				x				x						x	Sketch Reconstruction
[Hagbi et al., 2010]	Scene Authoring	Handheld AR	x				x				x						x	
[Bunnun and Mayol-Cuevas, 2008]	3D Modeling Reconstruction	Handheld AR	x								x						x	3D Reconstruction
[Pan, Reitmayr, and Drummond, 2009]	3D Modeling Reconstruction	Handheld AR	x								x							3D Reconstruction

Table 2.3: Virtual Reality Modeling interfaces comparison regarding the application type, the viewing mode, user or body awereness and the interaction metaphors used by each approach.

3

Our Approach

This chapter describes our modeling approach proposing a new gestural language combined with sketches for semi-immersive visualization environment. Based on the analysis of how physical mockups are usually created, we propose a plausible set of gestures mimicking the physical interaction with mockups such as architectural scale models. To devise our interaction modeling technique, we rely on an innovative user interface combining the bimanual asymmetric model with the notion of continuous space. Our semi-immersive environment enables the co-location between user hands and virtual content fostering direct manipulation metaphors. A set of operators based on a push and pull modeling techniques is mapped into gestures on and above the surface while taking advantage of user drawing skills through a sketching based modeling interface. Finally, the bimanual asymmetric model can be explored to propose constraint based modeling operations naturally.

3.1 Physical Mockup Construction

Our approach is inspired by the architectural scale models which are a popular tool to represent buildings in architectural projects. In order to identify plausible gestures and modeling operators in a semi-immersive environment, we visited the Architectural Faculty of Lisbon to study both manual and semi-automatic construction methods of both physical mockups and scale models. In addition, this section presents the review of several user interfaces envisioned by SciFi movies since they could be seen as a good way to foresee usage and impact of innovative 3D technologies.

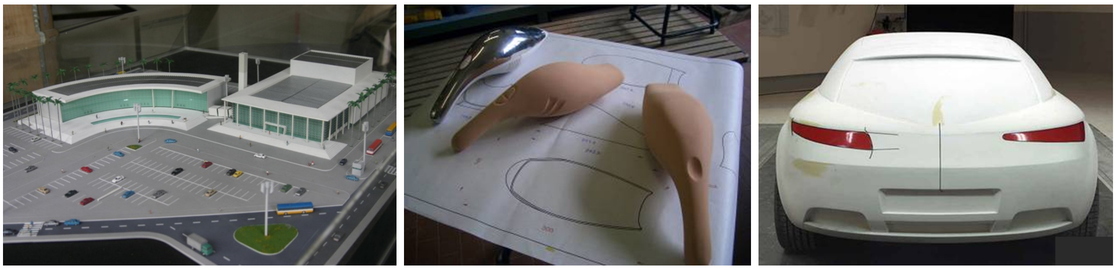


Figure 3.1: Example of Scale Models and Mockups from different fields: Architecture, Home Tool Design and Automotive Design.

3.1.1 Creating Real Scale Models

Scale models allow both expert and student architects to visualize their design in 3D space before its construction. Compared to other medias such as rendered virtual models, plans or video walk-through , its main advantage is the ability for the user to view the model from its own point of view giving a better spatial perception during design phases. While it is been used by students as an exercise of construction, it is a powerful tool for expert architects to better understand proportions and shapes as well as to convey their design ideas to others. Scale models provide an abstraction of the reality while simplifying the project to its essential and to a scale allowing its analysis. Scaling of architectural mockups may vary depending of representing urban areas, landscapes, a specific building, interiors or even a specific detail of the project. Scale models are not exclusive of the architectural field, since several industries use physical mockups in their design process when it come to manufactured objects. We use the term physical mockup as a broader term referring to any physical representations of objects independently of its scale which are used to preview aesthetic or physical properties during design stages. As depicted by the several examples of Figure 3.1, mockups can even been used at a one to one scale as prototypes from simple objects such as cutlery or tableware to complex engineering projects such an automotive.

Architectural scale models such as other kind of physical mockups can be nowadays created automatically from virtual models using 3D printers, computer guided milling machines or mold making. However, these technologies are expensive compared to both the scale and the detail that they can reproduce and they are usually less accessible to most users. In addition, they completely

Figure 3.2: Example of students creating a scale model manually using cardboard.

lose the learning potential provided by manual mockup construction. To devise our modeling method, we are more interested by semi-automatic and manual construction methods which could benefit to human computer interface design and CAD system modeling functionality. Mockups can be made of several materials such as paper, cardboard, polystyrene, wood, metal, plastic, modeling clay, plaster and other accessories such as sand, stones, plants or figurines. We focused on the usage of low-fidelity materials such as paper, cardboard and polystyrene. These materials are among the more popular materials for architectural students when it come to construct scale models manually.

Both manual and semi-automatic construction methods involve the usage of a wide variety of tools such as cutting tools, rulers, glues and pencils as it can be seen in Figure 3.2. Cutting planar shapes is the most basic action to construct volumetric shapes from paper and cardboards. It can be done semi-automatically using a laser cutting machine printing digital vectorial sections of building or landscapes. Or the user can sketch planar shapes with a pencil and ruler on the paper then cut them manually using for example a retractable blade knife. These planar shapes are then stacked or assembled to represent volumes by gluing them together. Alternatively volume shapes can be created by molding or using milling tools. Basically, complex physical scale models can be built simply by sketching, cutting and gluing 2D planar shapes.

While scale models are a good media to convey design ideas, their static nature limit their usage when it comes to discuss project revisions. They are mainly used as a presentation tools and both designers and architects need to rely on other solutions such as sketches and notes to describe future changes. Even crude scale models are difficult to be changed or reused to illustrate a different design idea. On the other hand, virtual models and CAD modeling systems allow to represent these changes but require skilled users on using their interface. Combining virtual environment with gesture based interface could leverage such issue by mimicking the physical interaction with scale models.

3.1.2 Modeling Interface Visions from SciFi Movies

Editable physical mockups or visualization of virtual mockup as they were real, have been a topic of interest in Science Fiction movies. During the last years, the constant evolution of user interfaces have intrigue researchers to compare and foresee new technology usage by analyzing SciFi movies. In particular, it is not uncommon for users to compare existing natural user interfaces with what they have seen in movies such as the [Minority Report, 2002] motion picture where the main character use both gestures and multitouch interfaces to navigate into a video database using its hands and voice commands.[Schmitz, Endres, and Butz, 2007] have presented a survey of human computer interaction designs in SciFi movies along the last decade. They compare theses designs to the state of the art in human computer interaction techniques while discussing both challenges and drawbacks. Recently,[Shedroff and Noessel, 2012b] describes a set of interface design lessons extracted from Sci-Fi movies. This research paper has resulted in a book [Shedroff and Noessel, 2012a] covering several topics related to interface design as well as their impact in assisting basic human activities such as communication and learning.

While several physical display technologies are discussed going further than existing holographic technologies and presenting animated virtual models which can be touched physically by the user, some 3D modeling interface using gestures are also presented. One of the interesting examples is presented in the Iron Man movie sequel [Iron Man, 2008; Iron Man 2, 2010; Iron Man 3, 2013] released in 2008 and 2010. The first movie illustrates a Holotable device used by the main character to review an armor design as depicted by Figure 3.3. The interface presents a graphical user interface on the Holotable surface and a 3D projection above it showing a virtual model. As illustrated by the two first images of Figure 3.3,the

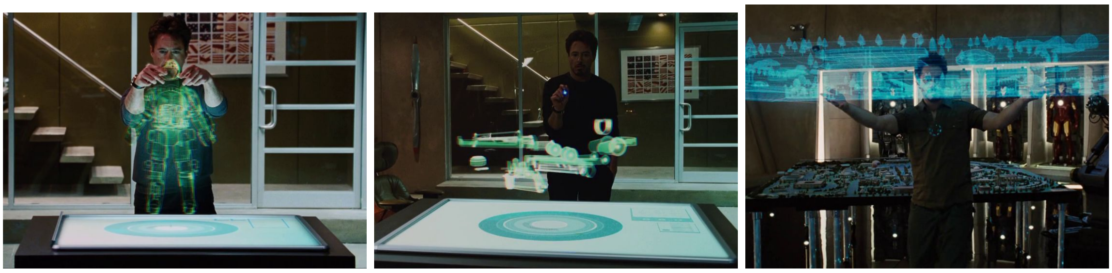


Figure 3.3: HoloTable interface from [Iron Man, 2008; Iron Man 2, 2010] movies. Credits: Paramount Picture.

user is able to rotate the 3D model using hand gestures, remove part of it by direct manipulation or using a laser like pointer to select parts at distance. Hand gestures such as zooming using both hands are revisited on the movie of the sequel (Figure 3.3 last image) where the hero scans a 3D scale model and interacts with the reconstructed wireframe 3D model presented as a floating 3D projection. We should highlight that on both examples, the user only control the visualization and remove parts of the model by direct manipulation using gestures in the air and he does not carry any complex modeling task. However, the idea to perform 3D modeling changes on virtual models using futuristic display technologies have been covered by other SciFi movies and we present the vision illustrated by three short movies in the following sub-sections.

3.1.2.1 Bruce Branit's *Word Builder* Short Movie (2007)

Our first modeling example is the [World Builder, 2007] short movies from a CG effect company by Bruce Branit. The scenario depicts a man creating a virtual city within one hour in what can be seen as a full immersive virtual environment. Instead of creating the city as a scale model, it is constructed at its real scale allowing the user to add geometric details while navigating in it. Buildings are created from large simple parallelepipeds defining the two diagonal extremities using both hands and pushing the dimensions along spatial axis using magic gestures when they are bigger than the volume defined by both hands as depicted by Figure 3.4. Duplicate volumes are created by taking a virtual snapshot using fingers to define a picture frame and moving it at distance to place them on scene trough hand gestures. This interaction scenario have been already

Figure 3.4: Modeling a city in an immersive environment at one to one scale from [World Builder, 2007]. Credits: Branit VFX.

explored by the SixSense research project [Mistry and Maes, 2009] combining wearable trackers and projectors. Regarding 3D modeling, details are added by sketching on building facades or by pushing and pulling parts as it is illustrated by the second image of Figure 3.4. The city seems to be procedurally generated and any feature can be manipulated by direct manipulation (Figure 3.4 third image). Menus to configure colors and textures and to select predefined models are displayed on the left arms of the user as a virtual palette which options can be picked and dropped directly using the right hand (Figure 3.4 last image). Virtual controllers appears directly on parts of the model to adjust some details, for example a slider on a glass window is used to control its degree of refection and transparency. This fictional scenario focuses on immersive virtual reality similar to the Star Trek holodeck scenario. While everything is at real size and behaves physically, everything can be customized or reedited using gestures and extending the physical limits of reality. At the end, the city is visited by the wife of the main character who is not able to distinguish it from the reality.

3.1.2.2 Dassault's *See what you mean* campaign (2005-2007)

As mentioned before, one of the advantages of scale models is its potential to be seen by several users and to be use as a presentation tool to support the discussion. The *See What you mean* advertising campaign was done for Dassault System (owner of computer aided design tools such as CATIA[1] and SolidWorks) to promote the benefits of 3D technology as a universal language. This campaign presents two videos around the concept of an editable scale model which users could manipulate naturally to discuss design changes in [See what you mean, 2005;

[1] 3D CAD design software CATIA, Dassault Systèmes: `www.3ds.com/products/catia/`

Figure 3.5: The living scale model from Dassault's *See what you mean* advertising campaign [See what you mean, 2005; See what you mean V2.0, 2007]. Credits: Dassault Systemes.

See what you mean V2.0, 2007]. The first video entitled "The meeting" (first three snapshots of Figure 3.5) shows a design review meeting with several participants from different fields i.e. design, engineering, architects and customers are around a scale model of a city. The scale model is "magically" animated and populated with both people and cars and is presented as a white living scale model of architectural proposal. Participants can remove parts of the model, experiment new design proposals of building, readjust parts of the model or visualize underground details of the virtual city. Everything is done using hand gestures and interacting directly with the animated scale model which is automatically updated illustrating and supporting the design discussion between the meeting participants. A second video shares another vision where three users discuss the same project remotely. On one side, the man is using its laptop for video conferencing near to an animated scale model of a bus. On the other side, two designers are looking to the 3D projection of the same model as illustrated by the Figure 3.5 last image. Both participants are performing changes on the virtual model which are automatically visible to the remote counterpart helping the communication and improving the understanding of the design review proposal on the fly.

Both videos focus on the ability of 3D technology as an unlimited media where users can interact physically as it is done with real objects or specifying gestures to update a scale model automatically improving the collaboration between users. The user interface is the 3D representation itself and users can interact naturally without needing any instrument or tool. The 3D model seems to be an intelligent living scale model. Physical user actions on the scale model are directly understood as modeling changes. While such vision is still out of science fiction, it illustrates possibilities between physical and virtual models which could be mimicked using existing 3D technology. In addition, such natural user interfaces might be possible in the future with the advances on interaction techniques using gestures and interactive surfaces.

3.1.2.3 Dassault's *Design Studio* video (2009)

Our last example of futuristic modeling interfaces revisits the Holotable scenario in a collaborative design session where several participants can interact remotely. The short movie illustrated by the Figure 3.6 snapshots was done the Dassault's *Design Studio* initiative joining several industrial designers and R&D engineers [Design Studio, 2009]. It presents a modeling vision around an interactive surface augmented by some 3d projection technology inspired by the Holotable from the IronMan SciFi movie. Using gestures and voice commands, the user is able to create a doorknob from a 2D drawing and to customize it. Compared to the previous examples, the graphical user interface of the modeling system is more visible and traditional closer to 3D virtual models than the idea of a scale physical model magically updated. Using gestures and touches, the user can interact with the CAD interface while viewing the 3D model in front of him through a 3d projection floating above the interactive surface. This video presents a multi modal interface, low level modeling operations can be activated through voice commands and controlled using gestures. While the user interacts by direct manipulation with 3D projected GUI, modeling changes are done indirectly using gesture above the projected area where the 3D model is displayed. For example, parametric patched defining the CAD model of the doorknob can be bent or twisted using both hands to define the deformation. The non co-location of user hands with the virtual model avoids occlusions while modeling. Co-location is mainly used to assign features such colors, rendering attributes, control the orientation of the object or even experiment the virtual model such as it is done in the Iron Man sequel. The vision presented by this movie proposes that each user has its own Holotable device and can work collaboratively complemented by video conferencing. While the scenario would be possible, they rely on complex gestures using fingers which might be unnatural for the user in non co-located scenario. This issues might increase due to the lack of haptic feedback when

Figure 3.6: The Holotable modeling environment from Dassault's [Design Studio, 2009] spot. Credits: Dassault Systemes.

interacting in the air.

3.2 Our Direct Modeling Approach

We propose a direct modeling approach to create, edit and manipulate 3D models using a small set of operations exploring gestures on and above an interactive surface. As depicted by Figure 3.7, we aim on mimicking the real interaction with physical mockups using plausible gestures to interact with the virtual model. Our visualization scenario is similar to the Holotable idea mixing an interactive surface and a stereoscopic projection fostering direct manipulation on 3D models. Such modeling environment is materialized with a multi-touch stereoscopic display showing objects as they were lying on top of the surface. Both the surface and the space above it can be used by a single user to model 3D shapes using sketches and gestures performed by its fingers or hands. To do so, we combine several

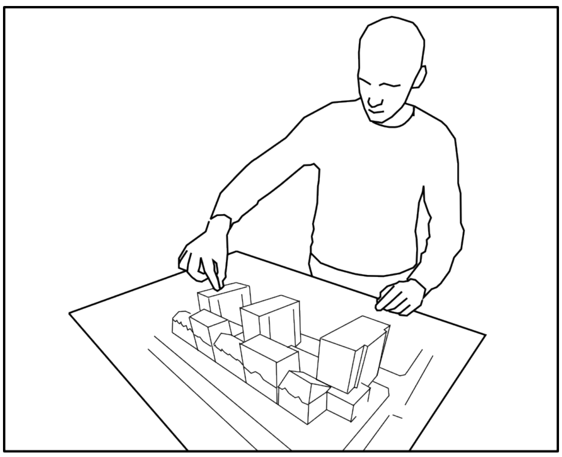


Figure 3.7: The Mockup Builder modeling environment scenario

tracking technologies such as depth camera sensors or input devices to follow rapid movements of fingers in space complementing the multi-touch inputs gathered by the surface. Multi-touch gestures can be used for sketching allowing to create 3D models by pushing and pulling existing content off the scene. The push and pull modeling metaphor, similar to operations provided by commercial tools as Google Sketchup or previous research prototype as Sesame [Oh, Stuerzlinger, and Danahy, 2006b], is ideal to foster direct manipulation over 3D virtual objects and explore both surface and space. One of the basis of this approach is to create 2D planar shapes then to extrude them along the normal to create volumetric shapes. While creation of planar shape would take advantage of the sketches performed by the user on the multi-touch surface, gestures in space maps the extrusion out of the surface naturally compared to traditional perspective clues used by desktop modeling applications. Such solution allows to further explore hand motions, since it can define 3D trajectories providing more expressiveness to the extrusion operation. In addition, features of the model such edges and faces can be directly selected in space without having to change the view as it is required by single perspective viewing in a standard monitor. Stereoscopic visualization provides a more complete view of the virtual model co-locating both the visualization space and the working space where users interact. The following subsections describe the four key concepts used by our direct modeling approach, i.e. using fingers and hands motion, combining on and above surface interaction spaces, 3D direct manipulation and the push and pull modeling metaphor.

3.2.1 Exploring Fingers and Hands Motion

Our approach explores gestures performed by both user fingers and hands tracked by a set of input devices on and above the surface. These gestures can correspond to sketches performed by the finger, motions perform by both hand and fingers describing trajectories or poses defined by hand fingers such pinching or pointing to select features on virtual objects. We choose fingers tracking complemented by handedness information as our main input modality captured by the multi-touch surface when user touches it and using Gametrak[2]

[2]See `http://en.wikipedia.org/wiki/Gametrak` for details.

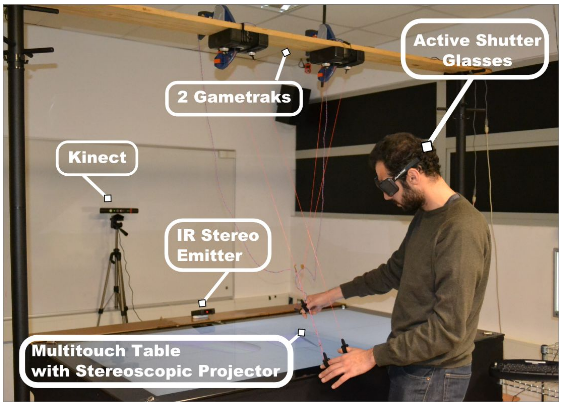


Figure 3.8: The Mockup Builder hardware setup overview

input devices once above as depicted by Figure 3.8. On the surface, the multi-touch technology is able to detect how many fingers are touching the surface, their position, movement and if a given finger leaves the surface. Such technology alone cannot identify which finger is in contact to the surface neither the hand they belong to. However combined with arms and hand tracking such information can be derived by proximity of finger touches regarding hand positions. As Figure 3.8 shows, we use the Microsoft Kinect[3] depth camera as a non invasive skeleton tracking solution. Such solution is used to identify how many users are around the table and to fully track their upper body part i.e. hands, wrists, elbows, arms, shoulders , head and spine. Currently our approach only supports one active user discarding the information from the other users located around the table. The active user is the first identified by the system. However it can be swapped to another of the tracked user if requested using a keyboard command in our current implementation. However this could be easily done using a specific gesture such as hand waving of one of the visible users.

Above the surface, the finger tracking relies on the usage of two Gametrak devices located above the surface (see Figure 3.8). The Gametrak are a simple and high frequency tracker based on two retractable string which we adapted attaching a ring at each extremity which can be worn on user fingers. Based on the length of the string and its orientation at the basis extremity, this input device provides the 3D position of the ring extremity. It was originally used as a gaming device for consoles such as the Sony PlayStation 2 or Microsoft XBox to track

[3]Kinect for Windows: `http://www.microsoft.com/en-us/kinectforwindows/`

both hands of a player using gloves attached to each string. However it became deprecated by next generation gaming controllers such as the Nintendo Wiimote, Microsoft Kinect and Sony Move. We use two of them above the surface in a reverse position to track two fingers (thumb and index) of each user hand. It provides a cheap and reliable finger tracking solution at 120 Hz which is unreachable at such cost by other tracking solutions. The handedness definition of each finger is explicit over the surface, since we are able to identify the device origin of each input. While the Gametrak device at the left most position is used by the user left hand, the one on the right is used by the user right hand.

While finger touches on the surface are well delimited knowing implicitly when sketches or gestures start and end, this is not so explicit above the surface when it comes to track gestures performed by user hands in the air. To propose a gesture based interface allowing direct manipulation of 3D models co-located with user hands, our approach needs an explicit mechanism to be sure when an user action starts and ends. Since gestural recognition by itself would introduce delay on triggering these actions, which is incompatible with direct manipulation metaphors, we decided to complement the Gametrak input devices with one button. The button is located on the ring of the Gametrak for each hand allowing to be pressed such as the user was pinching its thumb on its index finger. The user can define explicitly the beginning and end of a gesture above the surface by pressing and releasing the button. Such solution can also be used to confirm a selection and activate a specific modeling action.

Thanks to our input devices, we are able to track all fingers on the surface and at least two (thumb and index finger) of each hand above the surface. It means that for at least two fingers of each hand, we are able to track them all the time when they are on or above the surface. It will allow not only to chose the best input data when acting on the surface but also to explore gestures beginning on the surface and ending on space or vice versa since we can know when fingers leave or reach the surface. It can be explored by both gestures and sketches.

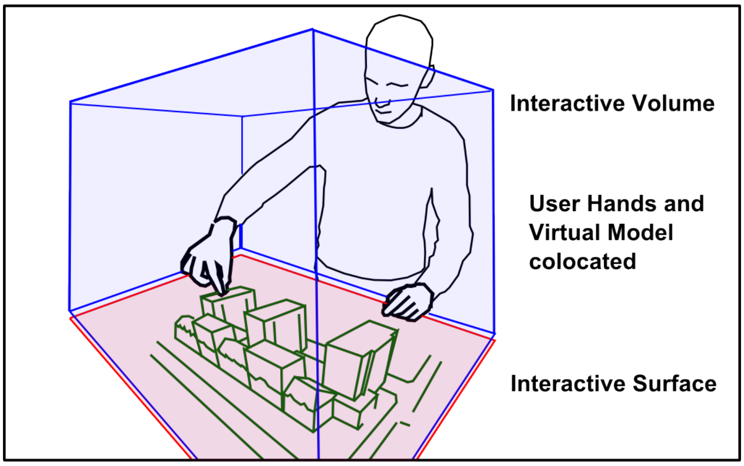


Figure 3.9: The Mockup Builder Interaction Spaces

3.2.2 Surface and In the Air Interaction Space

As explained before we are able to track user fingers on and above the surface and our modeling approach try to explore plausible gestures as the user was interacting with a physical mockup or scale model. The working space is compound two seamlessly integrated interaction spaces: the surface and the space above it as depicted by Figure 3.9. The stereoscopic visualization creates the illusion that 3D modeled shapes lie on top of the surface. The surface itself can be seen as a usual 2D drawing canvas where the user can sketch or perform gestures to move objects using a 2D direct manipulation metaphor. Sketches are performed by user fingers allowing to create 2D line and curve drawings or define planar shapes. By touching existing content on this drawing canvas users can select, edit or transform 2D shapes or even 3D volumetric shapes by interacting with their basis lying on the surface. The drawing canvas can also be used to present the graphical user interface such as circular menus to perform further modeling operations. When menus are visible, options can be selected by finger touches over the graphical representation. Touches can also define gestures similar to those used by existing multi-touch devices allowing to translate, rotate or scale shapes. When dealing with 3D shapes, one finger can be used to constrain the translation of objects on the surface plane. Using two fingers, both rotations around the surface normal axis and uniform scaling of objects can be done naturally such as the user was interacting with content on top of a table. The uniform scaling guarantees that the bottom face of the object remains on the surface.

The interaction space above the surface fosters 3D direct interaction metaphors to manipulate and select 3D objects. It is also adequate to complement the 2D surface with a third dimension and to support 2.5D gestures such as extruding a face from the surface into space. While interacting in space, the user can define curved trajectories or sketches or he can select object parts which are not lying on the surface. It gives access to unconstrained 3D manipulation of objects with more degree of freedom than the one available on the surface. However the control is not equivalent since the user is interacting freely in the air without the haptic feedback of the surface. We do not constraint the ability to use gestures in space as drawings even if planar shapes can be difficult or even impossible to sketch in space for some users. However this interactive space enables the user to select any visible feature of the geometric shape representation ranging from vertexes, edges to faces. It allows to fully take advantage of the depth perception and co-location between the virtual object and user hands. Beyond manipulating visible objects, we also allow the user to control the position, scale and orientation of the complete scene. It is particularly important since parts of the scene might be located outside both the interaction space and the visualization space presented stereoscopically on top of the surface. Scene manipulations allow the user to rescale the scene and to position it correctly overcoming the physical limits of our interaction space.

3.2.3 3D Direct Manipulation

The co-location between user hands and the virtual model invites the usage of direct manipulation metaphors turning the selection of shapes or its part into a critical operation for the suitability of our modeling approach. While this is done implicitly by touching a geometrical feature on the surface, we choose to use the explicit pinch gesture, i.e. accessible by pressing the Gametrak index button, in space mimicking a grabbing gesture of physical objects. While visual feedback on shapes and geometrical features is provided based on their proximity with fingers, the selection becomes active by pressing the Gametrak button in space. Our geometric shape representation decomposes the shape into vertexes, edges and faces. Faces are delimited by closed edges loops and edges might be represented by lines or curves. We highlight shape features in space depending of the index finger proximity taking advantage of the co-location between visualized objects and the interaction space on and above the surface.

Several selections can be performed with different granularity since any topological feature from our shape representation can be edited. A whole shape can be selected by intersecting its bounding box with a finger. Intersecting a face, edge or vertex highlights it for selection. Since edges and vertexes can be shared by more than one face or edge respectively, a continuous selection mechanism is provided to disambiguate the selection by analyzing the previously highlighted entity. Such approach considers that if the user wants to be more precise when specifying a shape feature, he will do it continuously starting with the selection of the shape feature with higher granularity, i.e. an edge related with a face or a vertex related with an edge. For example, it is possible to highlight a particular edge of face shared by two faces by selecting it from the face the user is interested in. Empty selections, which are useful for scene manipulation, are possible both on the surface or in the space above it by simply selecting an empty area of the scene (i.e. one that does not intersect any bounding box of a shape).

The direct manipulation allows to move vertexes, edges and faces along a given direction. The direction is given by the type of selection. If a highlighted face is is selected by pinching, the face will move along its normal direction until the selection ends. Other operations can be performed such as starting a curvilinear extrusion or a new extrusion by activating the corresponding option on the menu previously. Along the selection and the manipulation, the highlighted face is painted using a red color. When the user selects an edge, the continuous selection identifies a unique face which the edge belongs to. The direct manipulation of the edge will allow to move it along the plane defined by the face. We color the edge and the face into red and orange respectively. If the edge is selected directly without selecting any face, it will moves along the average normal direction of its adjacent faces. When selecting a vertex of an edge, we color the vertex, the edge and the face in red, orange and yellow respectively. The vertex will allow to be moved along the edge or freely in space it the vertex is selected directly. Thanks to the colored code highlight users can be aware of the type of selection, we also represent the normal of the face when a face is selected. Such selection method allows to perform in space well defined edition of our shape representation.

3.2.4 Push and Pull Modeling

As mentioned before, our approach proposes a push and pull direct modeling technique. This technique is supported by several CAD systems such as Google Sketchup, Dassault SolidWorks[4] and Autodesk 3D Studio Max[5] as an alternative modeling technique to boolean operators. This solution is well suited to 3D direct manipulation allowing to model complex boundary represented shapes. It mainly relies on using the extrusion operator to construct volumes from planar primitives as shows the example in Figure 3.10. Our approach allows to create new planar shapes by sketching on the surface. By default, any newly created faces can be extruded by direct manipulation. Extruding a face will update the topology of the face creating new adjacent faces which can be extruded afterward. When a face have been already extruded, the direct manipulation will move the face along its normal direction leaving the topology of the object unchanged. If the user wants to perform successive extrusion, he needs to change the default moving state of the face using the options available in our menus. We use contextual menus depending of the selection allowing to change the state of a selected face to move, extrude along the normal or along a curvilinear trajectory.

Another important operator of push and pull is the splitting operation. It enables to add detail or subdivide a face into several faces. This is done sketching over an existing face on the surface. We provide a mechanism to align an existing face with the surface avoiding having to sketch in space. The splitting operation can be done using a line, a curve or a polyline which will cut the face into parts. Any resulting face can be extruded as explained before. For example, several splits of the same face will subdivide it allowing the user to create a stair object similar to the one depicted by Figure 3.10 by successive extrusions of each subdivided

[4] 3D CAD Design Software SolidWorks: `https://www.solidworks.com/`

[5] Autodesk 3D Modeling and Rendering Software: `http://www.autodesk.com/products/autodesk-3ds-max/`

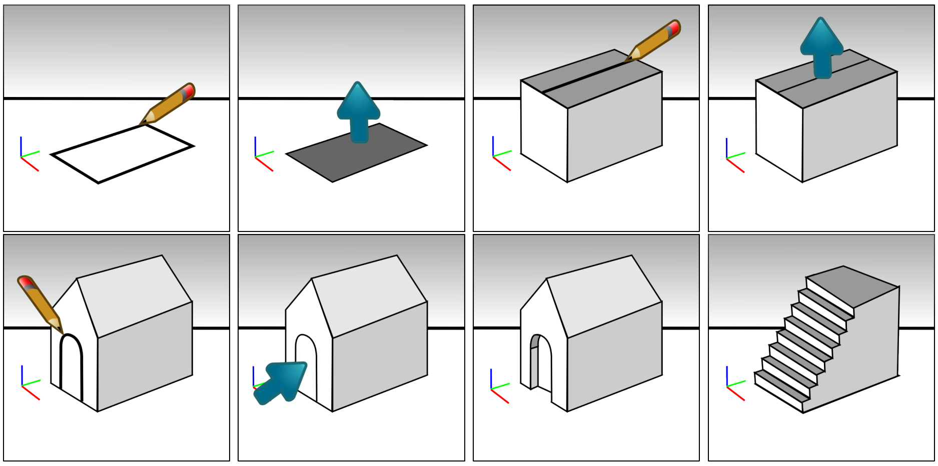


Figure 3.10: Push and Pull modeling sequence example of a simple house object (from left to right and top to bottom): user sketching a planar primitive, pulling the face to create a volume, sketching an edge on the top face, pulling the edge to create the roof, drawing a door on the front face, pushing the door to create an entrance. Finally we present the resulting simple house model and an example of a stair done by pulling an initial face split several times.

part.

3.3 3D User Interfaces

Our approach combines two interaction techniques to support 3D modeling tasks in our semi-immersive environment . The first is the usage of Guiard bimanual asymmetric model allowing to reduce the need of menus and enabling the user to focus on the 3D shapes to be modeled. The second concept is the continuous interaction space taking advantages seamlessly of our on and above surface interaction space for what it is more beneficial. Both techniques are used together using a rich set of input devices and propose a user interface which fosters plausible gestures mimicking the interaction with physical mockups or scale models. Before detailing the different modeling operations support by our user interface, Figure 3.11 presents an overview of all the user actions depending on the two interaction techniques.

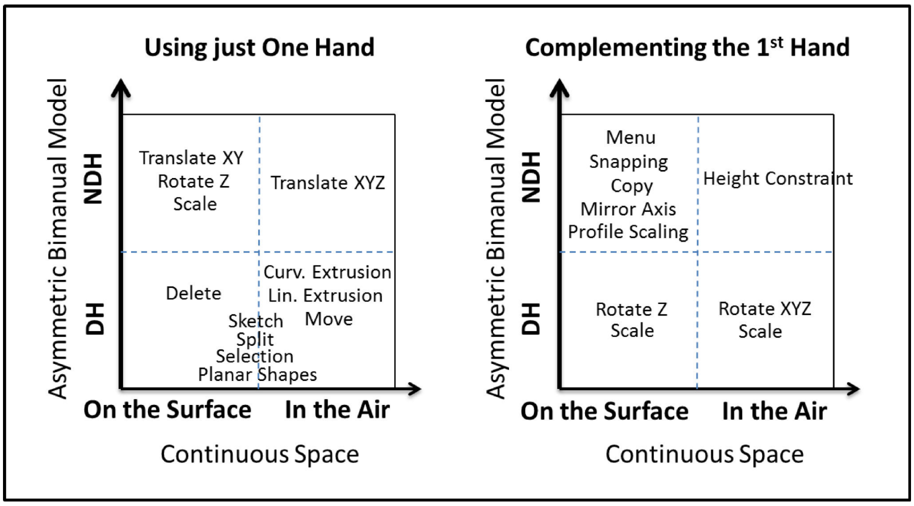


Figure 3.11: Overview of all user actions depending on the hand dominance versus the continuous space location: actions available when using only one hand (left) and actions activated by the second hand while performing a gesture with the first hand (right).

3.3.1 Bimanual Interaction for 3D Modeling

Our approach exploits the Guiard bimanual asymmetric model [Guiard, 1987] which results from the study of human manual asymmetry in everyday tasks.This model specifies different roles and distinct actions for each hand depending on human handedness. Handedness defines the preferred or dominant hand (DH) with which people have a finer motor skill compared to other hand named the non dominant hand (NDH). Table 3.1 identified the roles and actions of the non-dominant and dominant hand according to Guiard bimanual asymmetric model as summarized by [MacKenzie, 2003].

We use such descriptive model as a guiding principle to devise our model-

Hand	Role and Action
Non-Dominant	leads the preferred hand sets the spatial frame of reference for the preferred hand performs coarse movements
Dominant	follows the non-dominant hand works within established frame of reference set by the non-dominant hand performs fine movements

Table 3.1: Guiard bimanual asymmetric model principles

Dominant Hand	Non Dominant Hand
Sketching	Transform Shapes
Move Shape Features	Transform View
Highlight Shape Features	Show Menu
Extrude	Select Menu Options
Complement NDH Transformation	Activate Mirror Planes

Table 3.2: Mockup Builder bimanual model

ing interactive technique.While Sketching and Modeling operation are accessible using the dominant hand, the non dominant hand is mostly used for spatial transformations and interact with contextual menus. Table 3.2 presents with which hand the different operations provided by Mockup Builder are organized using Guiard principles. Operations which need fine movements such sketching or shape edition use the dominant hand. Spatial manipulations and menus are assigned to the non dominant hand allowing to implicitly change between modeling and navigation by alternating between both hands. Such approach reduce the needs of application modes or fixed menus such as it is done by traditional 2D graphical user interfaces. This model is valid on the surface and in the air taking advantage of the finger tracking provided by our input devices and the handedness defined by them.

3.3.2 Interacting on the Surface

The multi-touch surface is primarily used as a sketching canvas where the user interacts using its fingers. As previously explained, we followed the Guiard bimanual asymmetric model allowing the user to implicitly switch between sketching tasks and object transformation / world manipulation (scale, rotate, translate operations on objects or on the world) depending on the hand used. Using the DH , user can sketch on the surface creating planar shapes from close contours. Contours might use lines, curves or both and can be sketched using multiple strokes. Open strokes whose extremities are close to each other are merged into a single stroke. Topological shape features are highlighted if a touch selection is performed nearby. Additionally, planar faces can be sub-divided into an arbitrary number of faces with different shapes if a face is overlapped by an open stroke starting and finishing outside that face. Strokes are automatically fitted into lines and curves ready to be used as sketch. However, we also use a 2D shape recognizer [Fonseca and Jorge, 2000] allowing detecting simple gestures such as an erasing command by drawing a scribble. When an erasing gesture is recognized, if it overlaps open strokes, they are erased. However, if it overlaps only shapes and not open strokes, overlapped shapes are erased. This solution allows to use open strokes as construction lines while modeling.

When starting a gesture on the surface with the NDH , it is interpreted as object transformation if it is performed on an object, or world manipulation otherwise. Single touch gestures are interpreted as object or world translation. More than one finger gestures are interpreted as translation, rotation and scale operations on objects or world. 3D objects are constrained to movements along the plane parallel to the multi-touch surface. A gesture started with the NDH can be complemented by the DH allowing translation, rotation and scale with both hands (Figure 3.12).

Furthermore, bimanual interaction can be used to constrain drawing operations. In which case, the NDH defines constraints for the DH . For example, a user can sketch a straight line defining a plane of symmetry. First, the user selects the straight line using his NDH and sketches using the DH . As a result, the shapes sketched with the DH are mirrored by the plane of symmetry.

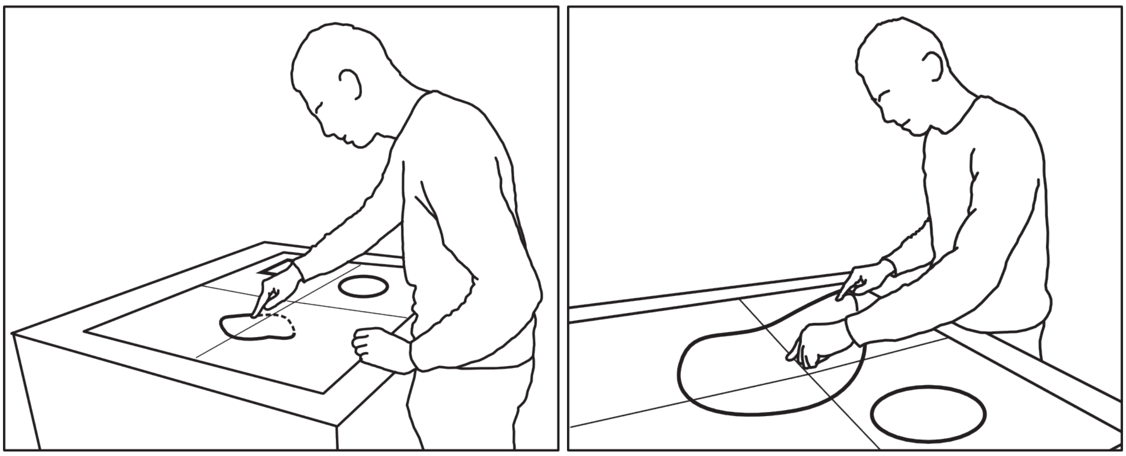


Figure 3.12: Bimanual Interaction on the Surface: Sketching using the DH (left) and scaling with both hands starting with the NDH (right).

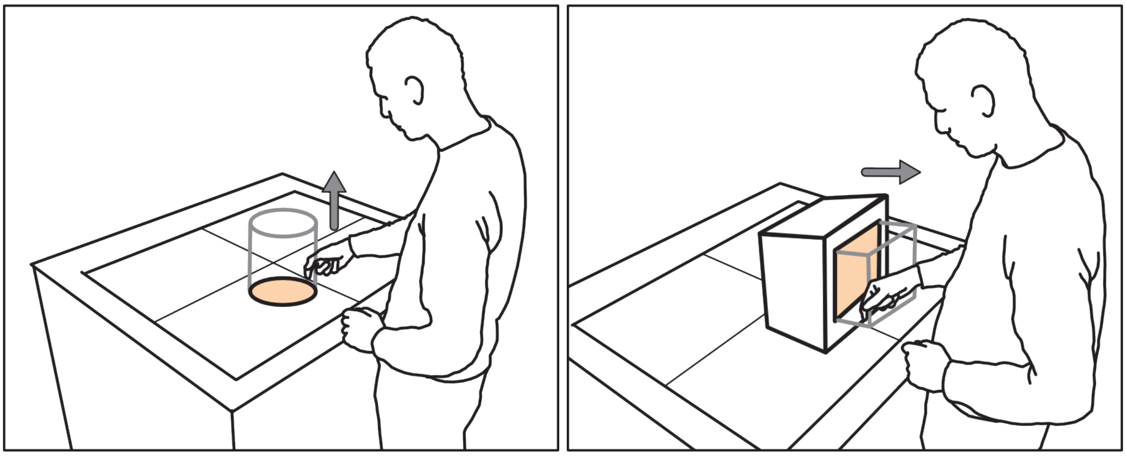


Figure 3.13: Face straight extrusion: along the surface normal direction (left), along a face normal direction (right).

3.3.3 Continuous Interaction Above the Surface

The notion of continuous interaction space introduced by [Marquardt et al., 2011] describes a direct touch surface and the space above it. Such environment is seen as a single interaction space where gestural acts flow from the touch to space and vice versa. This work illustrates a set of possible gestures to interact with 2D content mapping 2D gestures into 3D or combining them, Mockup Builder push this concept further applied to 3D modeling and combining a stereoscopic visualization enabling co-location between user hands and virtual content above the surface.

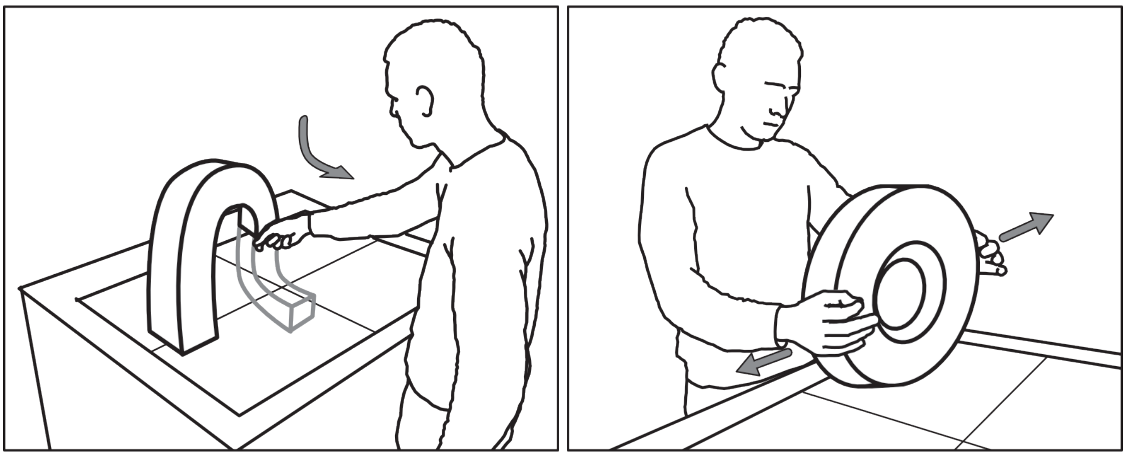


Figure 3.14: Extrusion along a curve gesture (left), 3D object scaling using both hands (right).

While DH gestures on the surface are used mainly for sketching, gestures with the DH above the surface are mainly interpreted as 3D object creation or edition. The DH also allows to sketch in the air creating 3D drawing. However the more natural usage of 3D space is to interact with the shape itself and perform volumetric operations. Creation consists in extruding a planar shape previously sketched on the surface. The user first approaches the DH index finger near a shape on the surface to highlight it. He then performs a pinch gesture ,pressing its thumb on the button located on its index finger, to extrude the shape along the normal of the surface (Figure 3.13). The height of the extruded object is then continuously updated and co-located with the finger position until the button is released. Planar shapes can also be extruded along the trajectory defined in the air after the user has selected this operation in a menu displayed on the NDH (Figure 3.14). While the user is defining the trajectory, the path is continuously re-evaluated and fitted into line segments and curve pieces to create a beautifulized freeform shape. Segments and curve pieces are created using the approach proposed by [Coquillart, 1987] to offset the gesture from the centroid of the face to its vertexes and create a smooth free form extrusion of the profile. This method allows to extrude both poly-line and curvilinear profiles along linear or curvilinear paths.

Editing follows the push and pull modeling metaphor where topological features of the shape (vertexes, edges and faces) are moved in the air along the normal direction of the face it belongs to. As described in Section 3.2.3, our continuous selection method allows to distinguish which face an edge or a vertex belongs to if needed. The user first highlights the geometrical feature by moving his DH index finger close to it. He then selects it with a pinch gesture using the index button. The position of the geometrical feature is then updated according to the finger position until the pinch gesture is released. Alternatively faces can be extruded along to their normal or following the trajectory defined by the user after the corresponding operation has been selected in the menu displayed on the NDH . If no geometrical feature is selected while doing the pinch gesture with the DH , the user can sketch 3D poly-lines or curves in space. Such as any editing operation, the 3D gesture in air is interpreted as a 3D drawing while the index button is pressed. All actions performed or finishing in the space above the surface will be

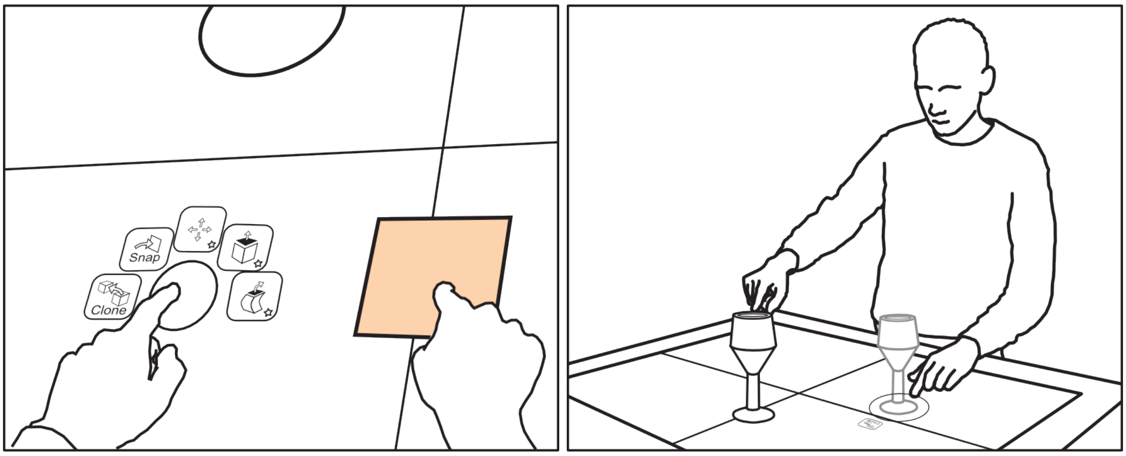


Figure 3.15: Example of menu presented under the NDH (left), cloning an object using both hands (right)

maintained while the user is pressing the index button and will end as soon as the button is released.

The bimanual interaction used on the surface is also valid above the surface allowing to rotate, translate and scale objects using two fingers. As on the surface, the NDH begins the interaction using a pinch gesture. The NDH defines translations only while the DH adds rotation and scale operations using the method proposed by [Wang, Paris, and Popović, 2011]. These direct 3D object manipulations appear much more efficient compared to indirect interactions on the multi-touch surface alone (e.g. changing the depth of an object while translating it along the surface plane).

3.3.4 Proximity Aware Menu Based Interaction

To support a versatile set of modeling operations, we still need to rely on menus for selecting these operations. Traditional 2D CAD systems use a variety of solutions to represent menus, from fixed toolbar to complex menu hierarchy to floating menus following the mouse cursor. Our idea is to minimize the need of menus to let the user focus on the 3D model and the drawing canvas. While the usage of the bimanual asymmetric model reduces the need of menu to switch between edition and manipulations, gestural acts are still ambiguous and might not be enough to distinguish between operations. For example, above the surface if the user wants to direct manipulate a face, extrude it along the normal or a curve, a similar gesture is natural to describe such change. Instead of making more complex gestures which could be harder to remember, we use menus to specify the different operations by changing the active modeling operation assigned to a shape feature. Then when the user selects the shape feature the correct modeling operation will be performed. Since the selected shape feature might be of different nature i.e. vertex, edge, face or even the whole shape selected, we propose contextual menus presenting the options or modes associated to the feature.

To present the contextual menu and available modes, we use the NDH while the user is highlighting shape features with the DH . Modes are presented through items shown in a contextual menu presented under the NDH . Modes presented in the contextual menu correspond to the ones available in the current mode associated to the operation performed by the DH (Figure 3.15). If the operation carried by the DH hand only supports a single mode, no contextual menu is shown under the NDH . To avoid visual clutter, the contextual menu transparency is adjusted based on the distance between the NDH and the surface. Above 15 cm, the menu is fully transparent and becomes progressively opaque as the NDH approaches the surface. To improve the accessibility, the contextual menu follows the NDH but its location is progressively fixed as the NDH comes closer to the surface to avoid spatial instabilities and reducing errors while selecting an item. This is simply done using the 1€ filter and adjusting its cutoff frequency based on the distance [Casiez, Roussel, and Vogel, 2012]. Such approach let the user invoke menus only when it is needed and they are positioned at more convenient locations on the drawing canvas.

3.3.5 Exploring On and Above the Surface Interaction

We have previously used asymmetric hand operations to implicitly switch between sketching, object transformation and world manipulation. We now illustrates how the NDH can complement the operations performed by the DH with two types of operations.

First, the discrete mode selection includes the type of extrusion (normal to a face or along a trajectory), the cloning operation and the snapping operation.

Once in the cloning mode, discrete touches with the NDH define the location where clones appear. Snapping is available when a face is selected. It consists in rotating the world to align the face with the surface.

Instead of defining discrete operations through a contextual menu, the NDH can be used to select a geometrical feature that defines a constraint for the DH . The constraint is enabled as long as the NDH keeps his selection active. We use plane and line constraints in the extrusion and positioning operations. For example, the NDH can select a face of an object to define the maximum or minimum height for an object being extruded with the DH as illustrated by Figure 3.16. Once the constraint is defined, the user continues to move his DH until the maximum or minimum height is reached. Further movements along the preceding direction do not continue to update the height of the object. This allows the user to also define that the height of an object should not be higher or lower that the height of another object. When translating an object, a plane constraint defines a limit beyond which an object cannot be moved further. While traditional modeling interfaces define constraints in a sequential way, we hypothesis that this definition of constraints on the fly allows to improve the flow of interaction.

Our last category of operations explores the usage of constrains continuously updated by the NDH instead of just defining discrete operations. This is illustrated with the scale constraint that consists in scaling the profile while extruding a

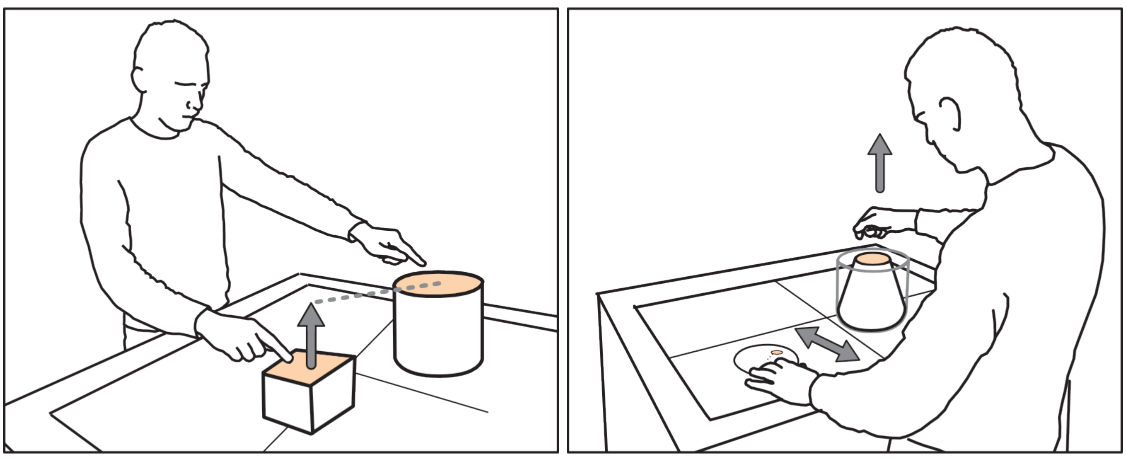


Figure 3.16: Defining an height constraint with the NDH (left), scaling with the NDH while extruding a shape (right).

shape (Figure 3.16). This allows to create a cone or a frustum from a circle or a quadrilateral planar face respectively. The scaling factor can be controlled dynamically using a 2D overlay menu accessible by the NDH while extruding the shape.

3.4 3D Manipulation and Scene Viewing

3D Manipulation and Scene control are important operations while modeling objects defining the spatial relationship or size between objects and helping the user to view and explore the 3D scene. By default, fingers of the NDH allow to perform spatial transformation of both objects and scene. Objects might be just a 2D planar shape or a volumetric shape lying on the surface or in the air. We present the gestural language used to perform these transformation based on hand and finger touch tracking. We use the NDH following the Guiard asymmetric principle which defines that on bimanual tasks the NDH is used as a spatial reference for the DH and should be preferred for coarse movements.

3.4.1 Manipulating Objects or Scene on the Surface

Finger tracking through multitouch is the main modality when interacting on the surface. Since the surface acts primarily as a 2D drawing canvas, we followed traditional touch based gestures to specified translating, rotating and scaling transformations as it is done by any touch based display. While touching a shape using a single finger will enable to translate objects on the surface plane, two fingers will allow to control its orientation and scale. The rotation and scale is controlled by the same gesture and depends of the angular changes and distance between both fingers. We use a physical based approach instead of gesture based approach. It means that from the beginning of the gesture until its end, the relationship between the position of the finger and the location on the shape remains unchanged. Looking to the rotation, such approach allows to define two type of rotations. While one finger remains fixed defining the rotation center, the other finger control the orientation around this rotation axis. The second rotation approach is moving both fingers in a counter clockwise or clockwise manner defining the rotation centers as the intersection between the line defined by the previous finger position and the line defined by the current finger position. Such model mimics the interaction with physical objects, allowing to even translate an object if both fingers move parallely. Volumetric shapes can also be manipulated by selecting faces lying on the surface. Thought the interaction on the surface, only rotation along the axis normal to the surface can be performed.

The scaling control is coupled to the rotation based on the distance evolution during the gesture performed using both fingers. If the distance increases, the shape will become bigger and if the finger get closer it become smaller compared to its previous size. The current scaling is uniform along each planar directions leaving the shape with the same proportions. One finger might remain at the same location while the other is moving or both fingers might move. For 3D shapes, the uniform scaling is performed such the intersecting plane with the drawing canvas remains unchanged guaranteeing that object lying on the surface remains on the same position regarding the surface.

Regarding scene control, we use the same gestural language to control position, rotation and scale on the visible drawing canvas. Both scene and object manipulations are differentiated by the content below the first touch. If the finger is over or intersect part of a shape, the transformation will be applied to the shape, otherwise it is applied to the scene. It means that for controlling the scene or viewing, the user is touching empty parts of the drawing canvas. We do not constraint the second finger location to control orientation and scale, since the first touch is enough to disambiguate between scene and object manipulation. Such approach is beneficial since when rescaling a small object, it might be difficult to place both fingers inside the shape.

As mentioned before, transformations are assigned to the NDH following the Guiard bimanual model. However, we decided to relax it to provide more control allowing the user to perform transformations using both hands. It appropriate since the shape might cover a large area of the canvas and they might be bigger than the hand being natural to manipulate them using both hand for the user. In addition, rotation and scaling based on two contact points can be seen as a asymmetric task when one of the contact points remains at fixed position while the other is moving, or as symmetric task when both finger touches are moving with the same logic. The relaxation of the Guiard asymmetric model was achieved using the predominance of the first hand. Any gestures started by fingers of the NDH will be used as transformation, so we can use the DH fingers to complement the action stated by the NDH . Such approach do not contradict the Guiard model and it even allow one to specify more controlled transformations since the DH , appropriate for fine movements, will complement the spatial reference defined by the NDH .

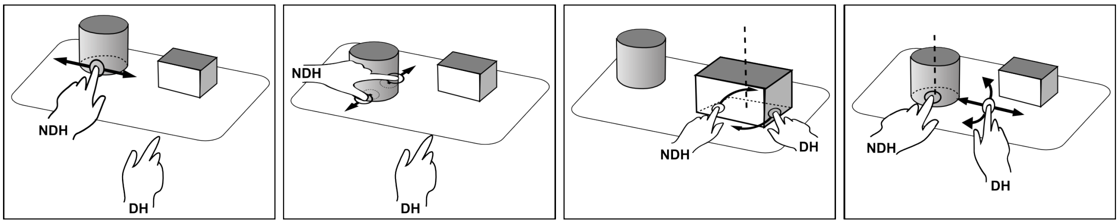


Figure 3.17: Multitouch gestures on the surface to control a 3D object or the view if the NDH finger do not intersect an object (from left to right): translating a shape on the surface using one finger, scaling using two fingers of the NDH symmetrically, rotating an object using one finger of each hand, using both hands asymmetrically for scaling or rotating shapes. The DH can support 3D manipulation on the surface since the gesture is originated by the NDH assigned for transformations.

3.4.2 Manipulating Objects or Scene in 3D Space

Such as on the surface, the NDH is used to start transformations in space. An object can be translated by moving the NDH over the virtual model intersecting part of the shape and performing the pinch gesture using the Gametrak button. It makes easy to stack objects over existing objects since the distance to surface can be controlled by direct manipulation. Due to the pinch activation which used both NDH fingers tracked by the Gametrak, it is not possible to specify rotation and scaling using two fingers as it is done on the surface. Following the relaxation to the Guiard Model, the DH can be used to complement the transformation started by the NDH . Such approach allows to control the scale and orientation of shape in 3D space using a handle bar metaphor. Using both hands, the user can define object scale by controlling the distance between both hands.

Regarding the rotation, it is coupled with the scaling transformation analyzing the angles between the line defined by the previous hand positions and the line defined by current hand positions. As on the surface, since the DH complements an actions started with the NDH , we do not force the DH to intersect the object. However, if both hand intersects the object to be manipulated, it behave much like a direct manipulation and it easier to control the transformation of the object. It can be seen as an extension of the direct manipulation provided on the surface to 3D space. The handle bar metaphor only allows to control two degree of freedom regarding the rotation. It is due to the fact that we only rely on hand positions without any information about their orientation. Such solution requires that the user correctly maps its hands on the object to perform the wanted rotation.

While performing the selection with the NDH to specify a transformation, if it does not intersect any object, the complete scene will be manipulated instead of just transforming one object. Such scene transformation uses the same gestures as the one used for transforming an object allowing to explore the scene or to align the model with the working area over the table. Thanks to availability of transformation at any moment, users can work at any scale and are not limited by the current scale or needing larger shapes to add details .

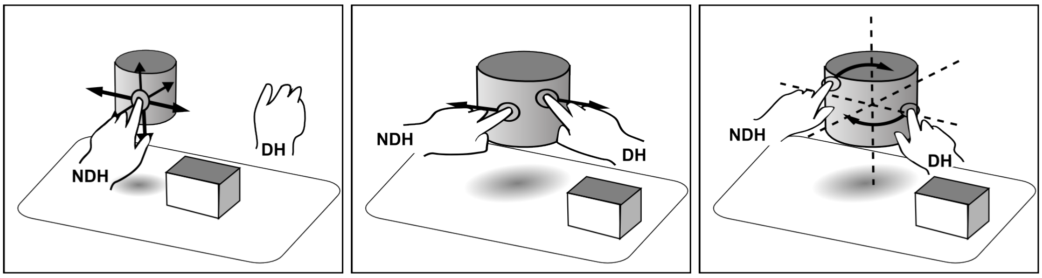


Figure 3.18: Multitouch gestures above the surface to control a 3D object or the view (from left to right): using the NDH to stack objects or translate them in space, using both hands to scale shapes uniformly , rotating an object using the handle bar metaphor in 3D space.

3.4.3 Transitioning between Surface and Space

The previous manipulations are unconstrained making it difficult to align a given face of an object with the surface which is the primary drawing canvas.

In addition, creating 3D planar shapes in space remains an operation difficult to perform due to lack of physical constraints to guide the hand. Even the simple operation of sketching a detail on an existing face floating in space might be difficult or even impossible to achieve with precision. To overcome such issues, we propose a snapping operator to easily switch between the surface and space allowing to use sketches on the surface or gestures in 3D space at convenience. Such operation allows to align a face of a shape with the surface by reorienting the complete scene. At any time, the user can get back to the original orientation of the scene.

Snapping is available through the contextual menu accessible on the NDH to snap on or back on any selected face (Figure 3.19 left). It works by computing a transformation matrix to align the 3D scene to the visible grid defined as a representation of the table surface. A simple linear animation between the two orientations is rendered to help the user understand the new orientation of the model. Furthermore, it allows sketching details on existing shapes (Figure 3.19 right) or guaranteeing that new shapes are created on top of an existing shape. Additionally, since existing objects can occlude the selected face due to scene orientation changed related to the snapping, we give to the user the possibility to clip part of the scene. The clipping is performed using a plane lying on the surface removing all the content of the scene which might exist between the user point of view and the surface. Such option can be activated and deactivated using our contextual menus.

3.5 3D Modeling Interface

Our modeling approach combines sketching with push and pull operators. While planar shapes can be creating sketching their contour, details can added by sketching strokes on it. These planar polygons can then be turned into volume by performing extrusions by direct manipulation of the polygon along its normal direction using gestures in space. We then propose a set of constrained operation allowing to propose a set of modeling operations allowing to create a wide variety of complex shapes as illustrated by Figure 3.20 showing several examples of 3D

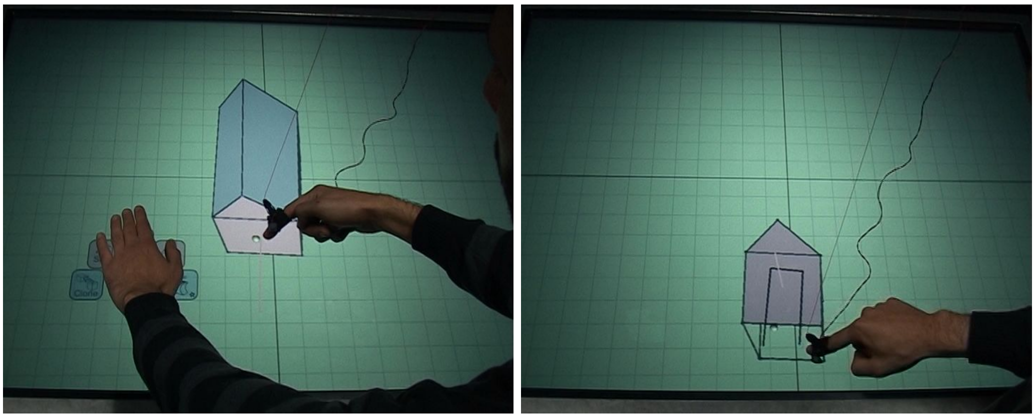


Figure 3.19: To sketch on a face perpendicular to the surface, the user first selects it directly in space using his DH and then selects the snapping option in the contextual menu displayed underneath the NDH (on the left). Then the scene is transformed aligning the face with the surface, allowing the user to add details on it by sketching (on the right).

models created using our approach.

3.5.1 Sketch based Modeling Interface

Our approach relies on drawing to create 2D planar polygon instead of instantiating 2D primitives provided by a graphical user interface such as squares, rectangles , circles etc... When the user is touching the surface, strokes are created based on the sequence of points from the first finger touch until it leaves the surface. These strokes are segmented into line and cubic Bézier curves creating a piecewise representation of lines and curves. The segmentation is based on an incremental fitting algorithm which generates lines or cubic Bézier curves according to an fitting error threshold. The user can sketch a planar shape using a single stroke or using several strokes. When adding a new stroke, we check if

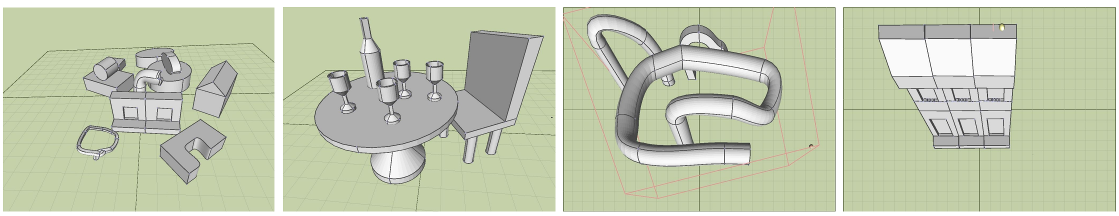


Figure 3.20: Examples of 3D models created using our modeling interface.

its extremity are close to another existing stroke. If the strokes close to each other are merges into a single stroke. Then if the both extremity of the strokes are close to each other, we create a planar polygon described by such contour stroke.

Symmetric shapes can also be created using a mirroring constraint. Any stroke representing a single line, can be used to describe a mirror plane defined by such line and the normal direction of the surface. When such mirror plane is activated by using the NDH to select the line, before starting a sketch using the DH , input strokes are duplicated sketching their reflections on the surface. It the reflection intersect the original stroke, it will be used to describe a symmetric planar shape. Otherwise, it will just create two symmetric strokes on the drawing canvas.

Given a close contour, we use a 2D recognizer named CALI [Fonseca and Jorge, 2000] to detect both circles and ellipses. CALI uses a descriptor vector to represent known shapes or gestures based on a set of 2D shape features. For each stroke query, this recognizer returns the most similar recognized shapes from a limited set of 2D shapes. If the stroke is similar to a circle or an ellipse, we replace the stroke by a piecewise representation of four cubic Bézier curves approximating the conic representation. If the recognizer fails, we try to beautify the contour based by energy minimization of detected constraints. We support the following set of constraints: parallelism, perpendicularity or edges with same length which we represent as energy functions. The detection is based on angular threshold between lines from the piecewise representation of the stroke. Such solution allow the user to sketch regular shapes as well as polygons using line and curved edges.

When the stroke sketched by the user does not represent a closed shape, it is processed as a single stroke to be added to the canvas. Before adding the stroke to the drawing canvas, we check if it does not correspond to an erasing gesture or if it does not intersect any existing 2D shape. We also use the CALI 2D recognizer [Fonseca and Jorge, 2000] to detect erasing gestures represented as wavy lines. If the erase command is detected, we erase all overlapping content which can be existing strokes on the drawing canvas or shapes intersected by the stroke. If the open stroke does not define a wavy line, we verified if it overlaps an existing 2D shape in order to be refined. Stroke intersecting shapes and both

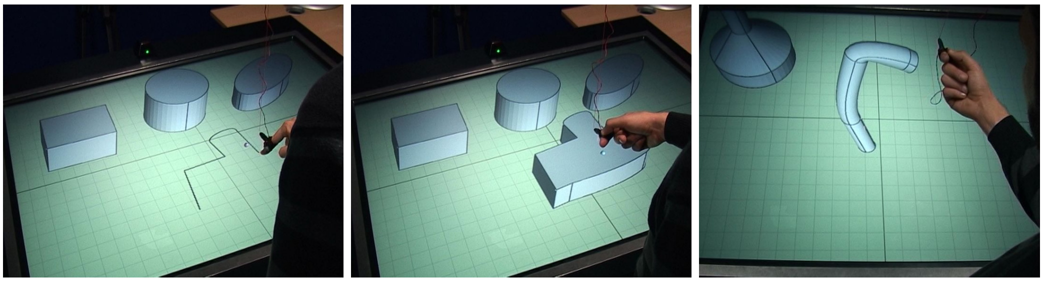


Figure 3.21: Example of the user sketching a planar complex shape on the surface then extruding it along the normal to create a volumetric shape. The last image presents a curvilinear extrusion using the hand motion as the extruding trajectory.

starting and finishing outside of a shape are interpreted as splitting strokes. These stroke will subdivide existing shapes using the stroke as a stencil. Any type of open strokes combining lines and curves can be used to split a shape and can even intersect a shape in more than two edges resulting in subdividing the shape in more than two parts. Such operation allows to add details to an existing shape.

3.5.2 Push and Pull Editing

Our sketch based approach allows to create a variety of planar shapes on the surface to be used as basis to create volumetric shapes lying on the surface . Such approach allow to create squares, rectangles, circles and ellipses as well as non regular polygons defined by curved and linear contours. The main operation of our push and pull is the extrusion operator. It allows to easily create cubes, parallelepipeds and cylinders by extruding regular 2D shapes along their normal direction. Irregular planar polygons can be extruding in the same way describing more complex 3D shapes as depicted by Figure 3.21.

The linear extrusion is interactive and its height can be readjusted until the end of the extrusion gesture performed using the Gametrak devices over the surface. As previously mentioned, any face of the 3D object can be extruded or moved linearly along its normal direction. Combined with the sketch based splitting approach, only parts of the shape can be extruded if wished by the user. Successive extrusions or curvilinear extrusions can be activated using the menu changing the default operation associated to the direct manipulation when selecting the shape.

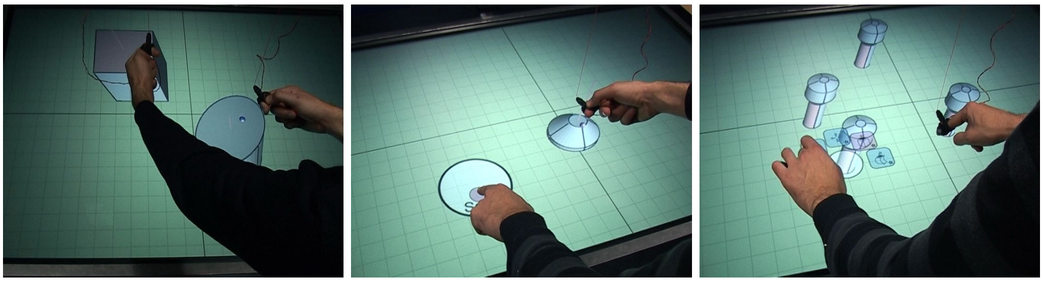


Figure 3.22: Constrained based operation examples. Using the NDH to specify a height constrain to limit an extrusion performed by DH . Middle image: using the contextual menu to scale the shape profile while extruding. Finally, using the contextual menu to copy a shape and to define its location.

The curvilinear extrusion uses the same metaphor than linear extrusion. However, it uses the trajectory defined by the hand motion instead of following the normal direction of a face as illustrates the last image of Figure 3.21. Such operator allow to create more free-form shapes extruding a profile along a rail interactively.

3.5.3 Constraint Based and Modular Modeling

To complement our sketch based and push and pull modeling approach, we propose a set of modifiers increasing our direct modeling ability as shown by Figure 3.22. The most basic one allows the user to limit the height of an extrusion to be aligned with another face from the same object of from a different one. While the user is extruding a shape with the DH , the NDH can be used to activate a planar constrain by selecting a face. Such constrain depending of the current height of the extrusion will create an upper or lower bound to the interactive extrusion. While the constrain remains active, it will influence the extrusion and it will be applied when the extrusion gesture ends.

Cones and pyramids can also be created using the profile scaling menu while extruding a circle or a square. If the menu is instantiated during the interactive linear extrusion, by approaching the NDH to the surface, users can control the scale of the profile of the top face of the extrusion. Such scaling is done according to the barycenter of the 2D shape. For example, if the user scales a circle, it will reduce or increase the radius of the circle allowing to represent a truncated cone or even a cone if it is scale until being a point. Such approach combine with successive extrusions can also be used to describe offsets of the shape within a shape.

Finally, we provide a cloning operator for replicating shapes on the scene. The update or modeling changes performed on a replicated object will be transfered to all instance making such operation different than a simple copy. This functionality is provided by the menu when selecting a shape or part of it and it will clone the shape as a whole. The position of the clone is defined by the location of the menu on the surface when selecting the option. Clones can be transformed separately if wished by the user, allowing to have several instance of an object with a different scale or orientation.

3.6 Summary

This chapter has described our direct modeling approach inspired by the construction of physical mockups. Combining gestures and sketches, we devise a modeling approach taking advantage of user drawing skills and advances on display technology to provide a semi-immersive environment. Our gestural modeling language relies on two interaction techniques: the bimanual assymetric model and continuous space fostering plausible gestures mimicking the interaction with physical mockups. We propose a reduce but still expressive set of modeling operators providing a new dimension to 3D direct manipulation and extending traditional 2D push and pull metaphor. This is achieved with a seamless integration of sketch based modeling techniques into a unique semi immersive environment defined by an interactive surface and the space above it. Finally we illustrated our modeling interface explaining how planar shapes are created and edited using push and pull operators which can be constrained on the fly by the user.

4

The Mockup Builder Modeling Environment

This chapter describes the different hardware components of our semi-immersive environment setup. We present how the different input devices are calibrated in order to produce an enriched description of user gestures. Then we describe our input data is processed to represent strokes and gestures to be used for 3D modeling tasks. Finally we describe how the interactive modeling operators where implemented.

4.1 Semi-Immersive Modeling Environment

Our modeling environment proposes an enriched tabletop environment using a stereoscopic screen and a set of input devices to track user gestures. This section presents our setup which was deployed in two research laboratories. While the initial setup was designed at the INRIA Lille in France, a second version using similar hardware was installed at João Lourenço Fernandes Laboratory from the Instituto Superior Tecnico at TagusPark in Portugal. This section discusses the hardware involved in our setup and explains how user gestures are tracked on and above the interactive surface. Finally, we explain how virtual content are rendered using a stereoscopic visualization creating the illusion that objects lie on top of the table.

4.1.1 Hardware Modeling Setup Overview

Our setup consists in a semi-immersive environment based on a stereoscopic multi-touch display 96×72 cm (42 inches) combined with a Kinect depth camera and two Gametraks used to identify and track the hands and fingers above the surface.

Head tracking is achieved in a non-intrusive way thanks to the Kinect using

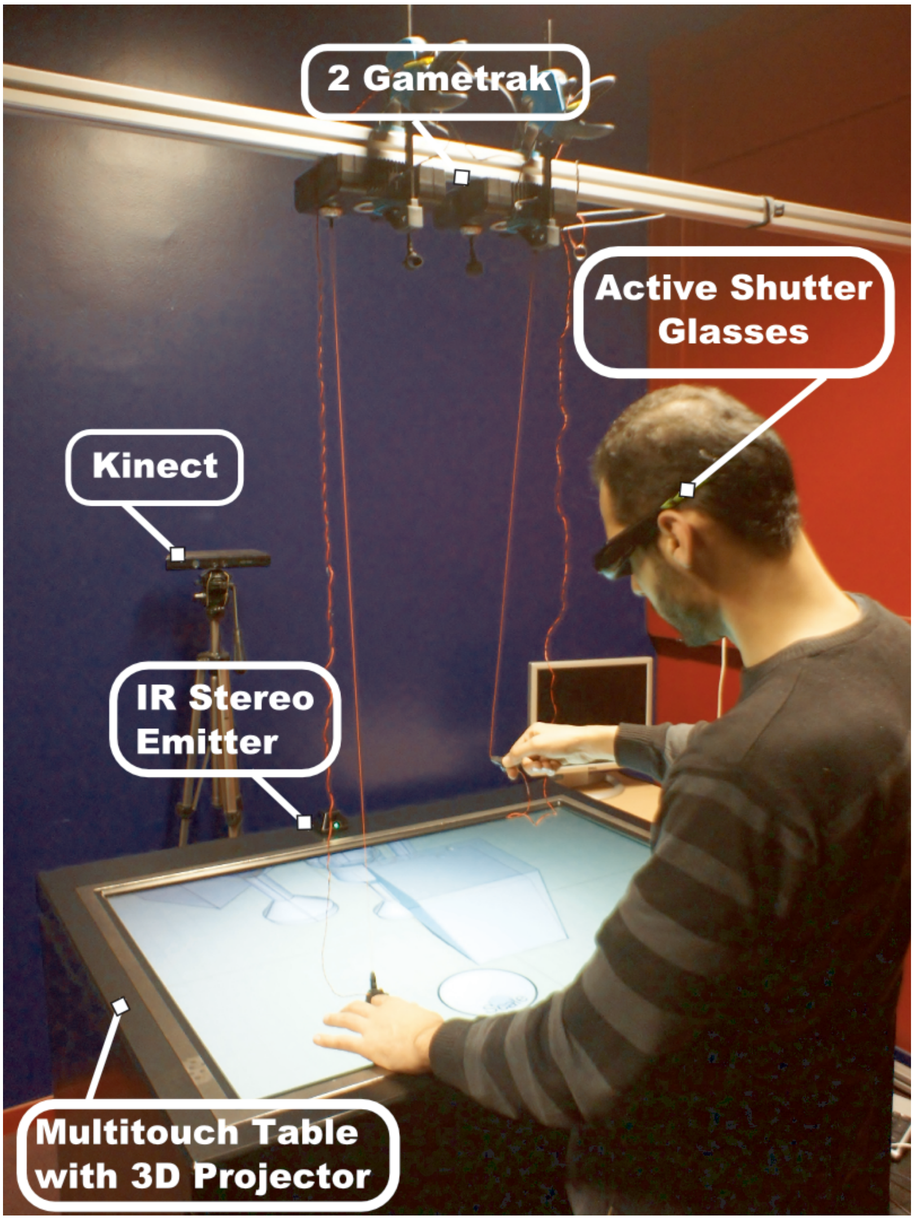


Figure 4.1: Overview of the setup.

its skeleton detection algorithm. The skeleton is also used to track user hands allowing to locate the dominant hand according to the handedness of the user. The Kinect skeleton algorithm, based on both OpenNI[1] and NITE[2] frameworks, provides a bone approximating the position and orientation of both head and hands centers. Finger tracking, i.e. positions of the finger tips, is operated through multi-touch on the surface and approximated using Gametrak devices in space (Figure 4.1). The visualization relies on a back-projection based system located under the table running at 120 Hz with a 1024 × 768 pixels resolution giving a pixel density of 10.6 pixels per cm (27 dots per inch (DPI)). It is coupled with active shutter glasses from 3D Vision NVIDIA for the stereoscopic visualization.The 3D scene is rendered on top of the surface and the point of view is updated according to the position and orientation of the user's head to take into account motion parallax. The IR transmitter for the glasses uses an IR wavelength different from the multi-touch table which is based on the Diffuse Illumination technique. It is set at a position to cover the working volume around the table where the user interacts.

A camera running at 120 Hz with a 640×480 pixels resolution and positioned under the surface records finger movements on the surface, providing a maximum resolution of 6.4 dots per cm (16.25 DPI) for finger tracking. We use the iLight[3] framework version 1.6 for fingers detection and tracking. Fingers data are then sent using tangible user interface objects (TUIO) [Kaltenbrunner et al., 2005] messages to our custom built application.

The two Gametraks are used to track the 3D position of the index and thumb of each hand when they are no longer in contact with the multi-touch surface. These low cost gaming devices are placed in a reverse position centered above the table at a distance of 120 cm. The 3D position of each finger is computed from the two angles of rotation and the length of each cable, digitalized on 16 bits and reported at 125Hz to the host computer, resulting in a theoretical position resolution going from 500 dots per cm (1250 DPI) when the finger is close to the surface to 900 dots per cm (2250 DPI) when it is 50 cm above it. However the

[1]OpenNI the standard framework for 3D sensing:`http://www.openni.org/`
[2]Nite Middleware PrimeSense: `http://www.primesense.com/solutions/nite-middleware/`
[3]iliGHT Tactile Table product page: `http://www.immersion.fr`

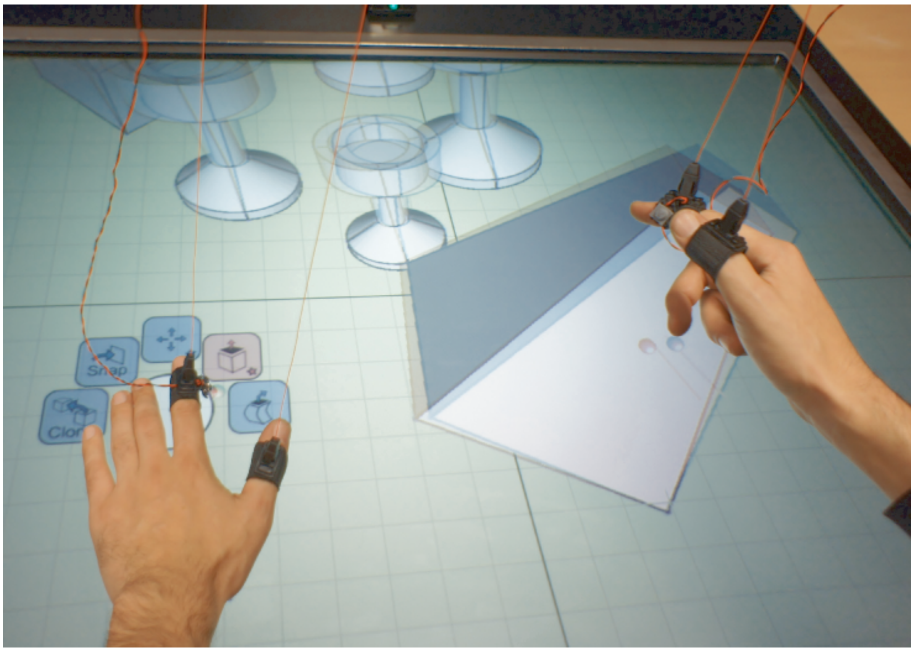


Figure 4.2: Detailed view of the Gametrak strings attached to the fingers with the buttons used for pinch gestures

effective resolution is far lower (around 10 DPI) due to measurement noise. The retractable strings are attached to the fingers through a ring. Although strings introduce some visual clutter, they were not found to distract users from their task. The strings create a minor spring effect which reduces user hand tremor without adding fatigue. We added a 6mm diameter low profile momentary switch button on each index finger to detect pinch gestures without ambiguity (Figure 4.2). This simple solution provides a good trade-off regarding precision, cost and cumbersomeness compared to using a high end marker based optical tracking system or low sampling frequency (30 Hz) device such as the Kinect. The latter presents also a low tracking resolution (from 3 to 8 DPI) and is subject to finger occlusion.

The redundancy of information from the different input devices allows us to identify which finger of which hand is interacting on the surface or in the air or to choose the input source with the best tracking resolution.

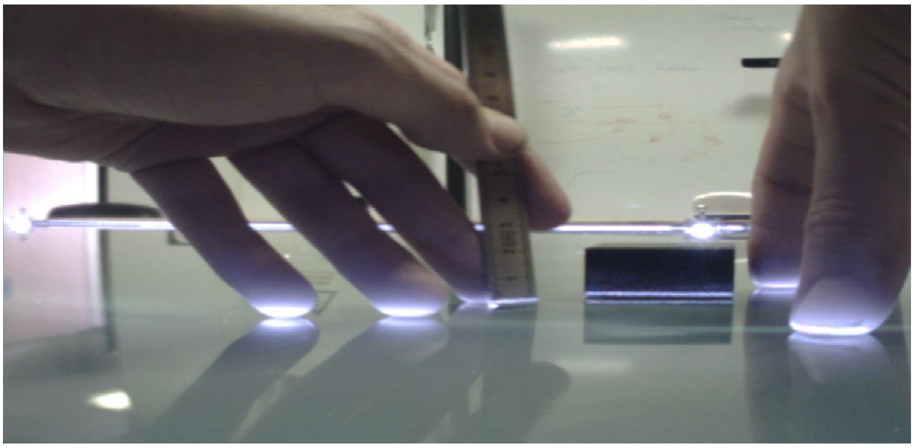


Figure 4.3: Visualization of the Laser Light Plane on the multi-touch surface to detect user fingers on the surface.

4.1.2 Tracking user fingers on the surface

We use a multi-touch projection based screen. Touches are detected using a Laser Light Plane solution using six lasers over the surface. Coupled with line generator lens such configuration is able to create a laser plane near to the surface. Dedicated mounts were created to correctly align the laser plane with precision. Thanks to the laser mounts , we are able to generate a plane at a distance closer than one millimeter to physical surface as it can be seen in Figure 4.3. This minimal distance between both the physical and laser plane, is important to guarantee that touches only happen when user touches the surface. We choose to rely on vision based multitouch technique instead of using capacitive technology to offer a large interactive surface. Currently, capacitive multitouch displays are still limited to 32″ screen diagonal and no commercial solution are available for such area coupled with a stereoscopic capable screen. While the laser vision technique is not as reliable for multitouch as capacitive technology, it provided a good trade-off between the coverage area, number of touches detected and cost.

While user interacts on the surface, user fingers intersect the laser light plane redirecting the laser light inside the projection table. The redirected laser light is captured by a camera inside the table. The camera uses a laser filter to only capture the wavelength corresponding to the lasers and a lens to be able to cover all the interactive projected area. Then, the image captured by the camera is processed in order to detect blobs corresponding to finger touches. We use an open source solution named Community Core Vision[4] to perform the image processing and generate multi-touch inputs using the TUIO specification protocol. The TUIO is a simple open, transport-independent, message-based protocol based on Open Sound Control (OSC) protocol developed for communication among computers, sound synthesizers, and other multimedia devices. This protocol describes for each frame of the camera the resulting set of alive touches. For each touch, a unique identifier is generated, its normalized 2D position, velocity and acceleration. The tracking tool guarantees the same identifier for a touch while the user is interacting on the surface i.e. from the moment it appears on a frame until the user release its finger from the surface. This is done by using a Kalman Filter to identify touches correspondence between successive frames. Using such solution, we are able to detect the set of finger touches over the surface and use them for modeling, sketching and manipulation.

4.1.3 Tracking user fingers above the surface

Above the surface fingers are tracked using the Gametrak input devices. As explained before, we use two Gametraks in a reverse position above the surface. Such configuration allow to track two fingers (i.e. thumb and index finger) per device. The strings of the Gametrak are attached to user fingers through a ring. Each device is assigned to track two fingers of one user hand, and one of the rings provides a button which can be pressed by the user mimicking a pinch gesture.

The Gametrak devices are seen as game input devices and implement the operating system human interface device protocol. Such interface provides the raw data of the device i.e. the XY orientation of both strings represented as an analogical cursor, the length of both strings and the status of the button. Such information is converted into a 3D position in meters regarding the basis of the Gametrak device using a lookup table created from physical measurements. Each device provides the 3D position of string extremities regarding its own basis at a 125 hz frequency. To remove the spatial jitter coming from the device and the

[4]Community Core Vision: `http://ccv.nuigroup.com/`

user, we start by filtering the Gametrak data using the 1€ filter [Casiez, Roussel, and Vogel, 2012] then we send the information through the network for our application using the TUIO protocol specification. While touches use the 2D cursor data representation described by the "/tuio/2Dcur" tag, the Gametrak data uses a revised version of the 3D cursor definition using the 3D "/tuio/3Dcur" tag modified to include the button state information. Thanks to these two devices we are able to track the position of two fingers of both hands above the surface. We also provide an explicit way to define gesture starts using the index button which can be pressed by pinch gesture while the user is interacting above the surface.

4.1.4 Tracking user posture

The user posture is tracked using the Microsoft Kinect camera. For our approach, we currently use both hand and head positions provided by the skeleton tracking algorithm offered by the OpenNI framework. Both hands and head correspond to bones of the upper part skeleton. Based on an initial calibration using a Y body pose, such framework is able to track a skeleton in real-time giving the position and orientation of each skeleton bone. To avoid recalibration of the system at each usage, we stored the skeleton definition of one user to be reused by the MockupBuilder system. If the algorithm is not able to match the skeleton to the current user, a calibration is required and can be performed on the fly. However, during the several experiments and evaluation of our system, we do not notice any need to recalibrate the skeleton tracking. The skeleton tracking is done up to 30 Hz due to the limited capture rate of the Kinect camera. While this solution is not the most accurate to track the user posture, it provides a good trade off between quality, setup and cost since the user do not need to be instrumented with optical rigid marker. The hand position information is in fact only used as a guess and can be combined to the information obtained by the Gametrak devices. Regarding the head position, its main usage is related to the head tracking for the stereoscopic visualization as we will describe in the next sub section.

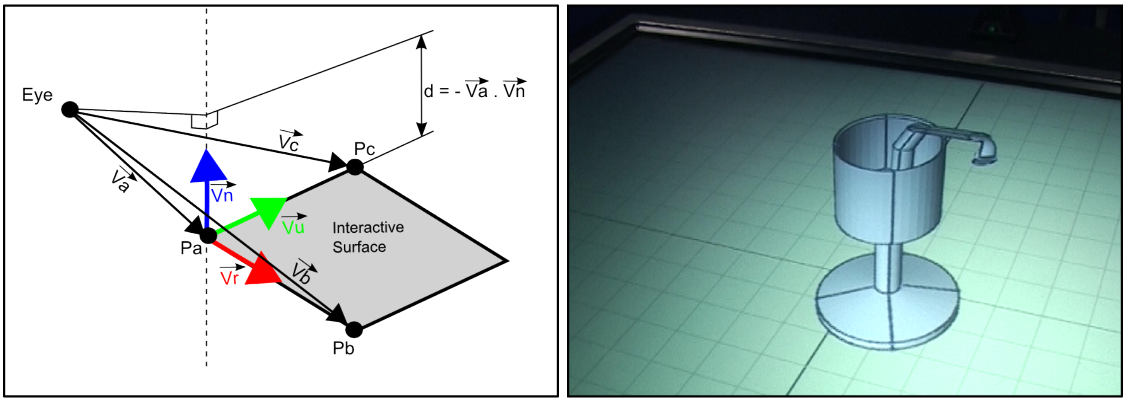


Figure 4.4: Definition of the off axis perspective projection to be used by each eye position of the stereo camera pair. The position P_a, P_b and P_c define the location of the screen in the Kinect coordinate space. The Eye position is computed per eye at each frame based on the user head position. The screen is then defined by the orthonormal basis (V_r, V_u, V_n). Image on the right: off-axis perspective example.

4.1.5 Visualizing stereoscopic off the screen content

Our semi-immersive environment relies on a stereoscopic visualization provided by 120 Hz 720p projector under the table. The goal is to create the illusion that objects are lying on top of the table which acts mainly as a drawing canvas. Such illusion provides a co-location between the virtual objects and the hands of the user as illustrated by the right image of Figure 4.4. To create such effect, we defined an off-axis perspective based on the head tracking and the location of the visualization surface.

Using the head position and orientation provided by the tracked skeleton, we estimate the position of user eyes using a fixed 3D vector distance over the tracked head bone of the skeleton. Based on this position defining the user eye center located between both eyes, we establish a eye separation of six centimeters to compute the stereo camera pair. The stereo camera is represented by two virtual cameras rendering the scene from each eye to create the stereoscopic effect. For each eye position, the perspective projection is computed as an off center axis perspective using the physical position of each corner of the surface. The perspective frustum is delimited by the four screen corners and the eye position in our world reference coordinate space described in Section 4.2. Using such information, we can compute both the projection and viewing matrices to render our 3D scene. The projection matrix requires the computation of bottom, top, left, right clipping values as follow using the orthonormal basis of the screen ($\overrightarrow{V_r}, \overrightarrow{V_u}, \overrightarrow{V_n}$), the vectors $\overrightarrow{V_a}$, $\overrightarrow{V_b}$ and $\overrightarrow{V_c}$, the distance d from the eye to the screen plane, the *near* and *far* plane value. These variables and vectors are depicted in Figure 4.4.

$$left = \frac{\overrightarrow{V_r} \cdot \overrightarrow{V_a} * near}{d}$$
$$right = \frac{\overrightarrow{V_r} \cdot \overrightarrow{V_b} * near}{d}$$
$$bottom = \frac{\overrightarrow{V_u} \cdot \overrightarrow{V_a} * near}{d}$$
$$top = \frac{\overrightarrow{V_u} \cdot \overrightarrow{V_c} * near}{d}$$

These values are then used to define the projection matrix which can be implemented in OpenGL using the *glFrustum* function.

$$ProjectionMatrix = \begin{bmatrix} \frac{2near}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & -\frac{2farnear}{far-near} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Finally, we can define the viewing matrix for each eye position of the stereo camera pair. The right image of Figure 4.4 presents an example of using both matrix and present our environment from the user point of view. To produce this view, we track a physical camera instead of the user head. The off center axis perspective creates the illusion that the virtual glass object is on top of the interactive surface.

$$ViewingMatrix = \begin{bmatrix} Vr_x & Vr_y & Vr_z & 0 \\ Vu_x & Vu_y & Vu_z & 0 \\ Vn_x & Vn_y & Vn_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & -Eye_x \\ 0 & 1 & 0 & -Eye_y \\ 0 & 0 & 1 & -Eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.2 Enriching User Input Model

Our setup relies on several input devices (Gametraks, Kinect and Multi-touch) which should be on the same coordinate system to obtain a continuous interaction space. Figure 4.5 presents the three coordinate systems related with the different

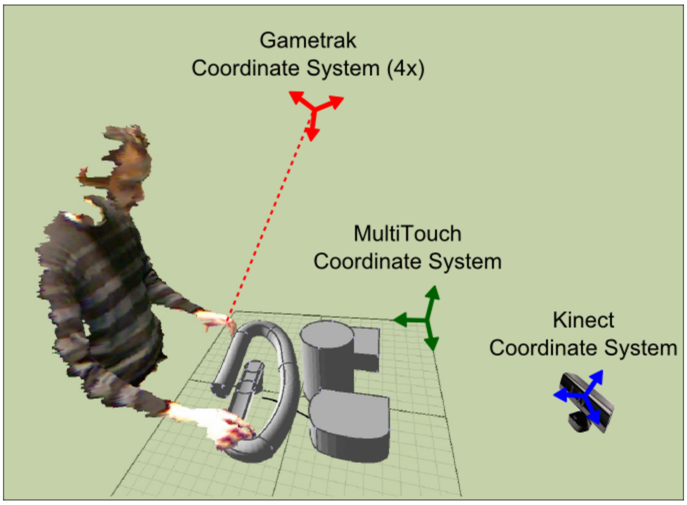


Figure 4.5: The three coordinate systems used in our semi-immersive modeling environment. Both Gametrak and Multi-touch input coordinates are transformed to the Kinect coordinate system thanks to the calibration procedure to create a continuous interaction space.

input devices. We chose the Kinect coordinate system as our primary coordinate system since it covers both the working and the user spaces. This section explains how we calibrate our continuous interaction space and how the input data is fused into a single user model. Combining the different input devices, our user model will be able to track the user head (orientation and position), hand positions and finger positions on the surface and in the air for at least two fingers of each hand. The handedness and location of the finger (on the surface or in the air) would be also computed combining the available information from each sensor.

4.2.1 Calibrating Multi-touch Input Data

While the Kinect camera is able to track the user body in particular its head and hands, the multi-touch surface is able to detect only 2D positions of the finger tips on the surface and can be used to define the location of stereoscopic display. We need to fuse both spaces in order to map 2D finger positions into 3D space and to correctly create the stereoscopic illusion on the surface. In addition, it will enable to use the 3D Kinect skeleton to track both hands and to specify the handedness of each finger touch on the surface.

We provide a simple application for the user to pick the four corners of the multi-touch display in an image captured by the Kinect. These four points coupled with the 3D coordinate extracted from the Kinect depth map are used to compute the plane which minimizes the distance between them. The plane is then used to define two matrices converting touches on the multi-touch surface into 3D positions and vice versa. Figure 4.6 presents a screenshot of our calibration application allowing the user to assess the correctness of the calibration thanks to a 3D preview of the plane and its mesh representation captured by the Kinect. The screen plane definition is used to define the frustum of the off-axis stereo perspective projection to render 3D content on top of the surface from the user point of view.

4.2.2 Calibrating Gametrak Input Data

After being able to place all the surface information (i.e. touches) into the Kinect coordinate system, we need to transform our Gametrak finger tracking information to the same reference space guaranteeing that all user inputs are defined in the same coordinate system. The nature of Gametrak device defines its input data in a framework centered on the device base which both location and orientation are unknown or imprecise if relying on physical measurements. Our solution is to propose a semi-automatic procedure to compute a transformation matrix into our primary coordinate system for each tracked finger. This is done

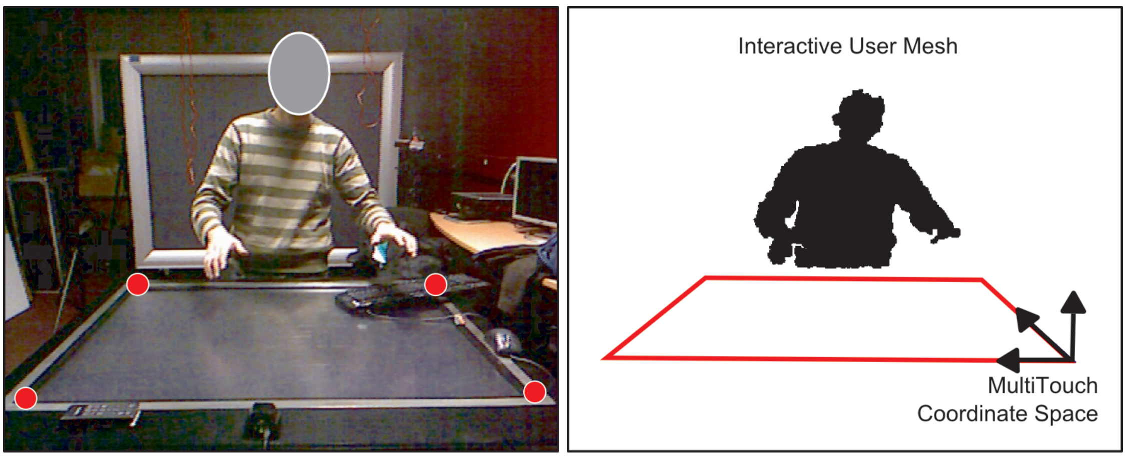


Figure 4.6: Calibrating 2D Touches: Kinect image camera with the four corner points selected by the user (red dots) on the left, 3D view of the user with the resulting screen plane on the right

collecting a set of one thousand matching 3D position pairs to compute the correspondence rigid transformation. The set is created by sampling the multi-touch surface screen and gathering both Gametrak positions and corresponding touch positions converted to our primary coordinate system using the matrix defined on the previous section. To guarantee the matching of both positions, we require that only the finger attach to the Gametrak string touches the surface while the sampling is performed. Then the rigid transformation is computed using a RANSAC algorithm [Fischler and Bolles, 1981], creating a matrix mapping Gametrak positions to our global coordinate system. This process is repeated four times, one for each Gametrak string.

4.2.3 Fusing Inputs into a Single User Model

All input data that belong to the same finger are fused together as an input gesture based on the proximity of positions when converted to our primary coordinate system. An input gesture might represent a stroke or gesture on or above the surface. Data coming from the multi-touch surface or the Gametraks has a unique identifier defined by the input device. After the coordinates have been converted into the same coordinate system, the fusing consists in determining when the identifiers from different sources correspond to the same finger. It also consists in adding the handedness information to each finger. A new input gesture is created when a finger touches the multi-touch surface without doing any pinch gesture, or when the finger performs the pinch using the Gametrak button and that finger was not touching the surface before. Input gestures are deleted when fingers are lifted from the surface without any pinching or when the pinch button is released above the surface. Otherwise the input gesture is updated. Multi-touch and Gametrak data are fused together based on close proximity. When a finger is on the multi-touch surface, we discard Gametrak data even if they are available as they were found to be less reliable. When a new input gesture is created, input handedness is determined by the closest hand position obtained from the Kinect skeleton. Figure 4.7 presents the different information which can be established and used to be aware of the user movements in our working space. While both head and hand position and orientation are tracked

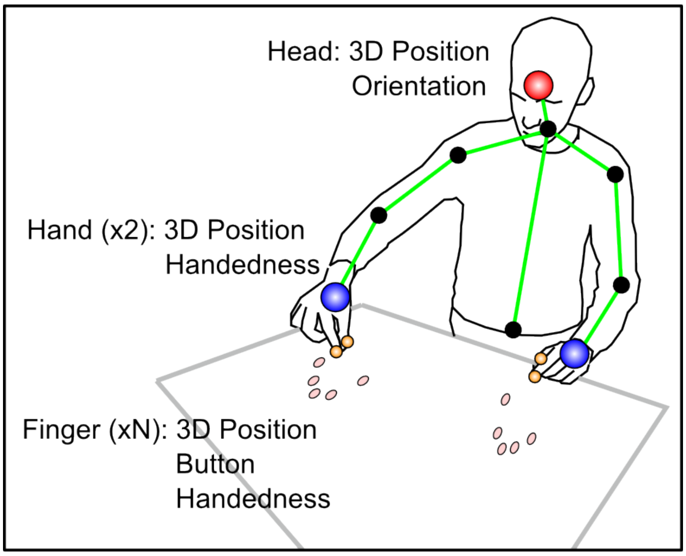


Figure 4.7: Mockup Builder User Model in a continuous interaction space. The Kinect camera captures the user skeleton (in green) defining both the head pose (in red) and hands information (in blue) . The Mulit-touch surface and the Gametraks are able to track user fingers on the surface and above it. Finger information is the main input in our user interface and is enriched with the information from the skeleton, the surface definition and the button in the air to start gestures.

by the Kinect, both thumb and index fingers of each hand are tracked by the two Gametrak devices in the air. Finally, finger positions of all touches on the surface are gathered by the multi-touch display sensor. Combining this information, we are able to identify the handedness of all the fingers on the surface and both thumb and index fingers in the air. The fusing of inputs also allow to be aware when the thumb or index finger is leaving or touching the surface. Finally, the Gametrak device can inform that the thumb finger of each hand is pressing or not the ring button corresponding to an explicit pinch gesture.

4.3 Implementing our 3D User Interface

While the previous sections presented how input data is gathered and fused to define our user model, this section presents how we devise our application logic to use such model and implement our 3D modeling user interface. We start to present our graphical user interface relying on contextual menus to complement the direct manipulation with additional modeling operators. Then we explain,

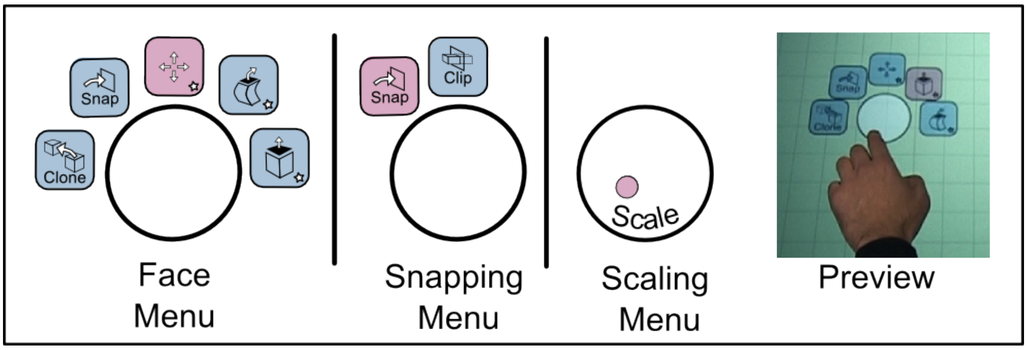


Figure 4.8: Examples of contextual menus used by the MockupBuilder application. Icons using a star symbol describe radio buttons defining the default modeling operation assigned to a face. Red colored options indicate activated modes when referring to both toggle and radio buttons. The image on right presents a preview of the graphical menu in our semi-immersive environment.

how the user input model is used to combine all the input modalities and propose a consistent user interface based on the interaction concepts presented in Chapter 3.

4.3.1 Graphical User Interface

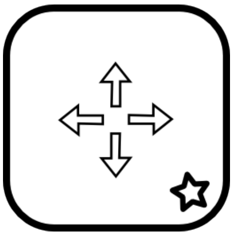
We use a circular menu layout with just one level hierarchy, positioning the options close to user fingers since the menu position on the surface is below the hand position as depicted by the preview image from Figure 4.8 . Menus are invoked by approaching the NDH to the interactive surface while performing a selection or an action using the DH . Such configuration only enables using NDH fingers to select options. Depending of the action or the selection performed by the DH , the content of the menu is adapted to propose only the valid options which can be done at the current moment. Figure 4.8 presents three examples of contextual menus. The first appears when the user is highlighting a face with the DH showing most of the option available in our application. The second example is invoked by an empty selection, i.e. touching the surface with DH when a snapping operation have been performed allowing to exit from this mode. The third example appears if invoked by the user while performing an extrusion to scale the profile of the extrusion by moving the circular manipulator.

Our menus use very simple dialogs, most of the available options are simple push buttons. However, we also use some buttons to give feedback of the current operation assigned to a face or if a given mode is activated or not. This is done using simple radio button identifies with a star symbol or using a coloring scheme to inform the state of activation. The Table 4.1 describes the different options and their functionality used by our graphical user interface.

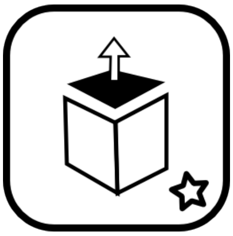
4.3.2 Putting all together

The interaction techniques previously described are supported by fusing input events from the multi-touch surface, Gametrak and Kinect and then handling strokes. Strokes for the multi-touch surface are defined by the sequence of points starting from the time the user touches the surface with a finger until that finger leaves the surface. For the Gametrak, a stroke is defined by the sequence of points with the same button state, for example after the pinch button is pressed and until it is released. Kinect inputs are only used to update the hand position information and the head tracking needed for the stereoscopic visualization. Figure 4.9 presents the pseudo code of our event update function which is called at each input device update. This function defines our application logic based on the changes of the fused input representation. For each new input event, we update the state of the contextual menu and its position depending on shape features selected and the distance between the non dominant hand and the surface. Then the fused input representation is updated with the new input and both commands and actions are fired according to the type of changes. Three types of changes regarding fused inputs can occur: a new fused input is generated, an existing fused input is updated or an existing fused input is deleted. Based on such definition, we use a lookup table to define the correct action or command to be fired by the application.

Table 4.2 presents an overview of the lookup table. It describes how our modeling technique is controlled based on the status of the fused input event (new event, updated event, deleted event) and state information of the application. For the sake of clarity, Table 4.2 does not represent the activation of constraints used for symmetric drawings and constrained height extrusion , which are applied by updating dedicated variables in the application. Regarding menu based commands related to shape features, we store the information directly with shapes

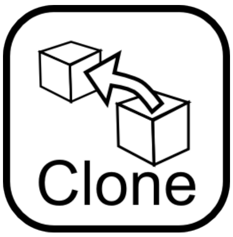


Option available when a face is selected to activate the moving mode of a face. The moving operation allows to push or pull a face along its normal direction updating the geometry definition of the face without any topological change. This option is part of a radio button with the following two option controlling the state of a face.

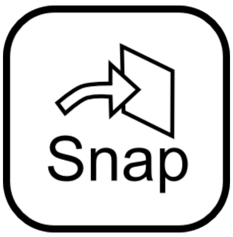


Option used to activate a linear extrusion of a face. By default, any newly created face is set to the linear extrusion state. After starting the extrusion which updates the topology of a shape, this option is automatically deactivated and the move state is set to the face. By using this option repetitively on the same face, successive extrusion can be performed by the user.

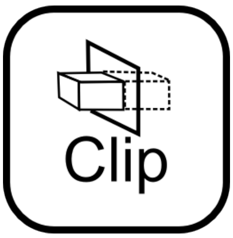
Option offering the curvilinear extrusion of a face updating both the topology and the geometry of a shape. This is the third possible state which can be assigned to a face. When activated, the curvilinear extrusion can be performed on the face and is valid until the option is deactivated using the face menu.



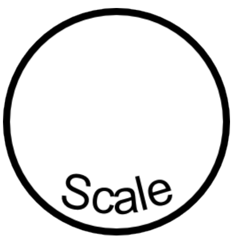
This option is available when a face or a shape is selected using the highlighted bounding box. It allows to clone the object. The resulting clone is created when the user press the option at the center location of the menu on the interactive surface.



This option allows to snap or unsnap from a face, aligning the model face to the surface. Such operation allows to add details on a face and is only available when a face is highlighted. It behaves as a toggle button allowing to get in or back from the snapping mode.



This option is available when a snapping is activated allowing the user to clip the scene out of the surface since the reorientation of model can put unwanted occluding part on top of the snapped face. It acts as a toggle button.



The scaling option is a manipulator available when the user is performing a linear extrusion. To control the scaling factor the user drags the manipulator controlling the distance regarding the menu center.

Table 4.1: Menu options available to support modeling tasks in the Mockup-Builder application

together with their geometric representation. This is represented by the face state column in Table 4.2 where MOVE, LEXTRUDE and CEXTRUDE values represent a shape to be moved, extruded along a normal or along a curve respectively. Thanks to this solution, we are able to deal with the different input devices update frequencies while abstracting the 2D and 3D nature of the input.

```
Input: An event from a given input device : MultiTouch, Gametrak or
       Kinect
if event.type== KinectEvent then
    handpositions ←UpdateHandPositions(event) ;
    headposition ←UpdateHeadTracking(event) ;
UpdateMenu(currentSelectionDH,handpositions);
if event.type==MultitouchEvent then
    forall the input 2D of event.newInputs do
        ComputeHandeness(input 2D,handpositions);
    end
    RefreshFusedSource(fusedInput,event);
if event.type==GametrakEvent then
    RefreshFusedSource(fusedInput,event);
if fusedInput.deleteChanged then
    forall the input of fusedInput.deleteInputs do
        ReleaseInputSelection(input);
    end
if fusedInput.newChanged then
    forall the input of fusedInput.newInputs do
        ProcessInputUsingLookupTable(input);
    end
if fusedInput.updateChanged then
    forall the input of fusedInput.updateInputs do
        ProcessInputUsingLookupTable(input);
    end
if fusedInput.deleteChanged then
    forall the input of fusedInput.deleteInputs do
        ProcessInputUsingLookupTable(input);
    end
```

Figure 4.9: MockupBuilder UpdateEvent pseudo code defining the application logic and triggering commands and actions depending of the events gathered from the different input devices.

4.4 Processing Gesture and Sketches

This section describes how fused input data is processed by Mockup Builder. As explained before, such inputs are converted into a stroke representation which can represent a sketch or a gesture. While gestures should describe smooth motions and trajectory, sketches need to be processed allowing users to draw easily line and curve drawings. Drawing improvements are also required to easy users to draw regular figures such as squares, rectangle, circle and ellipses using their finger on the surface. This is done combining smoothing techniques with sketching beautification and 2D recognition of drawing primitives from simple ones such as line or curves to regular 2D shapes and complex drawings combining line and curves while guaranteeing geometric properties such as tangencies, perpendicularities and parallelism.

4.4.1 Gesture, Motion, Stroke Representation

All our input data are presented as sequences of 3D positions originated by touches or Gametrak data. 2D touches are converted to 3D positions using the surface definition on the Kinect coordinate space. Gametrak 3D positions are converted using the calibration matrix from the Gametrak local coordinate space to the Kinect one used as our primary space. For each stroke input, we are able to infer the sequence of 3D positions, its handedness and the provenience of the input i.e. 2D from touches or 3D from Gametrak devices. Raw input data is also available into our stroke input representation since it can be preferred for 2D recognition for example.

Given any input state change, three stroke changes can occur regarding fused inputs: a new fused input is created, an existing input is updated or an existing input is deleted. Each of these changes will trigger a different action such as updating a visible stroke, smoothing a stroke description, perform a 2D shape recognition or a stroke beautification. The following subsections describe each of these steps.

4.4.2 Interactive Input Smoothing

To be able to use fused input as sketches or gestures, we first need to smooth the data and convert it to a higher level input representation. Depending of the input source, input points are filtered to remove any noise or jitter inherent from the input device. Such step is particularly important for the Gametrak data but it is not needed for touches since they are already stable enough thanks to the vision based blob detection algorithm. However, we still need a smoother and more compact representation as an alternative to the raw sequence of points.

When a new fuse input is created or updated, we interactively fit the input data into a sequence of piecewise cubic Bézier curves. Such representation allows to represent line segments, curves as well as control the smoothness between these primitives. We use a 3D adaptation of the 2D Cubic Bézier fitting algorithm, proposed by [Schneider, 1990], to create curve approximations according to a thresholded fitting error. Our fitting error segments the input data into curves to guarantees an error up to two millimeters. We remember that thanks to our calibration process, any input data is converted into physical measurements independent to the input resolution or visualization resolution. Given a sequence of points and positional conditions at the extremities, the Schneider's fitting algorithm is able to compute the best four control points of the cubic Bézier representation. Tangency conditions can be specified allowing to control the smoothness between primitives. At each stroke update, the fitting is recomputed for the active segment while the fitting error is valid. First we try to fit the input points to a line, then to a curve according to the current fitting error. When the maximum allowed fitting error is reached, a new segment is created and the last valid fit is used to approximate the active segment of the stroke. By doing so, we are able to convert the stroke into a piecewise cubic Bézier curve representation ready to be used as gesture or for stroke recognition or beautification. The piecewise curve description provides a compact and smoother stroke representation as well as a well defined normal direction of each point. Depending of the control point positions i.e if they are close to be aligned, Bézier segment curves are interpreted as line or curve segments. Figure 4.10 presents the code of the stroke updating functions. At each new input 3D position, the piecewise representation of the stroke is automatically updated with new line or curve segments. Our implementation use the same threshold value to accept both curve and line fits. We use a value corresponding to a distance of 7 millimeters on the surface. Such value seems to be adequate since the user is using its fingers instead of a pencil to draw on the surface or in the air.

Input : A 3D point inputPoint3D to be added to the stroke representation.
StrokeMember: The stroke representation is defined by the two vectors points and curves and the variable startindex initialized to zero.

```
points.pushback(inputPoint3D);
if points.size()-startindex >2 then
    lineFit =Line3D(points [startindex ],points [points.size()-1]);
    lineError =maxDistError(lineFit,points,startindex);
    currentFit = NULL ;
    if lineError <errorLineThreshold then
        currentFit =lineFit;
    else
        tangent =Vector3D(0.0,0.0,0.0);
        if curves.size()>1 then
            tangent =curves [curves.size()-2].getTangent(1.0);
        curveFit =fitCubicBezier(points,startindex,tangent);
        curveError =maxDistError(curveFit,points,startindex);
        if curveError <errorCurveThreshold then
            currentFit =curveFit;
    if currentFit ! = NULL then
        if curves.size() > 0 then curves.popback();
        curves.pushback(currentFit);
    else
        startindex =points.size()-2;
        if curves.size() > 0 then
            pt =curves [curves.size()-1].getPosition(1.0);
            curves.pushback( CubicBezier(pt,pt,pt,pt));
```

Figure 4.10: Stroke fitting algorithm updating the stroke representation with a new 3D point. Internal "curves" vector is updated automatically with lines and cubic Bézier curves creating a piecewise representation of the stroke. The internal "points" vector is updated with the raw sequence of 3D points. Both line and curve fits are accepted if the maximal distance is below a given threshold value.

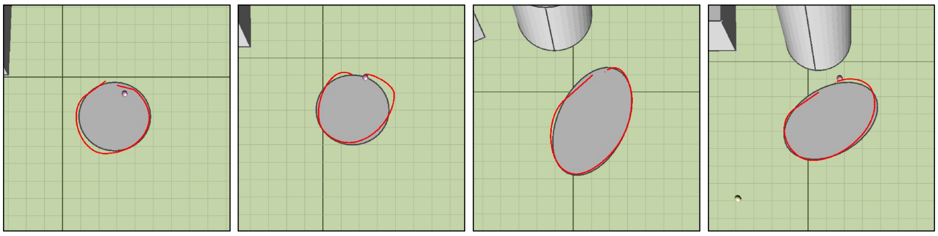


Figure 4.11: Examples of screenshots showing conic results from the CALI recognizer : two circles and two ellipses. The red stroke was sketched by the user and submitted to the recognizer.

4.4.3 Constraint Based Sketch Beautification

While we chose to rely on the CALI 2D recognizer for regular conic primitives such as ellipses and circles as illustrated by Figure 4.11 examples, all other 2D regular primitives such as squares, rectangles or right-angled triangles can only be obtained by sketching followed by our sketching beautification technique. This technique tries to regularize the drawing and enforces detected geometrical constraints from the drawing. By doing so, we allow complex line and curve regular drawings as well as simple primitives such as right triangles, squares and rectangles as shows Figure 4.12.

To beautify an existing drawing, we start by detecting the constraints to be fulfilled by the beautification process. Our approach focus on regularizing line segment parts of the drawing. We detect constraints based on threshold values of both relative angles between line segments and segment length ratios. First, we compute all the vector direction of each segment and its length. Then stroke segments are compared to each other computing the length ratio between seg-

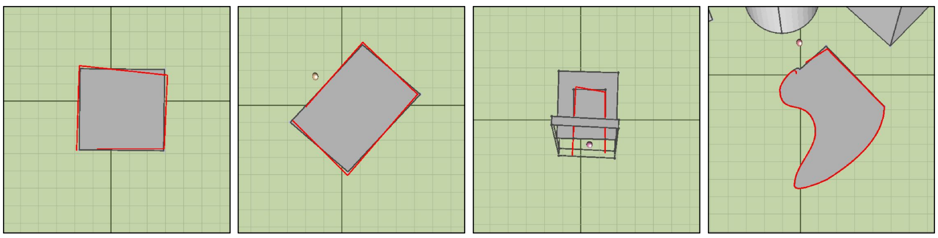


Figure 4.12: Examples of screenshots showing the user stroke colored in red and the resulting planar face created by the beautification process: a square aligned with the canvas axis, a rectangle, an open stroke used to add detail on a face and a shape using both line segments and curves.

ments and the relative angle based on the dot product of both vector directions. Using different threshold values, the following constraints are detected to define a problem to be solved improving our drawing:

- Perpendicularity between consecutive line segments,
- Parallelism between non consecutive line segments,
- Co-linearity between consecutive line segments,
- Co-linearity of an edge with a predefined direction,
- Adjacency of line segments followed by unconstrained segments,
- Edge pair with same edge length.

After detecting all possible constraints on the current drawing, we define a non linear problem to be solved. Our constraint solving approach is based on energy minimization. For each type of constraint, we define an energy function based on vector calculus returning zero if the constraint is fulfilled. Each energy function receives as parameter the positions of the line extremities involved in the constraint definition. Our constraint problem receives a vector containing all the points of each stroke primitive and a set of energy function to be fulfilled by the input points. The minimization is performed iteratively until the global energy function is minimized returning the final position of each point. The error minimization is performed using a conjugate gradient method since all energy function have a well defined derivative. Finally, we update all the stroke segments using the position information from the returned output vector.

4.5 Creating and Editing CAD Shapes

Figure 4.13 shows the block diagram of our modeling architecture. Our solution relies on the manipulation of a shape representation based on user interactions. The user interacts in our semi-immersive environment through direct manipulation, gestures and sketching over the shape visualization. Inputs are

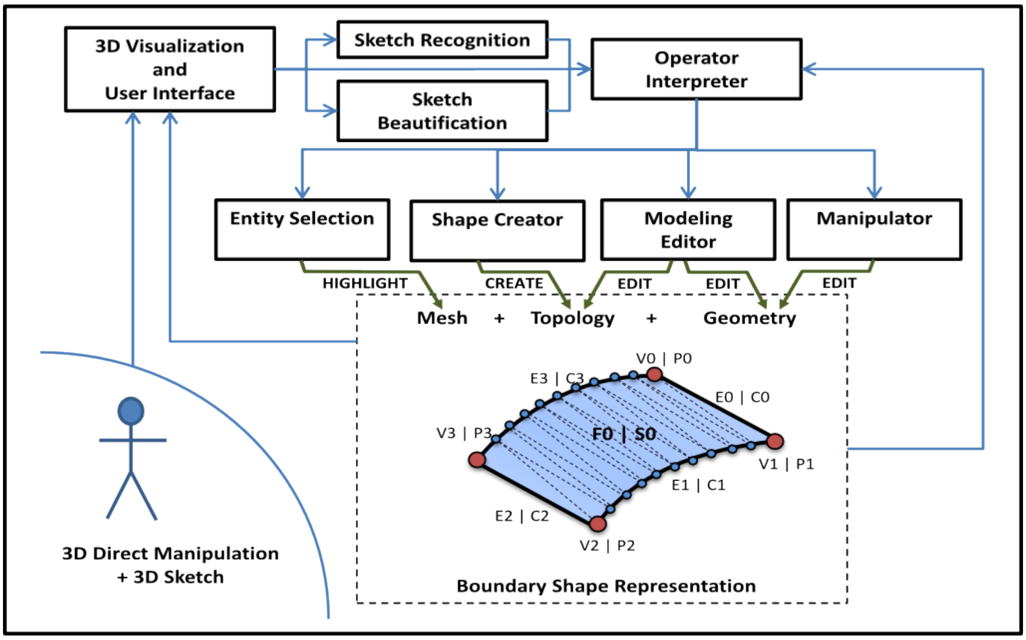


Figure 4.13: Our modeling architecture around our shape representation.

gathered from the different input devices and processed to be interpreted as modeling commands to create, edit and manipulate 3D shapes. As mentioned before, inputs are processed as sketches or gestures represented as 3D or 2D strokes which are processed using sketching recognition and beautification. Combined with the shape information co-located with the interaction space and our user interfaces, we are able to specify our modeling commands to support modeling tasks. Finally, All the modeling commands result into updates on our shape representation.

Our shape description relies on a boundary representation allowing to describe 3D volumes delimited by faces. This representation is divided into topology, geometry and mesh information used for graphic purpose and is described in detail in the remaining of this section. Both planar and curved faces can be represented and are delimited by edges which can be line or curve segments. Such representation can be created and edited through user sketches combined with direct manipulation of both shape geometry and topology. This is done by four different modules. The first is the Entity Selection module responsible to identify which parts of the shape are intersected by user gestures and considered by the modeling operation. It provides feedback to the user of the selection by updating the color of the mesh representation. The Shape Creator module is responsible to create initial shape representations based on sketches. Shapes can be edited by Modeling Editor module which translate user gestures into modeling operations. Both creation and push and pull modeling operators in particular the dynamic curvilinear extrusion are detailed in this section. These modeling operations can result into topological or geometrical changes or deletion of our shape representation. Finally, the Manipulator Module is responsible to apply spatial transformation which might update the geometrical representation of the shape since it where the position information of the shape is stored.

4.5.1 BREP Shape Representation

The boundary shape representation consists on a graph of vertexes, edges and faces delimiting the volume of the shape. Based on the representation used by existing CAD kernels and presented by [Stroud, 2006], we implemented our own graph based shape description separating the geometry from the topological information of the shape. We choose to create our own boundary representation to provide interactive manipulations which would be out of the scope if reusing an existing solution such ACIS[5] or OpenCascade[6] or interfacing with an existing CAD application through scripting or available Application programming interface (API). BReps extend simpler data structures such as the half-edge data structure (also named double connected edge list) [Weiler, 1985] by providing entities such as shell, wire and loop to handle non-manifold objects. Our boundary representation is based into two parts, a graph of topological features and a set of geometric descriptions referenced by the topological graph as illustrated by Figure 4.14.

The topological entities are the following:

- Vertex: Topological representation referencing a point on the shape geometry and belonging to a set of edges;
- Edge: Topological representation of a curve geometry delimited by two vertexes;

[5] Spatial 3D ACIS Modeler: `http://www.spatial.com/products/3d-acis-modeling/`
[6] Open CASCADE Technology, 3D modeling & numerical simulation: `www.opencascade.org/`

- Coedge: Oriented Edge belonging to a parent loop or wire, it allows to navigate along adjacent coedges;
- Loop: Representation of a sequence of coedges belonging to the same face and allowing to navigate to other loops of the face;
- Face: Topological representation of a geometrical surface delimited by a loop and belonging a topological shell, it also reference an adjacent face to navigate on the shape topology;
- Wire: Representation of a sequence of coedges which do not belong to a face but are part of the same shell;
- Shell: Topological representation of a shape which might be a set of faces or wires, it also reference an adjacent shell, if the shape is represented by several shells.

The geometrical information of the shape is represented by geometric entities references by the topological graph. The following three geometrical entities are used by our boundary representation:

- Point: represents a 3D position by its coordinate in space or defined by a parameter value along a parametric curve;
- Curve: represents the geometry of an edge defined by a parametric curve;
- Surface: geometrical information of a shape which might be associated with a planar definition if it is planar.

As mentioned before, our sketch based interface relies on piecewise cubic Bézier curves to represent user strokes. Such parametric curve representation is also used for shape edges as we will detail in Section 4.5. The curve geometrical entity abstracts the curve from its parametric representation. By doing so, it can represent both shapes and strokes allowing to reuse the current boundary representation to handle other type of curves. Our surface geometrical entity is only used to store surface attributes such as the planar definition of a topological face. The surface definition is represented by the boundary edges associated to

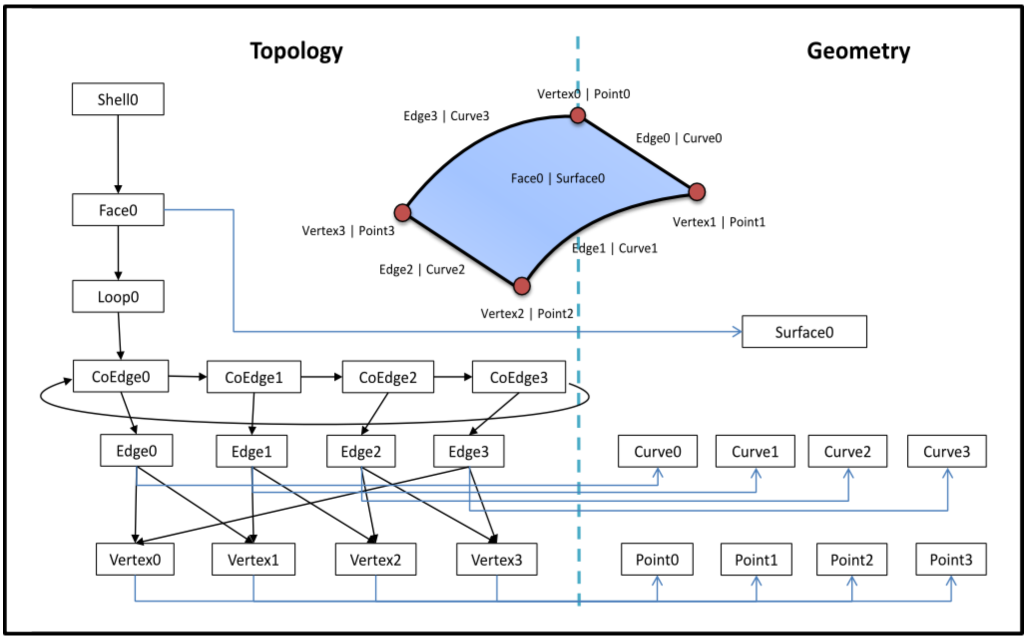


Figure 4.14: Example of our boundary representation defining the topology and the geometry of a simple curved surface.

the topological loops of the face. When at least one of the edges is represented by a planar curve, we use a Coons patch [Farin, 2002] to define the curved surface based on its boundary edges.

4.5.2 Mesh Shape Representation

Based on the topological and geometrical representation presented in the previous section, a mesh representation of the shape is generated to be visualized stereoscopically by the users. This visual representation needs to be consistent with the shape representation. It is updated after each modeling operation or selection from the user. We use an indexed mesh to render each of the main topological entities of a shape, i.e. vertexes, edges and faces. The indexed mesh uses a data buffer to store vertex positions, normal directions and colors obtained by the discretizing the shape definition. Such information is used by a tessellation of the shape which is represented by an interleaved index buffer to define it graphic primitives. We use common graphic primitives supported by any graphic API such as points, lines and triangles to represent our vertexes, edges and faces respectively. Figure 4.15 presents a diagram of a face example describing a curved surface. For a given topological representation of a shape, our mesh representation relies on a *Topology to Mesh* map, and interleaved index buffer and a data buffer.

To manage and reduce the need of tessellating the complete shape after each modeling operation, we associated a dirty flag to each topological entity of the shape. This flag is updated when performing a modeling operation to create or edit the geometrical and topological representation of the shape. Depending on the modeling operation only geometrical or topological parts of the shape representation are updated. After each modeling operation, we traverse the boundary representation to list all the updated topological entities. For each updated topological entity, we create or edit an indexed mesh representation. The *Topology to Mesh* map is used to bind the mesh representation to topological entities. It allow to identify if a given topological entity needs a new mesh representation or to access to its current mesh to be updated. Depending on the type of topological entity, the mesh information can be composed by one or more point primitives. For example, a vertex has only one point to represent it. However, an edge which is bounded by two vertexes, can have new points corresponding to the internal tessellation of a parametric curve. Faces representing a planar shape can be described using only the corresponding points of both edge and vertex entities. On the other hand, curved surfaces will be defined by the indexed of the triangulation plus the additional points needed to tessellate and correctly match the interior of the surface. While performing the tessellation of an edge or a face, normals vectors are computed for each new point and stored in the mesh data buffer.

To enable the visual feedback of selecting an entity, a vector of eight colors is stored in the data buffer of each mesh. Each of these colors corresponds to a different selection granularity, i.e. a vertex of an edge, the edge from a face or a complete face. The interleaved index nature of our mesh representation allow to reuse points, normals and colors while reducing the memory needs. By dividing the shape representation into topology, geometry and mesh information, we provide a compact graphical representation while it is flexible enough to support our modeling operations. This separation guarantees a solid implementation of the modeling operations while allowing dynamic content creation and interactive visual feedback to users.

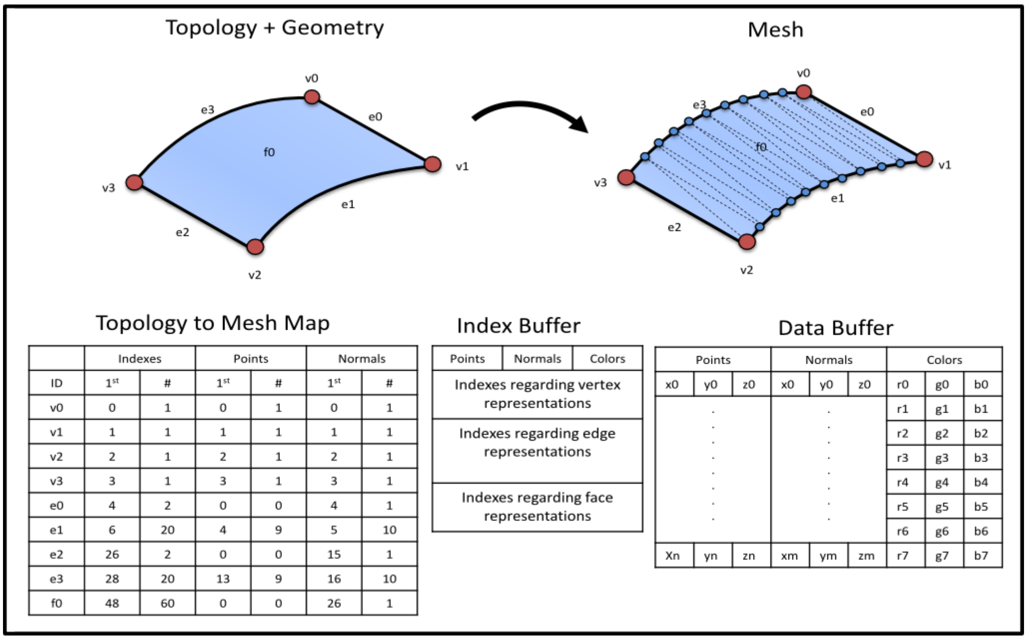


Figure 4.15: Mesh representation and its relationship to both the topological and geometrical representations. We use an indexed mesh to represent the graphical representation of the shape. The ”Topology to Mesh” map enables to find and update the corresponding mesh representation of any topological entity.

4.5.3 Direct BREP Modeling

In order to devise a 3D modeling system, the most important steps of our push and pull approach are the creation of face, the selection of topological entities and the extrusion operator. Planar shapes are created using the piecewise cubic Bézier curve from a closed stroke sketched by the user. Each curve of the piecewise representation originates a new topological edge and the both first and fourth control points define vertexes delimiting the edge. Then the edges are used to create oriented coedges and a topological loop to be binded to a new topological face. Finally a single shell is created using the topological face resulting on the topological representation of the shape. The surface of the face is delimited by the newly created edges as a coon patch using the parametric curves. For each new created face, we assign a new mesh representation tessellating the planar surface.

When an extrusion of a face occurs, new faces are created updating the shape topology. Given our face description, we use the existing edges to define new lateral faces resulting on new edges and a new top face. Finally, we remove the old face since it has been replaced by the new lateral and top faces ending the extrusion modeling operation.

To move an existing topological feature, the selection identifies the topological primitive by traversing the topological representation of the shape. Selected topological features are defined by proximity using the geometrical information or the mesh information when it is most convenient. The result consists on a vertex, an edge or a face which can be highlighted since we can access to the mesh data based on topological representation. To direct manipulate the shape features, we only need to change the geometrical information and then update the mesh representation. When a vertex is updated, all the coincident edges and faces are automatically updated notifying our mesh tessellation algorithm thanks to the dirty flag. Following the same approach, when edges or faces are moved, the topological graph allows to derive vertexes, edges and faces which need to be updated. Such approach allows to provide interactive modeling operators.

4.5.4 Dynamic Shape Generation

While most of our modeling operators are implemented as topological changes or as geometrical changes, the dynamic curvilinear extrusion operation is more complex combining both types of changes for each user update. The main problem is that the input stroke is always updates resulting into interactive both topological and geometrical changes. The curvilinear extrusion of a face is done along the piecewise cubic Bézier continuously updated by user hand motion. Such operation is more complex than the straight extrusion since we need to offset the curve to create lateral free-form surfaces instead of just using linear edges.

While the user is defining the trajectory of the curvilinear extrusion, new curves are added to the piecewise curve representation of the hand motion or the last curve segment is being reevaluated. For each new curve segment, we use a modified implementation of the [Coquillart, 1987] curve offsetting algorithm. Given the current planar shape to be extruded along a curved path as depicted by Figure 4.16a, we compute the barycenter of the face (Figure 4.16b) and each vertex of the face is defined using a polar coordinate system. The axis of the polar

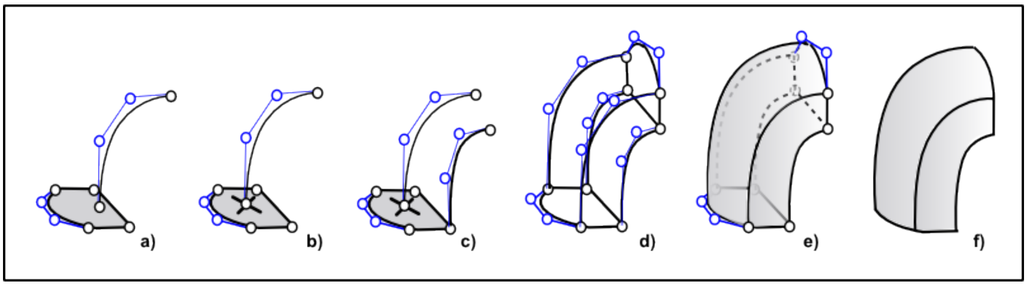


Figure 4.16: Curvilinear extrusion steps along a cubic Bézier of a planar face. Black circles describe edge extremities while blue circles are interior control points of curved edges. The curved path is translated to the barycenter of the face indicated by the cross, then offset curves are computed to generate the extruded surface.

coordinate system are defined based on the Frenet curve reference system which use the derivative of the parametric representation. The Frenet frame is defined by the tangent, the binormal and normal of the curve for each point along the curve. Given a curve segment , we translate its control points to the face vertex locations creating a geometrical approximation of the offset curve (Figure 4.16c). This is done for each vertex of the planar shape (Figure 4.16d) defining the curves delimiting the extrusion. The resulting curves are then used to defined both lateral and top faces of the extrusion (Figure 4.16e). Finally, the new shape and its surfaces are tessellated and presented to the user as depicted by Figure 4.16f.

For each hand motion update, the curvilinear extrusion will invalidate the last curve segment and generate a new curve offset based extrusion or create a new extruding part if a new curve segment is added to the piecewise stroke description. By doing so, the interactive curvilinear extrusion updates only the last segment of the extrusion allowing to achieve interactive framerate while generating smooth free-form surfaces.

4.6 Summary

In this chapter, we have presented our hardware modeling environment as well as how the different input device are used to define a single user model in a single reference space. Then we describe our sketch based modeling approach focusing in the sketching processing to enable user to create rigorous shapes based on sketches drawn using touches and gestures in space. Finally, we describe how modeling operator are implemented to create complex 3D shapes interactively by direct manipulation.

Condition								Result	
Input Event	**B.**	**Handedness**	**Input Usage**	**Selection Status**	**NDH Selection**	**Face State**	**Stroke Shape**	**Action**	**Input Usage**
New	Y	-	-	onWidget	-	-	2D	WidgetDown	WIDGET
New	-	-	-	!empty	-	-	-	Add Selection	-
Update	N	-	-	!empty	-	-	-	Add Selection	-
Update	-	-	CEXTRUDE	-	-	-	-	Update CurvedExtrusion	-
Update	Y	DH	NONE	-	-	-	-	Update Stroke	-
Update	Y	DH	!WIDGET	!empty	-	MOVE	-	Update LinearMove	MOVE
Update	Y	DH	!WIDGET	!empty	-	LEXTRUDE	-	Update LinearExtrusion	LEXTRUDE
Update	Y	DH	!WIDGET	!empty	-	CEXTRUDE	-	Update CuvedExtrusion	CEXTRUDE
Update	Y	1^{st} NDH	NONE	!empty	-	-	-	Translate Object	-
Update	Y	1^{st} NDH	NONE	empty	-	-	-	Translate View	-
Update	Y	2^{nd} NDH $\|\|NDH \geq 1$	NONE	!empty	empty || !empty	-	-	RotateScale Object	-
Update	Y	2^{nd} NDH $\|\|NDH \geq 1$	NONE	empty	empty	-	-	RotateScale View	-
Delete	-	-	CEXTRUDE	-	-	-	-	End CurvedExtrusion	-
Delete	-	-	LEXTRUDE	-	-	-	-	End LinearExtrusion	-
Delete	-	-	MOVE	-	-	-	-	End LinearMove	-
Delete	-	-	WIDGET	-	-	-	-	WidgetUp	-
Delete	Y	DH	NONE	-	-	-	2D Delete Gesture	Deleting Overlapping Content	-
Delete	Y	DH	NONE	-	-	-	Closed Stroke	Planar Shape Created	-
Delete	Y	DH	NONE	-	-	-	Open Stroke	Split Overlapping Faces	-
Delete	Y	DH	NONE	-	-	-	Other	Create Simple Stroke	-

Table 4.2: Mockup Builder Input Lookup Table showing main functionalities for each input modalities represented as the fused 2D/3D input events. Each row represent a set of input conditions and the resulting action or input state change. The B. column defines if the Gametrack buttons are pressed. It is considered as pressed for any surface touch. The input usage column is a state which can be defined for inputs. Empty cells indicate indifferent conditions or unchanged values.

5
Empirical Results

This chapter describes the empirical evaluation of our 3D modeling approach using the Mockup Builder prototype. While the prototype have been experimented informally along its design by 20 undergraduates student, we performed a preliminary evaluation comparing Mockup Builder to Rhino 3D with 2 experts. Then we continue with a formal user evaluation with 14 participants to compare Mockup Builder to Sketchup 8 in different modeling scenarios. Following human computer interface evaluation methodology, we selected a set of both novice and expert users mainly from the architectural field without any experience our semi-immersive modeling environment. However, most of the users were familiar with modeling tasks in its daily basis and already used a wide variety of modeling systems. We choose to use the Sketchup system since it is representative of the push and pull modeling metaphor and it is widely know for its simplicity and easiness of usage. This chapter presents and discusses the results of this evaluation by devising the benefits and areas for improving our Mockup Builder prototype.

5.1 Baseline User Experiments

We implemented a prototype to demonstrate our modeling approach in C++ using OpenGL[1] and OpenSG[2] for stereoscopic visualization. Our system was deployed on an Intel I7 920 2.67 GHz processor with 3 Gb of memory RAM and an NVidia Quadro 4000 graphics card running Microsoft Windows 7 64-bit operating system. Along the development, around 20 undergraduate and graduate students

[1] OpenGL API:`http://www.opengl.org/`
[2] Open Source portable scenegraph:`http://www.opensg.org/`

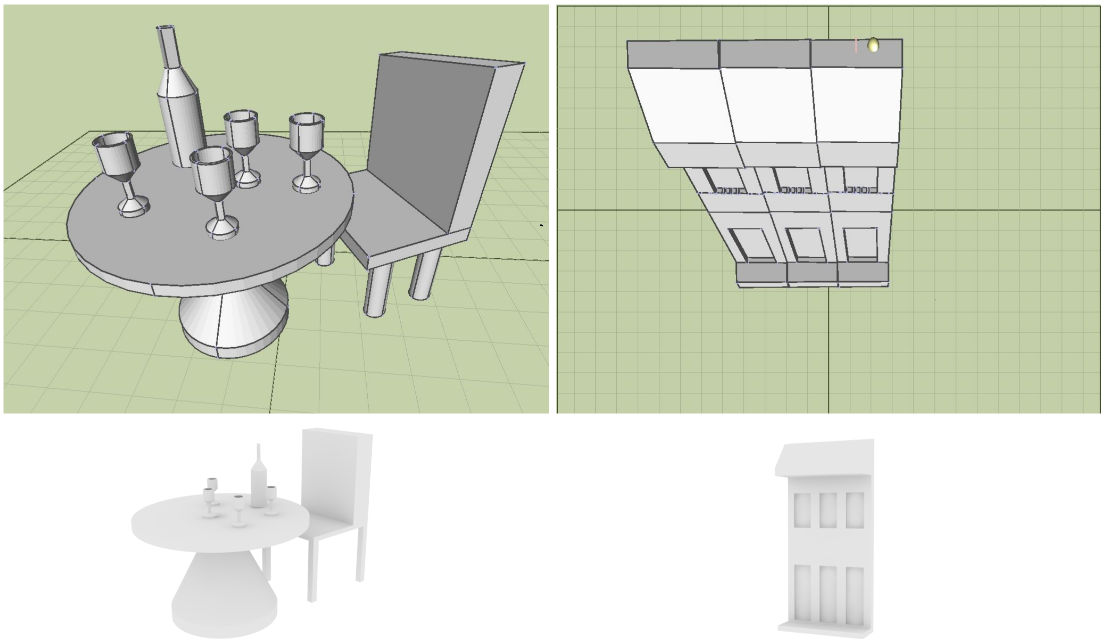


Figure 5.1: 3D models designed using Mockup Builder (top row) and Rhino 3D (bottom row): a table with a chair and a simple building façade. The bottom models were created by a professional architect based on the upper screenshots.

in Computer Science with variable experience with CAD applications and one Architectural researcher tested the system. They informally assessed the different design choices and iteratively improved the design of the interface. Thanks to stereo, we provide co-location between user hands and virtual objects adapted to direct modeling methods. While an initial version of our system used physics to detect collisions, this proved to be problematic when modeling. The problem was related with the physical simulation instability since user hands could easily interpenetrate virtual objects due to the lack of physical force feedback. Since it was difficult to meet user expectations while guaranteeing a stable physical simulation, this feature was removed in the subsequent versions. However further informal tests indicate this could reveal to be advantageous both for stacking and supporting 3D manipulations or assembly tasks as shown by [Fröhlich et al., 2000]. In order to create an initial baseline to design the user evaluation specially regarding timing performances compared to existing CAD applications, I created two models using most of the operators provided by Mockup Builder. These models are illustrated in the first row of Figure 5.1 representing a set of furnitures and tableware on one side and a simple building façade on the right side. While the left model was built in 5′20″, the second model was created in 2′45″. Then, we asked to a professional architect with 5 years of experience in several architectural offices as a 3D architectural modeler freelancer to built both models using his preferred modeling system based on just these two screenshots. The expert user chose the Rhino3D[3] modeler from Robert McNeel & Associates since it was the system he used on a daily basis. The user stated that he preferred this tool to other popular tools among architect such as AutoCAD[4] from Autodesk due to its easiness to define both curved and flat 3D surfaces. This expert user took 5′41″ and 3′46″ respectively to generate models as detailed as the initial screenshots. We should note that while the timing are similar, the modeling approach was different. MockupBuilder proposes a push-and-pull approach fostering consecutive extrusion operations while the expert user took advantage of symmetrical properties and revolution operation to generate the glass model. While these results were encouraging, it allowed to identify a set of models that could be used for a comparative user evaluation with a good trade-off between modeling complexity, details and geometric primitives. However, it also indicated that a professional CAD system should be chosen among the existing one which would be known by most of the participant. In addition, such system should rely on a modeling approach similar to the one offered by Mockup Builder, i.e. a push-and-pull modeling systems would be preferred for the comparative study. The following section presents a deeper evaluation of the system to better assess timing comparison with other systems, the modeling capability of our approach and the influence of our semi-immersive environment to support modeling tasks.

5.2 Formal User Evaluation

A formal within-subject evaluation of Mockup Builder was performed with 14 participants with an average duration of 1 hour and 31 minutes per user with standard deviation of 28 minutes. The goal of this evaluation was to compare and to assess the benefits of both our modeling approach and the user interface offered

[3] Rhinoceros 3D Modeler:`http://www.rhino3D.com/`

[4] AutoCAD 3D CAD Design Software:`http://www.autodesk.com/products/autodesk-autocad/`

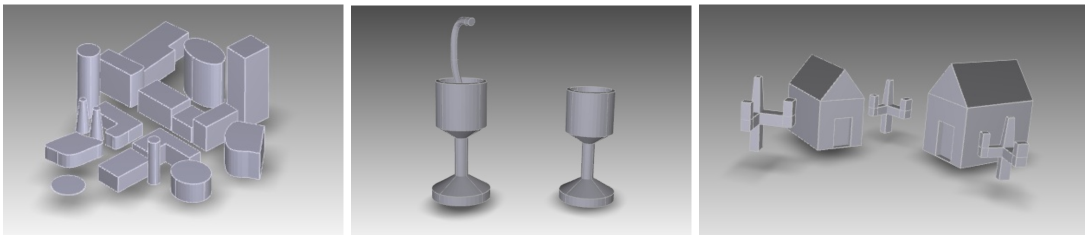


Figure 5.2: Screenshots of the three tasks performed during the user evaluation.

by our system. In order to perform such a study, the experiment was designed as a comparison of modeling tasks with our system and an existing commercial modeling application. Since Mockup Builder relies on a push-and-pull modeling metaphor and mainly it targets architectural and design users, we choose Google Sketchup as a representative of CAD modeling systems with a low learning curve compared to other more complex systems. Google Sketchup has become a popular tool specialty to create 3D content. It is also used by students in architecture as a teaching tool or advanced architects to roughly create 3D models instead of using more complex systems. While its interface is not fully representative of CAD systems based on four views and support of several geometric representations, it offers the opportunity to question several design choices presented by our interface, i.e. the stereoscopic visualization, mixed sketching and push-and-pull approach, the usage of gestures to define extrusions and the bi-manual interaction model. Table 5.1 presents a comparison between both systems and the scenario used during the evaluation.

The evaluation consisted in the comparison of three modeling tasks with variable complexity on both systems. For each task, two screenshots of the expected scenes to model were presented with information about specific items to be fulfilled by participants. No length measure was provided in order to focus on the modeling approach and the screenshots were exactly the same for both systems. The order of presentation of each system was counterbalanced across participants , i.e. a group of participants started the evaluation using Mockup Builder followed by Sketchup, and second group did the test in the opposite way. Figure 5.2 presents one screenshot of each task. The original size of the screenshots can be seen in Annex C. The first task consisted in the creation of a city using mass models around imaginary streets represented by a variety of extruded shapes. It was requested to experiment several profiles with regular or irregular shapes, at least one truncated cone and one of the object had to be an exact copy of an existing shape. The second task consisted in the creation of a glass which should be cloned. The duplicated glass had to contain a straw modeled by the user and placed correctly inside the glass. The last task was a scene representing a simple house with a roof and a front door and an object similar to a tree or cactus like shape. The composition of the scene had to fulfill the placement illustrated by the screenshot.

For each system a brief demonstration was performed by the facilitator leading the experiment on each system, followed by a practice session for the participant to become familiar with the modeling interface which we will discuss in Section 5.2.2. Then the three tasks were performed sequentially while the participant was videotaped, logging of all actions performed by the user were retrieved from each application and the resulting 3D model was saved. While the logging was implemented directly on our Mockup Builder application, we create a Ruby based plug-in using the Sketchup Ruby API to register a set of observers logging changes performed on the 3D model and the access to any feature of the Sketchup user interfaces such as activating an option using the toolbar. The scripts were used

System	**Mockup Builder**	**Sketchup 8**
Setup	3D stereoscopic tabletop	Desktop with 17″ screen
Visualization	Stereoscopic Perspective	Orthogonal or Perspective
Input Type	Multipoint or Bimanual	Single Cursor
Modeling	Push-and-Pull	Push-and-Pull
Shape Creation	Sketch based	Primitive Instantiation
2D UI	On Demand Contextual Menus	Fixed Menus
3D UI	3D Direct Manipulation	2D Direct Manipulation

Table 5.1: Property comparison between both system used during the user evaluation

to automate the testing process saving the 3D model and additional screenshots of the scene when closing the application after each task. A short user manual (five pages long) of each system was provided to the user explaining the basic features of each system and can be found in both Annexes A and B. Both user manuals were written in Portuguese and have been translated for the purpose of this document. A quick reference card[5] was also made available regarding the Sketchup system with an inventory of all icons and shortcuts of the application. Finally at the end of both tests, a questionnaire was provided to the participants in order to assess their satisfaction and the different aspects of each modeling interface. The questionnaire can be found in Annex D and was written in English. We were available during the questionnaire filling to explain any terms if requested by the participants. Since exactly the same questions were answered for both systems, the participant identified the system and its order on each questionnaire. The questionnaires start to profile the user experience regarding 3D modeling and important technologies to our approach such as 3D stereoscopy, 3D input devices mainly related to gaming and finally regarding multi-touch devices. The questionnaire part regarding each system tries to retrieve information about the experience on both systems. The questions survey different aspects of modeling tasks and the user interface for a subjective assessment of each system. Starting with general questions regarding the experiment, we asked the used to qualify the easiness of main modeling functionalities i.e. content creation, content editing and manipulation. We also asked specific questions to assess the limitations and benefits of each user interface. The questionnaire ends with a set of questions related to the perception of both shapes and modeling actions in both systems, as well as global questions about the system functionality. With these questionnaires, we asked indirectly the user to compared both systems using simple Likert scales regarding features and approaches that were similar or different between both approaches. The raw data collected from users can be found in the annexes of this document.

[5]Sketchup 8 Quick Reference Card:http://dl.google.com/sketchup/gsu8/docs/en/SketchUp8RefcardWin.pdf

5.2.1 User Profile

The 14 participants (10 males and 4 females) where mostly students (8 out of 14) from the Architecture and Gaming Course of Computer Science with ages ranging from 21 to 48 (M=26, IQR=6.25). Only one of the participant was left handed and performed the Mockup Builder test using the left hand as the dominant hand. In total 9 users had an Architectural background and 5 came from Computer Science. Regarding architectural background 4 of them were Architects on their daily activity, 1 Designer, 1 Professor and 3 undergraduate students. All these participants except one obtained their degree or were studying at the Architectural Faculty of Lisbon (*Faculdade de Arquitectura, Universidade Técnica de Lisboa*) or attended the Architectural course from *Instituto Superior Técnico*. Only two participants did not have any experience on 3D modeling but in Game Programming or Design tools such as Unity3D. Regarding the 5 participants from Computer Science, 4 of them were undergraduate students and one was a PhD candidate, all at *Instituto Superior Técnico*. Figure 5.3 presents the percentage of participants with experience on several modeling systems. Half of the participants were experienced with Sketchup. None of them had previously experienced Mockup Builder. 71.4% of the participants had previously experienced stereoscopic viewing mainly thanks to 3D movies (57.1 %). All participants were experienced with gaming: 92.9 % using a last generation gaming console such as Nintendo Wii, Sony Playstation 3 and Microsoft XBOX 360 and their input devices : Kinect (35.7%), Wiimote (57.1%) and Move (42.9%). Regarding multi-touch technology, 85.7% of the participants used it daily on mobile phones or tablets (57.1%) and 35.7 % had experience with larger multi-touch (≥ 15″) surfaces.

5.2.2 Task analysis

Regarding the task execution, we were able to retrieve the complete logging information from 10 out of 14 participants. We did not consider in this analysis the data of four participants since we could not use both the timing and command sequence information due to a logging problem in the Mockup Builder application at the beginning of our test. However, they are considered in the other topics of

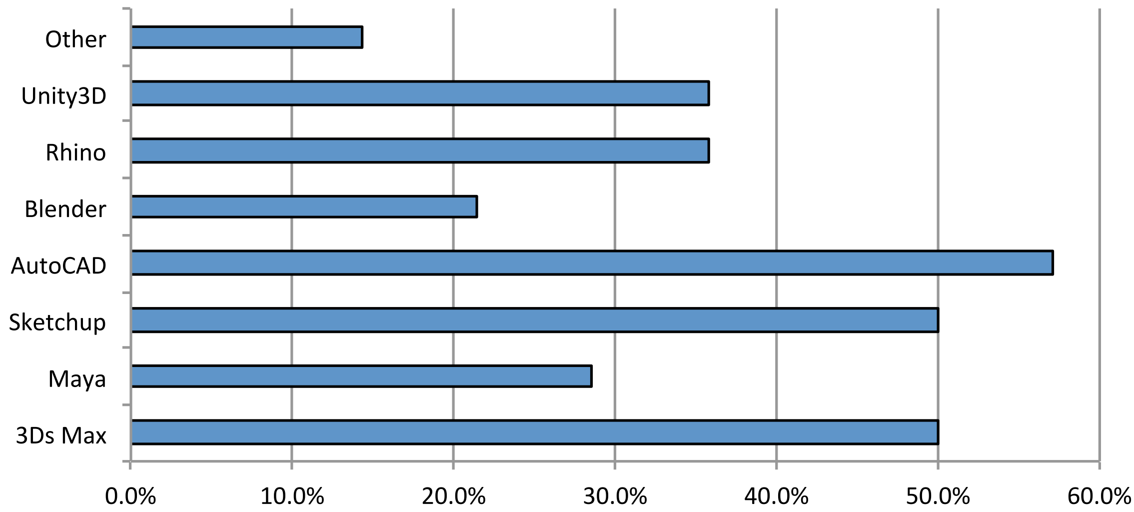


Figure 5.3: Percentage of participants with previous experience on each system.

the analysis since they correctly executed the test on both systems and answered the questionnaire. Table 5.2 shows the user profile for each participant whose timing information are reported in Figures 5.5,and 5.6. The timing information was retrieved automatically from the logging files while the user performed each task on each modeling system. The beginning was defined by the user launching the system after reading the task description. The finish time was registered automatically when the user closed the application as soon as he was satisfied with the modeled object. It provides a broad overview of the performance of each system. As mentioned before, the tasks were preceded by a practice period with no time limit. On average, participants spent 8.21 minutes on Sketchup and 15.19 minutes on Mockup Builder during this free test session as presented by Figure 5.4. For each system, we invited the user to explore all the commands described on each manual in order to be comfortable with the basic functionality of each application. During this practice session, we were available to demonstrate any functionality if they had not understood it during the initial briefing or following the manual. While most of the user had experienced both multi-touch and stereoscopic visualization as mentioned in Section 5.2.1, the large multi-touch area and the head tracked stereoscopic visualization proposed by Mockup Builder was new for all users except participant P5. It might explain the difference of time spent on both systems. During this training period, we provided a wooden

Id	Academic Course	Main Activities	Age	Modeling Experience	Sketchup Usage
P1	CS	Bsc Student	21	Novice	No
P2	ARCHI	Architect	24	Advanced	Yes
P3	CS	Bsc Student	23	Novice	No
P4	ARCHI	Msc Student	29	Advanced	No
P5	CS	Phd Student Researcher	26	Intermediate	Yes
P6	ARCHI	Phd Student Researcher	32	Advanced	Yes
P7	ARCHI	Phd Student Architect	41	Expert	No
P8	ARCHI	Designer	27	Expert	Yes
P9	ARCHI	Professor Architect	48	Expert	No
P10	ARCHI	Architect	27	Advanced	Yes

Table 5.2: Task Participant details: CS and ARCHI are participants with Computer Science or Architectural background respectively. Modeling experience was established based on questionnaire information and interview.

platform (20 cm height, 50 cm wide and 180 cm long) to better accommodate the user height regarding the table (100 cm height). Only one user preferred to use the platform during the tests. Apart from the height variation and the hand dominance, we did not considered other user variables during the test. No complains were made regarding the stereoscopic visualization during both the practice and task sessions. However, we believe that it is mostly due to its novelty. Only one user had experienced head tracked stereoscopy before the test. The interpupillary distance variation between subjects was not considered at the time of the user evaluation.

We performed a pair-wise comparison on modeling time between the system used (Sketchup, Mockup Builder) for each of the tasks. Only on the third task a significant effect or interaction was found. In this task, the medians of the group using Mockup Builder and the group using Sketchup were 10.35 minutes and 6.24 minutes, respectively. A Wilcoxon Signed-rank test shows that there is a significant effect of Group (W = 4, Z = -2.70, p ¡ 0.05, r = 0.60). No other significant effect or interaction was found showing that the participants perform similarly using both systems in the first and second task . However considering

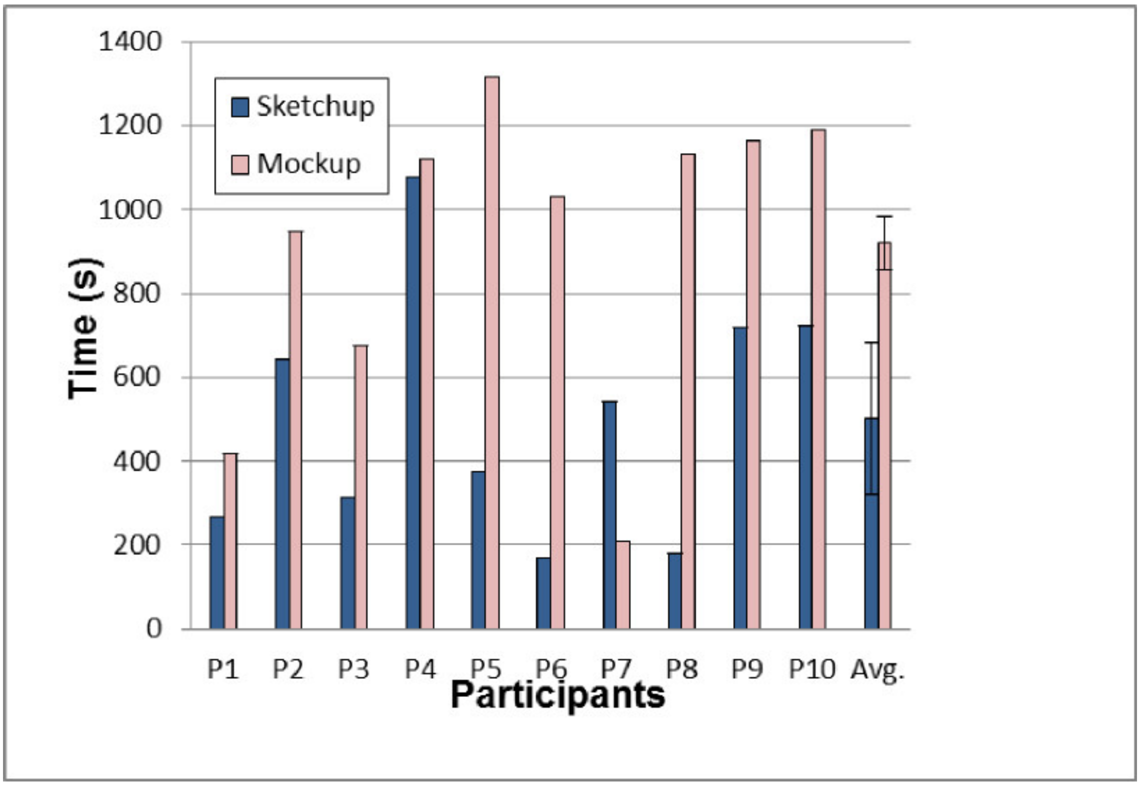


Figure 5.4: Time in second of the practice session performed by each participant before starting the three evaluation tasks, the last value presents the average time with error bars representing 95% CI for the mean.

the difference in expertise among the participants shown in Table 5.2, it is also interesting to look at the raw results. As presented by Figure 5.5, most of the participants were able to perform the first and second task more rapidly using Mockup Builder than using Sketchup. The lower time observed for the first task (average time of 9.49 minutes for Sketchup versus 7.27 minutes for Mockup Builder) can be explained by the easiness of creating a great variety of 2D shapes using our sketch based approach. In the second task (average time of 8.07 minutes for Sketchup versus 6.48 mintes for Mockup Builder), the difference can be mostly explained by difficulties to create freeform extrusions to represent the straw object using Sketchup. In addition, we observed it was harder to place the straw inside the glass using the Sketchup single view. The bimanual model and 3D direct manipulation from Mockup Builder shown to greatly ease the completion for these two tasks.

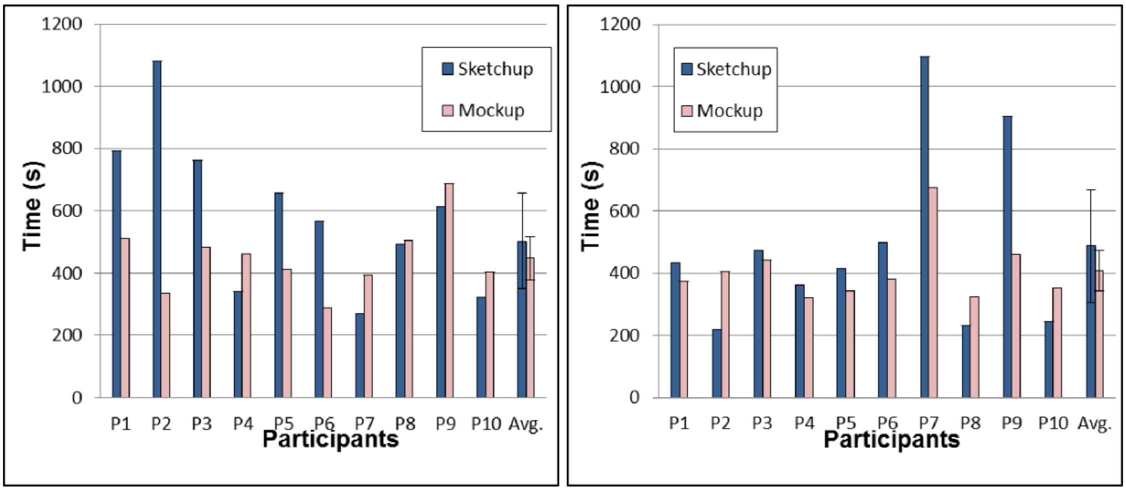


Figure 5.5: Time in second for the execution of Task 1 (left) and Task 2 (right) for each participant, the last value presents the average time with error bars representing 95% CI for the mean.

Regarding the third task, most of the participants performed better using Sketchup than Mockup Builder as shown by Figure 5.6. The main problem was related with the roof creation and the strategy followed by the participants to create the cactus like shape. The difference between systems is particularly visible for the novice user (P3). We noticed an initial difficulty from the participants to understand the snapping operation as a solution to sketch on a face located in space. During the experiment, the participants were first invited to consult the small user manual. It proved to be sufficient regarding this operation since we do not needed to explicitly intervene along the task execution. In addition most of the participants started by creating the roof using the scaling operation of the extrusion and they had to start a new house model since we do not provide the undo mechanism. This may explain the timing difference. However we believe that with this option, most of the users would be able to complete the task with a time similar to Sketchup. During the task execution, we noticed difficulties selecting menu options while maintaining the dominant hand on the correct feature. This aspect should be improved to reduce the number of necessary or incorrect access to the contextual menu. Figure 5.8 presents examples of the 3D models generated by five participants for each task using our Mockup Builder system and the Sketchup application.

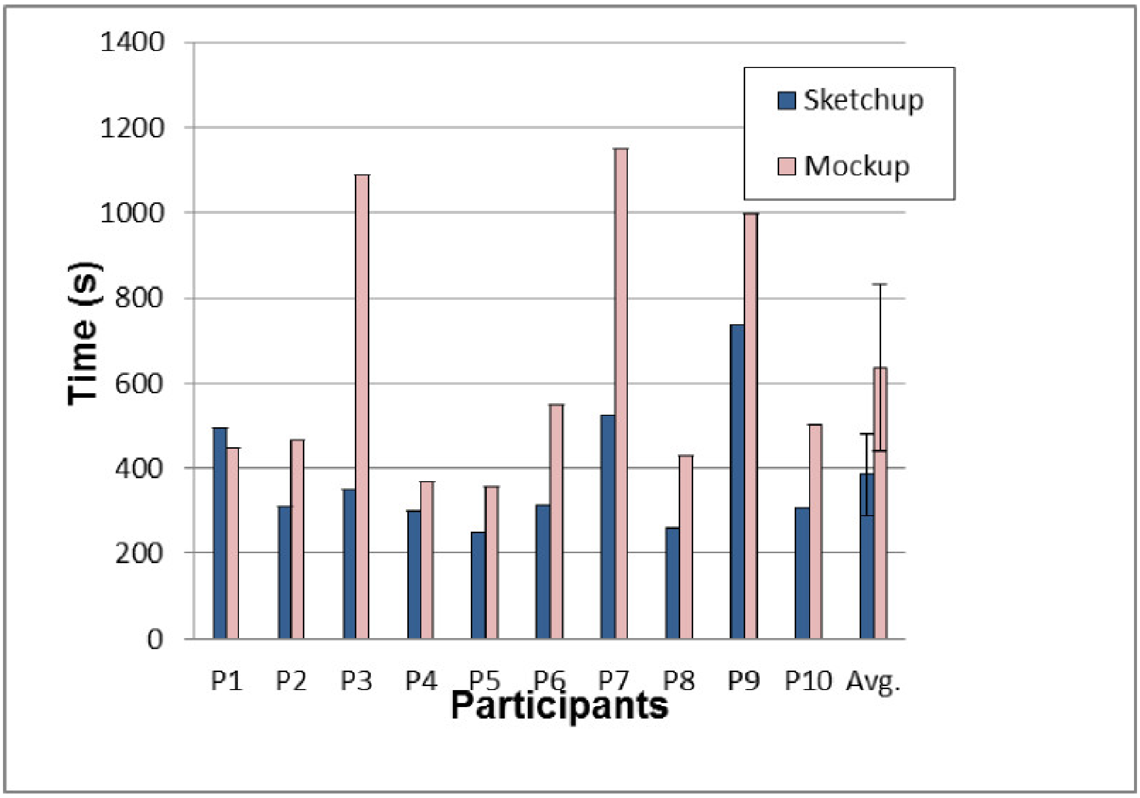


Figure 5.6: Time in second for the execution of Task 3 for each participant, the last value presents the average time with error bars representing 95% CI for the mean.

5.2.3 Questionnaire analysis

The questionnaire is based on a Likert scale from 1 to 5 for each questions. The median values and the corresponding interquartile range are presented in Figure 5.7 for each system. For all the questions the best score is the best system, since we invert the result presentation for negative affirmations such as "I had difficulties", "It was difficult" or "My hands or the cursor disturbed me" (i.e. Questions 4, 5, 30, 31, 32, 33 and 34). The questions were organized into groups relative to global aspects of the interface, 3D perception and easiness of the operations. Regarding the easiness of operations the main aspects requested were creation of shapes, extrusion, manipulations, selection and graphical user interfaces. The same questions were asked for both systems.

While both systems scored high values on the Likert scale (≥ 3), most of the answers do not show statistically significant difference between the two systems.

Participants found easier to create 2D shapes on Sketchup than Mockup Builder (Question 6) with a respective median value of 5.0 and 4.0 and a significant effect group ($W = 3.5$, $Z = -2.39$, $p < 0.05$, $r = 0.45$) as shown by a Wilcoxon Signed-rank test. However curve selection scored better in Mockup Builder since it offers a more flexible representation based on sketching and a better 3D perception. We also note that selection of features was easier in Sketchup using the mouse that in Mockup Builder using the 3D space as shown by Question 23 (median value 5.0 and 3.0 for Sketchup and Mockup Builder $W=0$, $Z=-3.173$, $p<0.005$, $r=0.59$) and Question 24 (median value 5.0 and 3.0 for Sketchup and Mockup Builder $W=0$, $Z=-3.34$, $p<0.005$, $r=0.63$). This drawback is mainly due to the hand occlusion problem in Mockup Builder as revealed by the questions 32 (5.0 vs 3.5 with $W=3$, $p<0.05$, $r=0.49$), 33 (5.0 vs 3.0 with $W=2$, $p<0.005$, $r=0.53$) and 34 (5.0 vs 3.0 with $W=4$, $p<0.005$, $r=0.54$) and to correctly identify which face was highlighted as shown by Question 30 (4.0 vs 3.5 with $W=3.5$, $p<0.05$, $r=0.44$). However both systems ranked similar scores regarding the usage of menus. On the other hand, the erasing solution of Mockup Builder which is subject to recognition does not seem to be as efficient as the undo operation as shows Question 28 (4.5 vs 1.5 with $W=0$, $p<0.05$). However we can notice a preference regarding manipulation and view control in Mockup Builder. It was easier for participants to place objects in space using our system and to perceive position and size relation between objects.

5.2.4 Additional user Comments

Through the questionnaires we asked participants for what they liked or disliked most on each system. We collected informal comments during the experiment and participants were free to make any additional suggestion to improve each system. Regarding Mockup Builder, 6 out of 14 participants (6/14) had positive global comments such as ”the system is fun to work”, ”it is easy to draw shapes” , ”I like the concept” or ”the system is interactive and easy to map ideas to commands” and the most attractive aspects mentioned by participants were the 3D perception (6/14), direct manipulation (5/14) and the drawing approach (4/14). On the other hand, the most relevant negative aspects were related to hand coordination problems (3/14), lack of precision (3/14), difficulty of selection (3/14),

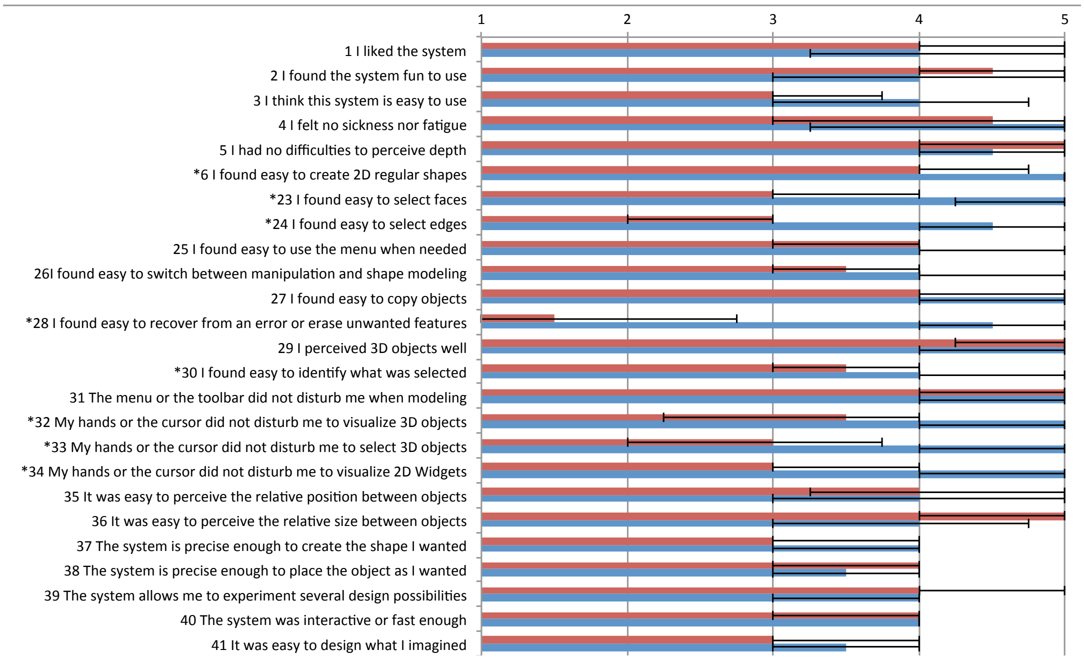


Figure 5.7: Details of the questionnaire results showing median values and quartiles for Sketchup (blue) and Mockup Builder (red). Each question was answered on a Likert scale where 1 represented "Strongly disagree" or similar and 5 represented "Strongly agree" or similar, depending on the question. A star(*) in front of a question number represents a significant effect found.

fatigue (3/14) and cyber-sickness (3/14). Additional suggestions to improve the system include the missing undo/redo functionality (6/14), the need for a more effective erasing method (4/14), an easier way to perform the snapping operation (3/14) and missing modeling features such as boolean operators or basic primitive instantiations (3/14). Finally, 3 out of 14 participants proposed to include a system to define objects' dimensions to overcome the inherent lack of precision with the drawing approach. Two participants suggested to improve the sketching technique such as including the ability to draw inside a shape or revise the split operation that was found to be too restrictive to add details on a face.

Regarding Sketchup, participants preferred to highlight global positive aspects (7/14) instead of specific characteristics of the system. For example they commented that "it is fast to test simple ideas", "it is easy to create shapes" or "more practical". Only 3 out of 14 participants pointed out specific aspects such as the extrusion system or the easiness to select shape features. The main drawbacks highlighted by the participants were related to difficulties in controlling the view (3/14) or manipulating shapes especially regarding rotations (3/14). Three participants mentioned the lack of flexibility regarding curve based modeling operations and two participants had negative comments about menus due to the constant need to access the toolbar or the complexity of the menu hierarchy. Most of the complains were related to the 3D perspective visualization and the traditional 2D cursor based interaction metaphor. 5 out of 14 participants suggested to improve the viewing system by increasing the feedback through rendering effects or animations, the usage of predefined views or even a spectator view to navigate into the 3D model to improve the perception of depth cues. 3 out of the 14 participants also proposed new features such as boolean operations or free form operations (3/14). Only one architect participant commented that he would not use a commercial product such as Sketchup for his daily activities since it was found too restrictive to create complex models.

5.2.5 Areas for Improvement

On both systems, participants were able to fulfill the requested tasks. While it is difficult to formally measure and compare the quality of the models produced, their informal comparison suggest a similar quality. Figure 5.8 presents the final models for each task for five of the participants on both systems. Each column presents the results for one participant on each system alternatively. Mockup Builder models were exported in VRML97 format and the screenshots were rendered using a 3D model visualizer. These are followed on each column by the corresponding Sketchup screenshot. From the five participants presented in this figure, we should note that the third column (participant 5) is from a user with no architectural background and the second and fourth columns represent participants (participants 4 and 9) with no prior experience with Sketchup. Compared to the initial screenshots, it is visible that the resulting models are very similar on each system showing the reliability of our prototype compared to a commercial product such as Sketchup, which is very encouraging.

For the different tasks with the two interfaces, the participants managed to replicate as faithfully as possible the different objects, except for the straw object in

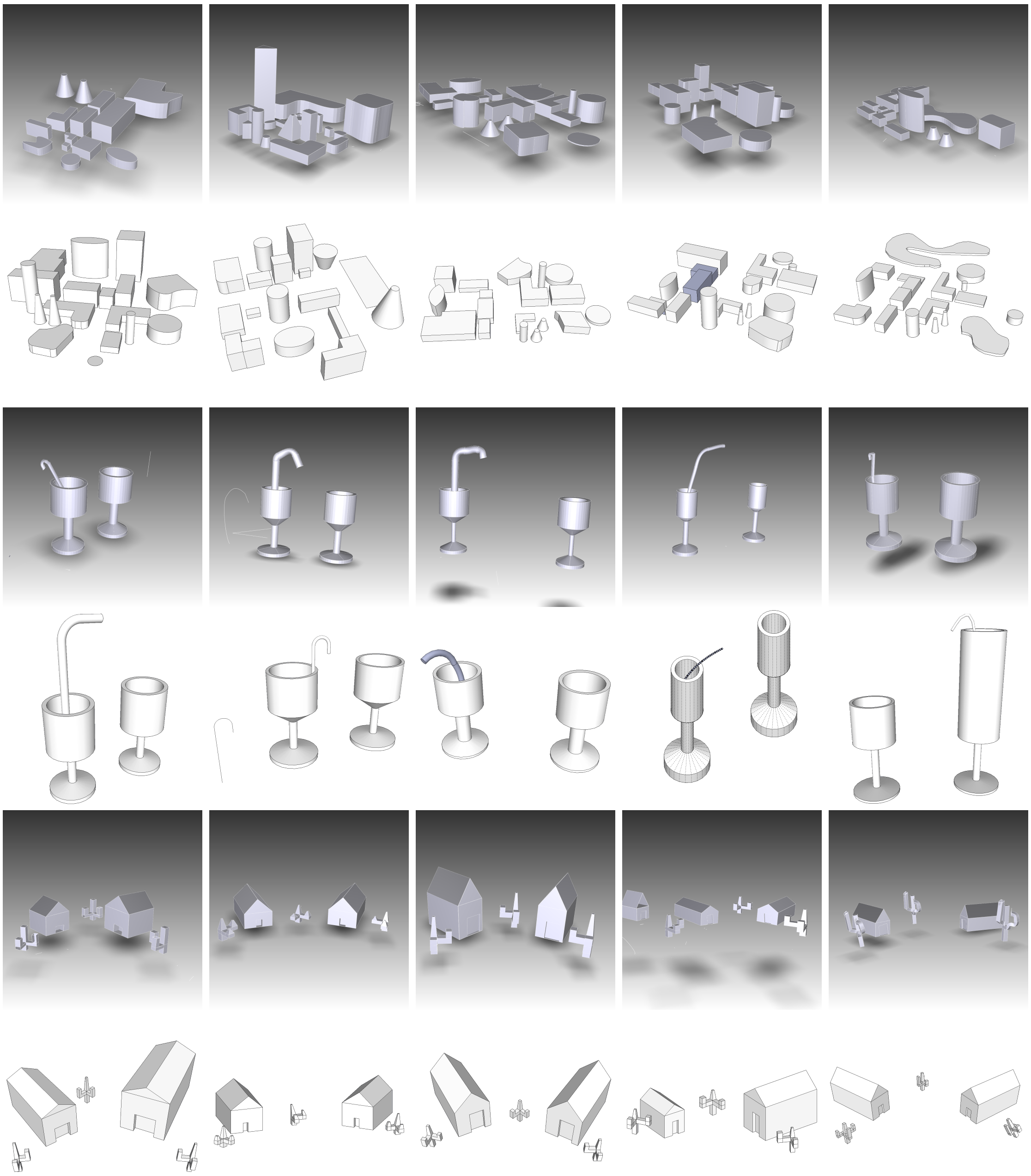


Figure 5.8: Resulting 3D models for 5 participants (P2,P4,P5,P9,P10) where each column represents a participant. The first, third and fifth rows present models obtained using Mockup Builder for Tasks 1,2 and 3 respectively. The second, the fourth and sixth rows present models obtained using Sketchup for Tasks 1,2 and 3 respectively.

the second task with Sketchup. Due to the limited ability to model curved shapes with Sketchup, participants could not create the straw as they actually wanted.

We also observed difficulties to correctly position the straw inside the glass with Stechup while this task was performed seamlessly with Mockup Builder. Finally for the third task, we observed that the objects modeled using Mockup Builder were not correctly aligned with the scene floor plane. This is due to the fact that Mockup Builder does not represent explicitly it. Future releases should represent the scene floor plane combined with a separate representation of the drawing grid. This would help maintaining a fixed reference when the user snaps on face.

3D perception For Mockup Builder, both the stereoscopic visualization and the interaction above the surface helped manipulating 3D objects and perceive 3D relationships between shapes. This was highlighted by 43% of the participants as the most attractive feature of the system. However, 21% of the participants raised possible problems related to fatigue, nausea and motion sickness if they would have to use it during a prolonged period of time or on a daily basis. Such problems could be minimized using faster and more precise head tracking solutions than the current Kinect device. Currently, both the low frame rate (30 frames per second) of the Kinect and its inherent latency (around 80ms) create a jellied effect on the 3D object visualization when performing fast head movements.

Accurate calibration between the Gametrak and the visualization is also important to keep fingers and virtual objects co-located. The current calibration achieves around 1 millimeter offset close to the surface. However, the offset is around 20 mm above the surface due to the distance between the real fingertip position and the position where the Gametrak's string is attached to the finger (Figure 4.2). Most of the participants were able to accommodate this offset thanks to the virtual cursor representation. However, we had to re-calibrate the system with a different ring position on the finger for participant P7 due to his difficulty for selecting shape features in space. New systems for tracking fingertips such as the Leap Motion might lessen such problems. In addition, visual effects such as shadows could be added to improve depth perception not only between existing virtual objects but also between the user's hand and virtual content, which could be done easily implemented using the 3D data captured by the Kinect. These improvements would minimize problems related to hand occlusion and the lack of haptic feedback in space.

3D User Interface On both systems, participants pointed out problems with menus. Some participants complained about Sketchup due to the need of using the toolbar each time they needed to invoke a new command or switch between manipulation and modeling modes. Some participants also complained about the contextual menu in Mockup Builder. We observed during the experiment that participants did not always took advantage of all the modeling possibilities offered through the contextual menu. In fact, some shape features have available options in the contextual menu while other features have no available option, which can confuse users. To cope with this problem, we could improve the visual feedback to encourage users to invoke the contextual menu when needed.

We need to deepen our analysis of bimanual asymmetric interaction considering some participants were sometimes confused during the experiment. Finally, a more effective undo / redo mechanism should be added to recover from errors and the erasing solution should be improved to allow erasing local features instead of all the lines or shapes as it is currently performed.

Sketch based modeling Mockup Builder provides a good support for sketching and freeform shapes while following a consistent approach for both curves and lines creation. On the other hand, Sketchup relies primarily on primitive based creation and it only provides few operators for freeform shapes as highlighted by the participants. However, the sketching recognition system in Mockup Builder should be improved to ease the creation of lines and arcs and the creation of simple primitives.

On both systems, participants characterize the modeling ability as too simple or not precise enough. By enriching sketching with construction lines, we would help to overcome the lack of rigor inherent from sketching based approaches. Measurement feedback such as the one proposed by Sketchup is also a key step to solve such issue and should be considered in the future. We could improve our boundary representation to support sketching inside a face, since our splitting operator alone is too limited to add details on a face. Such improvement will enable a better combination between sketches and existing shapes. By providing the ability to reuse sketches in our modeling operations, such as extruding along an existing curve, we could enrich our sketching based approach further with boolean operators. New operations such as revolution and support for non planar surface creation would enforce both the usage of 3D gestures and sketching for 3D modeling as a broader communication tool compared to existing traditional approaches.

5.3 Summary

We have described an approach to model 3D scenes using semi-immersive virtual environments through a synergistic combination of natural modalities afforded by novel input devices. While early experiments and informal assessments of our system show promise and seemingly validate some of these assumptions, we performed a formal user evaluation with both novice and expert users to highlight and explore both the strengths and the weakness of our modeling interface. This study allowed us to obtain user feedback about Mockup Builder. It revealed that the global usability of Mockup Builder was good. It also highlighted areas of improvements for a number of functionalities were participants encountered difficulties. The overall adhesion of the participants to Mockup Builder is comparable to Sketchup. This is a very positive result as Sketchup is a very popular, well established tool, whereas Mockup Builder is still in a prototype stage.

6

Conclusions and Future Work

The research conducted in this dissertation proposes an approach to model 3D scenes in a direct way using semi-immersive virtual environments through a synergistic combination of modalities afforded by novel input devices. Our system and experimental setup show that it is possible to enhance interaction by fusing data coming from different sensors. This provides a plausible environment combining benefits of multi-touch and stereo, using simple 3D operators, to model shapes using direct manipulation and simpler dialogues as compared to traditional and current systems. Combining the power of *bimanual interaction* with the flexibility of *continuous space*, we can provide effortless transition between modes and make it simple to switch between multi-touch 2D and spatial 3D gestures. This allows selecting the manipulations best suited to each task in non-obtrusive ways. In addition, the bimanual model makes frequent operations more accessible to the user allowing to easily switch between content creation and spatial manipulations avoiding the necessity of interruptive explicit modes to be selected by the user. This solution reduces the need of dialogs allowing the user to spend more time on content creation and direct manipulation of the 3D model. The bimanual model also enables to propose new 3D modeling operators complementing the action performed by the dominant hand with additional attributes or constraints.

Our Direct Modeling technique mixes a Sketch based Modeling interface with a Push and Pull modeling metaphor in a semi-immersive environment. Sketching allows to take advantages of user drawing skills in design phase and the expressiveness of such modality on the surface compared to primitive based instantiation modeling techniques. By combining both sketch recognition and beautification using constraints, users can create well defined regular and complex shapes easily even using imprecise gestures or sketches performed with its fingers. In addition, the Push and Pull correctly maps direct manipulations and fosters gestural based interaction taking advantage of interacting using both hands in 3D space. The combination of 3D gestures with 2D sketches allows to propose plausible gestures to define 3D shapes in space by co-locating the third dimension of shapes into the space above the surface. Our gestures extend basic 3D Direct Manipulation with modeling operations such as the linear and curvilinear extrusions allowing to interact with the shape representation without having to deal with the underlying mathematical definition of the shape. Users can change the topology adding details by sketching directly on the visual representation to create more complex shapes from simple ones. This is achieved thanks our extended boundary representation which updates the topological, geometrical and visual representations interactively increasing the expressiveness of gestures in 3D space and on the surface.

To validate our semi-immersive modeling environment approach and its interaction principles, we devise a within-subject user evaluation with users mostly experienced in 3D modeling. Our approach was compared to an existing CAD application following a similar Push and Pull paradigm however using a traditional WIMP based user interface. The results shows that our approach performs similarly to traditional desktop based modeling system providing a better depth perception between 3D shapes. In addition, it allows to better perceive curves and curved surfaces while defining them in 3D space which might be not trivial due to view changing in existing CAD systems.

6.1 Our Approach Benefits

To support modeling tasks taking advantage of VEs, we proposed a 3D modeling environment combining a multi-touch surface with a stereoscopic display and 3D input devices. Modeling operations relies on user interface mixing sketch based modeling principles and a gestural interface extending 3D direct manipulation concepts. Such approach showed several benefits presented in this section.

While semi-immersive environments are mainly used for visualization purpose, our goal was to use such technology to propose an alternative to traditional CAD systems to model 3D shapes and review 3D designs. By doing so, we expected to propose a new design environment with a better 3D perception than existing CAD systems. The combination of the surface with the space above it, enables to render virtual content on top of the surface such as the virtual objects were lying on the table. Such stereoscopic visualization scenario provides co-location between user hands and virtual objects adapted to direct modeling methods. Our environment makes virtual models more real, inviting the user into an environment with a better 3D perception and allowing him to interact more "physically" with virtual objects such as it is done naturally with physical mock ups. Compared to 2D user interfaces,our surface enables the user to correctly place objects between them while providing a better perception of the proportions between objects than traditional desktop orthogonal or perspective viewing. In addition the user does not have to rely only on the shading information or perspective clues to understand the shape since the parallax motion of the head tracked stereo instantly gives such feedback. While such conclusion was expected when users are visualizing their design, it also brings other advantages while the user is performing modeling tasks. First, it avoids the need to constantly adjust the 3D view before starting a modeling operation or relying on several orthogonal views as it is done by most of CAD systems. The stereoscopic visualization allows to fuse the viewing of the object with the viewing needed to correctly define a geometric operation. Such attribute has been preferred by the users compared to the unique 3D view solution proposed by the Sketchup application. For example, simple shape feature selections can be done directly on 3D space without having to accommodate the view to access a given feature. Secondly, it allows a better definition of curves and curved surface as shows our user evaluation. Finally, it makes 3D object placements within existing objects easier as remarked by the users during the second task of our user evaluation. From our evaluation participants, only one of them had a prior experience with head-tracked stereoscopic visualization. However, such novelty was not a major issue to avoid the completion of the tasks requested to the users.

While existing research on 3D modeling systems choose to adapt WIMP based GUIs to 3D space or to propose new tangible interfaces to support 3D modeling tasks in semi-immersive environments, our solution mainly relies on 3D Direct Manipulation, Sketching and Gestures. We combined two interaction techniques: the Guiard asymmetric model and the notion of continuous space to fosters plausible gestures mimicking physical interaction with objects. The user is able to interact directly with the virtual content as it was real taking advantage of our semi-immersive environment. 3D shapes can be spatially manipulated using the same direct based manipulation method on both the surface and the space above it. On the other hand, direct manipulation can be also done to edit shapes by selecting and moving its features. Using the bimanual asymmetric model, we can easily switch between spatial and geometric manipulations. Such approach is beneficial and the user is not forced to sequence manipulation related with modeling and model inspection as it done by conventional monoscopic displays or access to dialogues to switch between modes. It allows to correctly accommodate the model scale to add large or fine details in the 3D scene. We also showed that the bimanual model can be explored to propose new constrained based modeling operations since the NDH can be used to specify additional attributes to the action performed by the DH. Compared to traditional WIMP based interfaces which mainly rely on single cursor interaction, it enables to propose concurrent modeling operators such as our scaling widget while extruding, or limiting the height of an extrusion on the fly. Using the continuous space model, we can seamlessly abstract from the interaction on the surface and the space above it. Most operations follow the same paradigms on both space which make easy to introduce our gestures to the users during the evaluation. This seamless integration would not be possible without the fusion of all input data into a single user model providing continuous knowledge about user head, body and fingers.

We choose to propose a modeling approach combining seamlessly sketching and gestures in our continuous interaction space. While these modalities can be used separately in the more propitious reference space, i.e. sketching on a surface and gesturing in space, its combination favors and correctly maps 2.5 D approaches. For example, the extrusion based modeling paradigm invites the user to define planar representations on the surface and to extrude them to 3D space. Such as for the 3D manipulation on the surface and in the space above it, we mapped all inputs to 3D space. Our single user model abstracts inputs and does not differentiated sketches from gestures. This abstraction is beneficial since it allows to support functionalities identically in both spaces, leaving to the user the choice of which space is more adequate for its actions. In the implementation point of view, such strategy allows our system to follow the same approach to transform objects on the surface and above it. Since all surface manipulations are implemented in 3D space transforming 2D user touches into 3D, the lexicon of gestures used on the surface is still valid in space. By doing so novice users can rapidly be proficient in our modeling system reducing the learning curve.

As mentioned before,we use Push and Pull modeling method since it mainly relies on Direct Manipulations. This solution allows non experienced users to perform modeling tasks and it does not limit experienced user to create complex shapes. We show that a reduced but yet powerful set of modeling operations can be sufficient if they are correctly map through gestures. Participants only required a short training period to handle our system for the first time and they were able to fulfill the modeling tasks creating models similar to those created with professional modeling tools. Theses results are encouraging since our technique introduces several concepts which users are not accustomed with when it comes to computer user interfaces. The head tracked stereoscopic visualization, the gestural interface, the bimanual interaction and the sketching ability are very different features compared to traditional 2D user interfaces. However, it has not demonstrated to constraint the user, showing that these concepts are easy to maneuver and correctly map what the user is expecting from the system.

Regarding Sketch based Modeling interfaces, the research conducted in this thesis demonstrates that this concept which have been explored on 2D user interfaces can be also be applied in a semi-immersive environment. Sketching is a powerful language to define geometric content and can be used orthogonally in space. Our approach mainly relies on simple planar shapes mixing curves and lines. However, this is still expressive enough to represent most of manufactured objects combined with appropriate recognition and beautification techniques. The 3D visualization even simplifies the problem of reconstructing and interpreting drawings. We do not have to deal with projective distortions such as it is done in 2D interfaces where 3D content is usually presented using perspective or isometric views. It leverages the sketching skill requirement from the user, since it is easier for most people to draw in orthogonal views than understanding the principles of perspective drawing.

6.2 Limitations

The bimanual asymmetric interaction model provides an implicit switch between modeling and manipulation, letting the user to focus on its design. However it might be confusing for some users, in particular when interacting with a large multi-touch surface. That is why, we relaxed such model and we allowed users to scale objects using both hands if they so do wish. Still, users should heed the precedence of the non–dominant hand.

As in other modeling applications, menus could not be avoided altogether and are still required in particular when selecting from several modeling operations. In addition, gestural user interfaces are not as explicit as traditional GUIs even if they rely on plausible gestures. Without training or interactive tutorials, it is not clear for the user what are the modeling capabilities provided by the user interface. Another limitation is related with the solutions to define non ambiguous start and end gestures. Our strategy was to reuse the 2D mouse based *Drag and Drop* paradigm where interaction transitions are defined by button–like activations. However, users do not expect to define these interaction transitions explicitly when performing mid–air gestures as it is done when using an input device such as a mouse or a tangible tool.

Our approach tries to mimic plausible physical interaction based on finger tracking. However, direct manipulation methods still present some limitation due to the limited human accuracy to select items in the air. Even if the virtual model is co-located with user fingers precise selection might be difficult for some users. While we use threshold distances to identify which features are highlighted by the user, better mechanism should be explored to easy the selection of features in space. The quality and easiness of the selection in 3D space is intrinsically related to the finger tracking accuracy. However, the current calibration solution does not exactly match the tip of the finger. While it might be considered as a limitation solvable by improving the tracking accuracy, a better co-location between the virtual representation and fingers would not guarantee a better results. As demonstrated by our evaluation, the co-location of virtual content with user hands might difficult the selection in the air due to inherent physical occlusions and lack of haptic feedback. It is similar to the multi-touch fat finger problem. Another side-effect when dealing with mid-air gestures is the user arm fatigue also known as *Gorilla arm* problem. Even if modeling gestures are usually short gestures and our setup uses an horizontal screen as basis, further tests should be done to identify problems related to using our setup for long time periods and address fatigue issues.

Regarding sketching, our solution fits user finger inputs into a piecewise curve representation of Cubic Bézier curves or straight lines. Our fitting algorithm try to maximize the continuity between curve segments, which can difficult the definition of non continuous parts. Fitting algorithms should be improved or complemented to allow the user to define other types of curves and control the continuity between segments. In particular, we should favor the construction of arcs since it is a common representation in manufactured objects. Our constraint based approach only recognize constraint within strokes. This condition should be relaxed allowing constraints and beautification between strokes and existing 3D shape to propose a more reliable sketching interface. In addition, it should be possible to reuse stroke data for modeling operations instead of just relying on direct manipulation methods to propose more controlled modeling functionalities. Finally, methods to specify measures should be coupled with our sketch based modeling interface.

Our modeling environment relies on a head tracked stereoscopic visualization. While the current environment can only be used by one user, the 3D perception and illusion of content lying on the surface is only valid while the object projection remains on the limit of the surface. Multi–user stereoscopic environment should be addressed to allow several user to interact around the table for collaborative design review tasks. Regarding the 3D illusion limited by the surface, mechanisms should be explored to correctly render content out of the visualization volume without breaking the 3D illusion. During the user evaluation, we provide a platform to adjust the height of the user regarding the display. However, even if it was used by only one user, such possibility needs to be considered During the user evaluation, we used a fixed interpupillary distance of six centimeters. However such item should be adjustable manually or automatically to present a more adapted stereoscopic visualization to the user. In addition, further tests should be done to analyze the impact of working in stereo during long time period considering the user comments regarding nausea and fatigue. Alternatively, advances should be done regarding display technology to provide solutions to project volumetric content in the air and avoid motion sickness. This work shows that holographic like technology if achievable would be useful not only to visualize 3D objects but also to interact and work with them opening new opportunities for virtual prototyping.

Regarding 3D modeling, the Push and Pull approach was seen as too simple by some users for our system and the Sketchup application. Other modeling paradigm should be explored such as deformation based modeling and we should allow users to define boolean operations. Such as we did for the interaction, where the user can select the surface or the space for what is best suited for, 3D modeling should be also enriched providing redundant operators with different paradigms. It will easy the mapping of conceptual changes into modeling operations. In addition, it will attract more users taking more advantage of not only its sketching skills but also its modeling skills. A balance should be found between the functionality offered by our modeling environment and what is available in nowadays CAD systems. A better finger tracking would enable to explore more complex 3D modeling such as defining deformation by twisting and bending without increasing the complexity of our system.

To assess both benefits and limitations, we devise a user evaluation comparing our system to an existing application using mainly users experienced in 3D modeling. As mentioned in our evaluation, a total of 10 participants were considered in this experiment with an average duration of 1 hour and 31 minutes per session. While such approach allowed to retrieve comments regarding the technology and modeling approach proposed by our system compared to the Sketchup application and their prior CAD experience, it did not allow to present conclusive results regarding usage at long time period or on a daily basis. While the technology novelty does not seem to impact on the effectiveness of our method compared to an existing commercial system, repetitive tests distant in time should be done to better evaluate the learning curve requirements of our approach. The short briefing and simple manuals provided along the evaluation show promising results. However, further tests should be done to really quantify the easiness of usage of our approach involving novice users or users with a different modeling experience. By doing so, we could better assess if our user interface is more natural or efficient compared to existing CAD systems.

6.3 Future Work

The system shows clear promise and provides a good case for augmenting interactive surfaces with gesturing, gaze and body posture to support interactive modeling operations. The approach can be further extended by exploring combinations of different modalities and experimenting with mode-inferencing to further enhance the fluidity of our modeling techniques. The bimanual interaction model shows benefits, however it should be further researched to better fit users with different backgrounds. In addition, we should consider more participants with different background to be able to conclude if our proposal is more adequate to novice users than expert users in future evaluations. Combined with longer periodic trials, it will be possible to better assess easiness of usage and user learnability. The 3D direct manipulation metaphor should be also improved, since the drag and drop approach using an explicit button to start and end actions is confusing for some users. While the touch feedback delimits implicitly gestures and sketches on the surface, gestures in space are more difficult to delimit. This issue could be minor by complementing our environment with tangible tools such as a tracked 3D pointer or a 3D pen leveraging activation states for the user.

Our above the surface interaction techniques proposes a 3D Direct Manipulation method combined with a gestural user interface for 3D modeling. We plan to experiment the Akimbo Kinect based solution[1] or the Leap Motion[2] device to take more advantage of the expressiveness of finger gestures. Further research should be done regarding 3D user interfaces to validate the transcription of 2D Direct Manipulation metaphor to 3D space. 3D Modeling presents a good scenario where physically based interaction methods are not always the best choice. In addition, a consistent solution should be found to delimit gestures for both the surface and the space above it. Better non intrusive finger tracking technology should be researched to minor such issue in order to be as reliable as finger tracking on the surface using the multi-touch technology.

The usage of stereoscopic displays in a daily basis should be further researched since user fatigue and nausea are still frequent with existing technology. Rendering techniques of stereoscopic content should be as realistic as current gaming rendering technology. VR frameworks are still complex environment and high rendering quality at high frequent rates in stereoscopic environment is still out of the scope of most Virtual Reality systems. Non photo realistic rendering techniques should be also included in stereoscopic environment to convey more information about the modeled shape and its design to the user.We plan to include shadowing and approximation of global illumination in future release of our system to increase the 3D realism. In addition, it would benefit user perception of shape as well as improving user interaction with virtual content. Shadows casting from the user silhouette on virtual objects could benefit simple operations such as selection in 3D space. Haptic feedback could also improve the physical based interaction making interaction virtual objects even more real. While current haptic technology is still too cumbersome, some illusions such vibrations when touching an object might minor the lack of touch on virtual objects in space.

Regarding Sketch based interfaces, further research should be done to support other curve representations and combination of line and arcs which are typical to manufactured objects. Constraints usage should be extended to beautify user

[1]3Gear Systems: gestural interfaces `http://www.threegear.com/`
[2]See `https://leapmotion.com/` for more information about the Leap Motion sensor.

stroke using also the information from existing shapes. For example, it should be possible to draw a line parallel to an existing edge of a face. While drawing inside a face, existing features of face should be considered allowing to snap to vertexes or center the drawing inside the shape. It will enable to better explore drawings as construction lines in the modeling process. In addition, it will enable to create regular content within a face such as regular subdivisions or define specific proportion relationship. We plan to add a measure system to reduce the lack of rigor of our sketching based approach.

Regarding 3D modeling, the push and pull metaphor should be extended with boolean and deformation operations allowing to target more expert users. A balance between the functionality of CAD systems and the complexity of the interface should be found to enable experienced users to edit the underlying shape representation with more controls. Better understanding of finger gestures can be used for more modeling operations. By increasing the modeling ability of our system, it will be easier for the user to map its conceptual idea into modeling changes taking also advantage of its CAD experience. An undo and redo mechanism should be also supported to easy the recovery from errors.

We plan to extend our shape representation to support to represent shapes inside an existing face. Currently holes or extension of shape can only done by rescaling successive extruded shapes. Simpler methods should be used relying on sketching. Our shape representation allows to represent such topology and operators should be included to enable it. On the other hand, we plan to combine our representation with procedural descriptions to present a high level representation of shapes. This could be done since the main operations of our modeling approach are the split and the extrusion operators which can be easily express by shape grammars. To create such high level representation, we can analyze the sequence of operation performed by the user in order to detect pattern to be used by the shape representation. Our sketching approach should detect both existing proportions and regularities on the shape to create a higher level representation. This will enable new high level operators following inverse procedural techniques ideas. For example, combined with adequate gestures, a novice user could resize a building and automatically more floors will be added. It will take advantage of the interaction modeling history and the modeling logic of the user to represent concepts closer to the user domain.

6.4 Final Remarks

The ultimate goal of the research conducted by this thesis was to envision the future of modeling interfaces considering 3D technology advances in both input and output. While existing CAD systems have choose to mainly rely on graphical user interface idioms such as the WIMP metaphor, our attempt was to foster basic human hand skills such as drawing and sculpting. Currently, we are far from providing user interfaces to support modeling tasks as easy and flexible as using a pencil on a paper or cardboards to create scale models. This thesis started with the assumption that sketching, gestures and stereoscopic displays could benefit existing design tools and get closer to the way people learn to conceive, reason about and manipulate three dimensional shapes. Inspired by the way how users create and interact with scale models, we devised our modeling environment with a user interface relying on sketches and gestures. The thesis of this dissertation stated that 3D modeling tasks could be performed as effectively as current 2D graphical user interface in a semi-immersive environment by providing adequate interaction techniques and modeling operators while offering new design opportunities. We proposed a novel design environment with an innovative combination of interaction techniques fostering sketching and plausible gestures in 3D space. We consider that the research conducted partially validates this hypothesis. Even with the novelty of the technologies used by our environment and the short training period, it allowed users to be as effective as using an existing CAD system to model 3D shapes. While we think to contribute on interaction technique basis to achieve such goal, modeling operators should be better explored and enhanced to get closer to the user domain.

This thesis mainly focused on modeling interaction techniques. The combination of the bimanual asymmetric model and the continuous interaction space are key elements to use semi-immersive environments to support modeling tasks. By fusing different input sensors in a unique reference space, we enhance the knowledge of the system regarding the actions performed by the user and their location i.e. on the surface or in the space above it. While tracking, visualization and recognition technology will evolve, it is mainly these interaction concepts combined with enriched modeling techniques that will enable to transfer CAD modeling to VEs and become a reliable alternative to existing CAD systems. All novel input and output hardware devices, such as the Leap Motion sensor, Kinect camera and finger tracking such as the 3DGear Systems solution or the Zpace 3D display[3] claim to support the appearance of new 3D design solutions. They can be easily seen as the basic components of the next generation of drafting or light table with promising impact on design as emergent 3D printing technologies. However, it will be only possible mastering both 3D direct manipulation and 3D gestures adapted to high level modeling operators.

Our modeling environment provides possible clues to support modeling tasks in a semi-immersive environment. Thanks to the interaction techniques proposed by this thesis, initial steps are given to take more advantage of the interaction modeling history to support modeling systems with a better understanding of the user design intentions. 3D modeling is an interactive incremental task which should not be limited as a simple set of manipulations to achieve a given geometrical representation mapping user conceptual ideas. Our environment enables the user to express its modeling intentions using sketching and gestures. The sequence of actions could be reused by CAD systems to complement geometric representations with construction information allowing to better structure 3D models internally and automatically. The sequence of modeling operations performed by the user could be used by the system as a way to describe 3D shapes. Combined with inference mechanisms based on the analysis of the sequence of modeling actions performed by the user, the geometric representation could use procedural grammars to represent different abstraction levels from the model. It will allow to propose more high level and meaningful modeling operations than the one proposed by existing CAD systems.

[3]Zpace is a standalone solution combining a 3D display with a tracked 3D pen device: `http://zspace.com/about-zspace-holographic-computing/`

Bibliography

[Abdelmohsen and Do, 2007] Abdelmohsen, Sherif and Ellen Yi-Luen Do (2007). Tangicad: Tangible interface for manipulating architectural 3d models. In *CAADRIA 2007: Proceedings of the 12th International Conference on Computer Aided Architectural Design Research in Asia*.

[Aliaga, Rosen, and Bekins, 2007] Aliaga, Daniel G., Paul A. Rosen, and Daniel R. Bekins (2007). Style grammars for interactive visualization of architecture. *IEEE Transactions on Visualization and Computer Graphics* 13(4): 786–797.

[Amicis, Conti, and Fiorentino, 2004] Amicis, Raffaele De, Giuseppe Conti, and Michele Fiorentino (2004). Tangible interfaces in virtual environments for industrial design. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pp. 261–264, New York, NY, USA. ACM.

[Anderson et al., 2000] Anderson, David, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S. Yedidia (2000). Tangible interaction + graphical interpretation: a new approach to 3d modeling. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 393–402, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Anderson, Esser, and Interrante, 2003] Anderson, Lee, James Esser, and Victoria Interrante (2003). A virtual environment for conceptual design in architecture. In *EGVE '03: Proceedings of the workshop on Virtual environments 2003*, pp. 57–63, New York, NY, USA. ACM.

[Bae, Balakrishnan, and Singh, 2008] Bae, Seok-Hyung, Ravin Balakrishnan, and Karan Singh (2008). Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pp. 151–160, New York, NY, USA. ACM.

[Bae, Kijima, and Kim, 2003] Bae, Seok-Hyung, Ryugo Kijima, and Won-Sup Kim (2003). Digital styling for designers: 3d plane-symmetric freeform curve creation using sketch interface. In Kumar, Vipin, Marina L. Gavrilova, Chih Jeng Kenneth Tan, and Pierre L'Ecuyer, editors, *ICCSA (3)*, Vol. 2669 of *Lecture Notes in Computer Science*, pp. 701–710. Springer.

[Bae et al., 2004] Bae, Seok-Hyung, Takahiro Kobayash, Ryugo Kijima, and Won-Sup Kim (2004). Tangible nurbs-curve manipulation techniques using graspable handles on a large display. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pp. 81–90, New York, NY, USA. ACM.

[Balakrishnan et al., 1999] Balakrishnan, Ravin, George Fitzmaurice, Gordon Kurtenbach, and William Buxton (1999). Digital tape drawing. In *UIST '99: Proceedings of the 12th annual ACM symposium on User interface software and technology*, pp. 161–169, New York, NY, USA. ACM.

[Bein et al., 2009] Bein, M., S. Havemann, A. Stork, and D. Fellner (2009). Sketching subdivision surfaces. In *SBIM '09: Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, pp. 61–68, New York, NY, USA. ACM.

[Bekins and Aliaga, 2005] Bekins, D. and D.G. Aliaga (2005). Build-by-number: rearranging the real world to visualize novel architectural spaces. In *Visualization, 2005. VIS 05. IEEE*, pp. 143–150.

[Ben-Joseph et al., 2001] Ben-Joseph, Eran, Hiroshi Ishii, John Underkoffler, Ben Piper, and Luke Yeung (2001). Urban simulation and the luminous planning table: Bridging the gap between the digital and the tangible. *Journal of Planning Education and Research* 21: 196–203.

[Bergig et al., 2009] Bergig, Oriel, Nate Hagbi, Jihad El-Sana, and Mark Billinghurst (2009). In-place 3d sketching for authoring and augmenting mechanical systems. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pp. 87–94.

[Berndt, Fellner, and Havemann, 2005] Berndt, Rene, Dieter W. Fellner, and Sven Havemann (2005). Generative 3d models: a key to more information within less bandwidth at higher quality. In *Web3D '05: Proceedings of the tenth international conference on 3D Web technology*, pp. 111–121, New York, NY, USA. ACM.

[Birch et al., 2001a] Birch, P., V. Jennings, A. M. Day, and D. B. Arnold (2001a). Procedural modelling of vernacular housing for virtual heritage environments. In *Conference Proceedings of Eurographics UK Chapter 2001*, pp. 20–25, UCL, London.

[Birch et al., 2001b] Birch, P. J., S. P. Browne, V. J. Jennings, A. M. Day, and D. B. Arnold (2001b). Rapid procedural-modelling of architectural structures. In *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pp. 187–196, New York, NY, USA. ACM.

[Bokeloh et al., 2009] Bokeloh, M., A. Berner, M. Wand, H.-P. Seidel, and A. Schilling (2009). Symmetry detection using feature lines. *Computer Graphics Forum* 28(2): 697–706.

[Bokeloh, Wand, and Seidel, 2010] Bokeloh, Martin, Michael Wand, and Hans-Peter Seidel (2010). A connection between partial symmetry and inverse procedural modeling. In *SIGGRAPH '10: ACM SIGGRAPH 2010 papers*, pp. 1–10, New York, NY, USA. ACM.

[Bourdot et al., 2010] Bourdot, Patrick, Thomas Convard, Flavien Picon, Mehdi Ammi, M. Damien Touraine, and Jean-Marc Vézien (2010). Vr-cad integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of cad models. *Computer-Aided Design* 42(5): 445–461.

[Brandl et al., 2008] Brandl, Peter, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen (2008). Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pp. 154–161, NY, USA. ACM.

[Brito, Fonseca, and Jorge, 2005] Brito, Tiago, Manuel J. Fonseca, and Joaquim A Jorge (2005). Decosketch towards calligraphic approaches to interior design. In *eCAADe Conference, Digital Design: The Quest for New Paradigms*, pp. 665–670.

[Broll et al., 2005] Broll, Wolfgang, Irma Lindt, Jan Ohlenburg, Iris Herbst, Michael Wittkamper, and Thomas Novotny (2005). An infrastructure for realizing custom-tailored augmented reality user interfaces. *IEEE Transactions on Visualization and Computer Graphics* 11(6): 722–733.

[Broll et al., 2004] Broll, Wolfgang, Irma Lindt, Jan Ohlenburg, Michael Wittkamper, Chunrong Yuan, Thomas Novotny, Ava Fatah Schieck, Chiron Mottram, and Andreas Strothmann (2004). Arthur: A collaborative augmented environment for architectural design and urban planning. *Journal of Virtual Reality and Broadcasting* 1(1). `urn:nbn:de:0009-6-348`, ISSN 1860-2037.

[Broll, Stoerring, and Mottran, 2003] Broll, Wolfgang, Moritz Stoerring, and Chiron Mottran (2003). The augmented round table - a new interface to urban planning and architectural design. In Rauterberg, Matthias, Marino Menozzi, and Janet Wesson, editors, *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*. IOS Press.

[Browne et al., 2001] Browne, S. P., J. Willmott, L. I. Wright, A. M. Day, and D. B. Arnold (2001). Modelling & rendering large urban environments. In *Conference Proceedings of Eurographics UK Chapter 2001*, pp. 95–105, UCL, London.

[Bunnun and Mayol-Cuevas, 2008] Bunnun, Pished and Walterio W. Mayol-Cuevas (2008). Outlinar: an assisted interactive model building system with reduced computational effort. In *ISMAR '08: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 61–64, Washington, DC, USA. IEEE Computer Society.

[Casiez, Roussel, and Vogel, 2012] Casiez, Géry, Nicolas Roussel, and Daniel Vogel (2012). 1€ filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of CHI'12*. ACM.

[Chen, Wang, and Wang, 2009] Chen, Irene Rui, Xiangyu Wang, and Wei Wang (2009). Bridging shape grammar and tangible augmented reality into collaborative design learning. *International Conference on Computer Supported Cooperative Work in Design* 0: 468–473.

[Chen et al., 2008] Chen, Xuejin, Sing Bing Kang, Ying-Qing Xu, Julie Dorsey, and Heung-Yeung Shum (2008). Sketching reality: Realistic interpretation of architectural designs. *ACM Trans. Graph.* 27(2): 1–15.

[Cheok, Edmund, and Eng, 2002] Cheok, Adrian David, Neo Weng Chuen Edmund, and Ang Wee Eng (2002). Inexpensive non-sensor based augmented reality modeling of curves and surfaces in physical space. In *ISMAR '02: Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, p. 273, Washington, DC, USA. IEEE Computer Society.

[Choi et al., 2005] Choi, Han-wool, Hee-joon Kim, Jeong-in Lee, and Young-Ho Chai (2005). Free hand stroke based virtual sketching, deformation and sculpting of nurbs surface. In *ICAT '05: Proceedings of the 2005 international conference on Augmented tele-existence*, pp. 3–9, New York, NY, USA. ACM.

[Coquillart, 1987] Coquillart, S. (1987). Computing offsets of b-spline curves. *Comput. Aided Des.* 19: 305–309.

[Dachselt and Hübner, 2006] Dachselt, R. and A. Hübner (2006). A survey and taxonomy of 3d menu techniques.

[Day et al., 2003] Day, A.M., D.B. Arnold, S. Havemann, and D. Fellner (2003). Combining polygonal and subdivision surface approaches to modeling urban environments. In *Cyberworlds, 2003. Proceedings. 2003 International Conference on*, pp. 189–197.

[de Araújo et al., 2010] de Araújo, B., T. Guerreiro, M. Fonseca, J. Jorge, J. Pereira, M. Bordegoni, F. Ferrise, M. Covarrubias, and M. Antolini (2010). An haptic-based immersive environment for shape analysis and modelling. *Journal of Real-Time Image Processing: Special Issue on Improving Display and Rendering Technology* 5(2): 73–90.

[de Araújo et al., 2005] de Araújo, Bruno, T. Guerreiro, R. Jota, J. A. Jorge, and J. A. Madeiras Pereira (2005). LEMe Wall: Desenvolvendo um sistema de multi-projecção. In *13° Encontro Português de Computação Gráfica*, pp. 191–196, Vila Real, Portugal.

[de Araújo and Jorge, 2003] de Araújo, Bruno and Joaquim Jorge (2003). Blobmaker: Free-form modelling with variational implicit surfaces. In *Proceedings of 12 Portuguese Eurographics Chapter (12 EPCG)*, pp. 17–26.

[De la Rivière et al., 2010] De la Rivière, Jean-Baptiste, Nicolas Dittlo, Emmanuel Orvain, Cédric Kervégant, Mathieu Courtois, and Toni Da Luz (2010). ilight 3D touch: a multiview multitouch surface for 3D content visualization and viewpoint sharing. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10.

[Deering, 1996] Deering, Michael F. (1996). The holosketch vr sketching system. *Commun. ACM* 39(5): 54–61.

[Design Studio, 2009] Design Studio (2009). [Short Film]. DS DesignStudio (Director), DS DesignStudio (Producer), Dassault Systemes (Client).

[Dipietro, Sabatini, and Dario, 2008] Dipietro, L., A.M. Sabatini, and P. Dario (2008). A survey of glove-based systems and their applications. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 38(4): 461 –482.

[Do, 2002] Do, Ellen Yi-luen (2002). Drawing marks, acts, and reacts: Toward a computational sketching interface for architectural design. *Artif. Intell. Eng. Des. Anal. Manuf.* 16(3): 149–171.

[Dorsey et al., 2007] Dorsey, Julie, Songhua Xu, Gabe Smedresman, Holly Rushmeier, and Leonard McMillan (2007). The mental canvas: A tool for conceptual architectural design and analysis. *Computer Graphics and Applications, Pacific Conference on* 0: 201–210.

[Farin, 2002] Farin, Gerald (2002). *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition.

[Fatah gen. Schieck et al., 2005] Fatah gen. Schieck, Ava, Alan Penn, C Mottram, Andreas Strothmann, Jan Ohlenburg, Wolfgang Broll, Francis Aish, and Simon Attfield (2005). Interactive space generation through play: Exploring the role of simulation on the design table. *International Journal of Architectural Computing* 3(1): 3–26.

[Finkenzeller, 2008] Finkenzeller, Dieter (2008). Detailed building facades. *IEEE Computer Graphics and Applications* 28(3): 58–66.

[Finkenzeller and Bender, 2008] Finkenzeller, Dieter and Jan Bender (2008). Semantic representation of complex building structures. In *Computer Graphics and Visualization (CGV 2008) - IADIS Multi Conference on Computer Science and Information Systems*, Amsterdam (Netherlands).

[Finkenzeller, Bender, and Schmitt, 2005] Finkenzeller, Dieter, Jan Bender, and Alfred Schmitt (2005). Feature-based decomposition of façades. In Fischer, Xavier and Daniel Coutellier, editors, *Research in Interactive Design: Proceedings of Virtual Concept 2005*, Biarritz, France. Springer.

[Finkenzeller and Schmitt, 2006a] Finkenzeller, Dieter and Alfred Schmitt (2006a). Rapid modeling of complex building façades. In Fellner, Dieter and Charles Hansen, editors, *Short paper proceedings of Eurographics 2006*, pp. 95–98.

[Finkenzeller and Schmitt, 2006b] Finkenzeller, Dieter and Alfred Schmitt (2006b). Representation of complex façades using typed graphs. In Fischer, Xavier and Daniel Coutellier, editors, *Proceedings of Virtual Concept 2006*, Cancun, Mexico. Springer.

[Fiorentino et al., 2002] Fiorentino, M., R. De Amicis, A. Stork, and G. Monno (2002). Surface design in virtual reality as industrial application. In *Proceedings of DESIGN CONFERENCE - DESIGN 2002*, Dubrovnik, Croatia.

[Fiorentino et al., 2004] Fiorentino, M., G. Monno, , and A. E. Uva (2004). Smart tools for virtual reality based cad. In *ADM-AIAS International Conference*.

[Fiorentino et al., 2002] Fiorentino, Michele, Raffaele Amicis, Giuseppe Monno, and Andre Stork (2002). Spacedesign: A mixed reality workspace for aesthetic industrial design. In *ISMAR '02: Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, p. 86, Washington, DC, USA. IEEE Computer Society.

[Fiorentino, Uva, and Monno, 2008] Fiorentino, Michele, Antonio E. Uva, and Giuseppe Monno (2008). Vr interaction for cad basic tasks using rumble feedback input: Experimental study. *Product Engineering: Tools and Methods Based on Virtual Reality* pp. 337–352.

[Fischler and Bolles, 1981] Fischler, Martin A. and Robert C. Bolles (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24: 381–395.

[Fitzmaurice, Ishii, and Buxton, 1995] Fitzmaurice, George W., Hiroshi Ishii, and William A. S. Buxton (1995). Bricks: laying the foundations for graspable user interfaces. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 442–449, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Fjeld, 2000] Fjeld, Morten (2000). Evaluating navigation methods for an ar system. In *Designing Augmented Reality Environments*, pp. 157–158.

[Fjeld et al., 2002] Fjeld, Morten, Kristina Lauche, Martin Bichsel, Fred Voorhorst, Helmut Krueger, and Matthias Rauterberg (2002). Physical and virtual tools: Activity theory applied to the design of groupware. *Computer Supported Cooperative Work* 11(1-2): 153–180.

[Fjeld et al., 1998] Fjeld, Morten, Kristina Lauche, S. Dierssen, Martin Bichsel, and Matthias Rauterberg (1998). Build-it: A brick-based integral solution supporting multidisciplinary design tasks. In Sutcliffe, Alistair G., Jürgen Ziegler, and Peter Johnson, editors, *Designing Effective and Usable Multimedia Systems*, Vol. 133 of *IFIP Conference Proceedings*. Kluwer.

[Fjeld et al., 1999] Fjeld, Morten, Fred Voorhorst, Martin Bichsel, and Helmut Krueger (1999). Exploring brick-based camera control. In Bullinger, Hans-Jörg and Jürgen Ziegler, editors, *HCI (2)*, pp. 1060–1064. Lawrence Erlbaum.

[Flack et al., 2001] Flack, P. A., J. Willmott, S. P. Browne, D. B. Arnold, and A. M. Day (2001). Scene assembly for large scale urban reconstructions. In *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pp. 227–234, New York, NY, USA. ACM.

[Fleisch et al., 2004a] Fleisch, Timo, Gino Brunetti, Pedro Santos, and André Stork (2004a). Stroke-input methods for immersive styling environments. In *SMI '04: Proceedings of the Shape Modeling International 2004*, pp. 275–283, Washington, DC, USA. IEEE Computer Society.

[Fleisch et al., 2004b] Fleisch, Timo, Florian Rechel, Pedro Santos, and André Stork (2004b). Constraint stroke-based oversketching for 3d curves. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pp. 161–165.

[Foley et al., 1990] Foley, James D., Andries Dam, Steven K. Feiner, and John F. Hughes (1990). *Computer graphics: principles and practice (2nd ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Fonseca and Jorge, 2000] Fonseca, M.J. and J.A. Jorge (2000). Using fuzzy logic to recognize geometric shapes interactively. In *Proceedings of The Ninth IEEE International Conference on Fuzzy Systems 2000*, Vol. 1.

[Fröhlich et al., 2000] Fröhlich, Bernd, Henrik Tramberend, Andrew Beers, Maneesh Agrawala, and David Baraff (2000). Physically-based manipulation on the responsive workbench. In *Proceedings of the IEEE Virtual Reality 2000 Conference*, VR '00, pp. 5–, Washington, DC, USA. IEEE Computer Society.

[Gal et al., 2009] Gal, Ran, Olga Sorkine, Niloy Mitra, and Daniel Cohen-Or (2009). iWires: An analyze-and-edit approach to shape manipulation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 28(3).

[Ganster and Klein, 2007] Ganster, Björn and Reinhard Klein (2007). An integrated framework for procedural modeling. In Sbert, Mateu, editor, *Spring Conference on Computer Graphics 2007 (SCCG 2007)*, pp. 150–157. Comenius University, Bratislava.

[Gausemeier, Fruend, and Matysczok, 2002] Gausemeier, J., J. Fruend, and C. Matysczok (2002). Ar-planning tool: designing flexible manufacturing systems with augmented reality. In *EGVE '02: Proceedings of the workshop on Virtual environments 2002*, pp. 19–25, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Geiger and Rattay, 2008] Geiger, Christian and Oliver Rattay (2008). Tubemouse - a two-handed input device for flexible objects. *3D User Interfaces* 0: 27–34.

[Gerth et al., 2005] Gerth, Björn, René Berndt, Sven Havemann, and Dieter W. Fellner (2005). 3d modeling for non-expert users with the castle construction kit v0.5. In *VAST 2005: 6th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, pp. 49–58.

[Gingold, Igarashi, and Zorin, 2009] Gingold, Yotam, Takeo Igarashi, and Denis Zorin (2009). Structured annotations for 2d-to-3d modeling. *ACM Trans. Graph.* 28(5): 1–9.

[Greuter et al., 2003] Greuter, Stefan, Jeremy Parker, Nigel Stewart, and Geoff Leach (2003). Real-time procedural generation of 'pseudo infinite' cities. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pp. 87–ff, New York, NY, USA. ACM.

[Grossman et al., 2001] Grossman, Tovi, Ravin Balakrishnan, Gordon Kurtenbach, George Fitzmaurice, Azam Khan, and Bill Buxton (2001). Interaction techniques for 3d modeling on large displays. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pp. 17–23, New York, NY, USA. ACM.

[Grossman et al., 2002] Grossman, Tovi, Ravin Balakrishnan, Gordon Kurtenbach, George Fitzmaurice, Azam Khan, and Bill Buxton (2002). Creating principal 3d curves with digital tape drawing. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 121–128, New York, NY, USA. ACM.

[Grossman, Balakrishnan, and Singh, 2003] Grossman, Tovi, Ravin Balakrishnan, and Karan Singh (2003). An interface for creating and manipulating curves using a high degree-of-freedom curve input device. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 185–192, New York, NY, USA. ACM.

[Guiard, 1987] Guiard, Yves (1987). Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior* 19: 486–517.

[Hachet et al., 2011] Hachet, Martin, Benoit Bossavit, Aurélie Cohé, and Jean-Baptiste Rivière (2011). Toucheo: multitouch and stereo combined in a seamless workspace. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 587–592, NY, USA. ACM.

[Hagbi et al., 2010] Hagbi, N., R. Grasset, O. Bergig, M. Billinghurst, and J. El-Sana (2010). In-place sketching for content authoring in augmented reality games. pp. 91 –94.

[Hahn, Bose, and Whitehead, 2006] Hahn, Evan, Prosenjit Bose, and Anthony Whitehead (2006). Persistent realtime building interior generation. In *Sandbox '06: Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, pp. 179–186, New York, NY, USA. ACM.

[Havemann and Fellner, 2004] Havemann, S. and D. Fellner (2004). Generative parametric design of gothic window tracery. In *Shape Modeling Applications, 2004. Proceedings*, pp. 350–353.

[Havemann and Fellner, 2008] Havemann, S. and Dieter W. Fellner (2008). Progressive combined b-reps - multi-resolution meshes for interactive real-time shape design. *Journal of WSCG* 16(1-3): 121–133.

[Havemann and Fellner, 2001] Havemann, Sven and Dieter Fellner (2001). A versatile 3d model representation for cultural reconstruction. In *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pp. 205–212, New York, NY, USA. ACM.

[Havemann and Fellner, 2003] Havemann, Sven and Dieter W. Fellner (2003). Generative mesh modeling. Technical report, Institute of ComputerGraphics, TU Braunschweig.

[Hinckley et al., 2010] Hinckley, Ken, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton (2010). Pen + touch = new tools. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pp. 27–36, NY, USA. ACM.

[Hohmann et al., 2009] Hohmann, Bernahard, Ulrich Krispel, Sven Havemann, and Dieter Fellner (2009). Cityfit - high-quality urban reconstructions by fitting shape grammars to images and derived textured point clouds. In *Proceedings of the 3rd ISPRS International Workshop 3D-ARCH 2009*, number 01 in TUBSCG-2003.

[Hohmann et al., 2010] Hohmann, Bernhard, Sven Havemann, Ulrich Krispel, and Dieter Fellner (2010). A gml shape grammar for semantically enriched 3d building models. *Computers & Graphics* 34(4): 322–334. Procedural Methods in Computer Graphics; Illustrative Visualization.

[Hosokawa et al., 2008] Hosokawa, Takuma, Yasuhiko Takeda, Norio Shioiri, Mitsunori Hirano, and Kazuhiko Tanaka (2008). Tangible design support system using rfid technology. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pp. 75–78, New York, NY, USA. ACM.

[Huang, Mann, and Cowan, 2009] Huang, Jingyuan, Stephen Mann, and Bill Cowan (2009). Inca reconstruction using shape grammar. In *SIGGRAPH '09: SIGGRAPH '09: Posters*, pp. 1–1, New York, NY, USA. ACM.

[Igarashi and Hughes, 2001] Igarashi, Takeo and John F. Hughes (2001). A suggestive interface for 3d drawing. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 173–181, New York, NY, USA. ACM.

[Igarashi, Matsuoka, and Tanaka, 1999] Igarashi, Takeo, Satoshi Matsuoka, and Hidehiko Tanaka (1999). Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 409–416, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Iron Man, 2008] Iron Man (2008). [Motion picture]. Jon Favreau (Director), Avi Arad and Kevin Feige (Producers), Marvel Studios (Studio), Paramount Pictures (Distributor).

[Iron Man 2, 2010] Iron Man 2 (2010). [Motion picture]. Jon Favreau (Director), Kevin Feige (Producer), Marvel Studios (Studio), Paramount Pictures (Distributor).

[Iron Man 3, 2013] Iron Man 3 (2013). [Motion picture]. Shane Black (Director), Kevin Feige (Producer), Marvel Studios (Studio), Walt Disney Studis Motion Pictures (Distributor).

[Ishii et al., 2004] Ishii, H., C. Ratti, B. Piper, Y. Wang, A. Biderman, and E. Ben-Joseph (2004). Bringing clay and sand into digital design — continuous tangible user interfaces. *BT Technology Journal* 22(4): 287–299.

[Ishii et al., 2002] Ishii, Hiroshi, Eran Ben-Joseph, John Underkoffler, Luke Yeung, Dan Chak, Zahra Kanji, and Ben Piper (2002). Augmented urban planning workbench: Overlaying drawings, physical models and digital simulation. In *ISMAR '02: Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, p. 203, Washington, DC, USA. IEEE Computer Society.

[Jin, Kim, and Park, 2007] Jin, Yoon-suk, Yang-wook Kim, and Jun Park (2007). Armo: Augmented reality based reconfigurable mock-up. In *ISMAR '07: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1–2, Washington, DC, USA. IEEE Computer Society.

[Jorge et al., 2004] Jorge, Joaquim A, Tiago Cardoso, Nelson Silva, and Jo Paulo Pereira (2004). Gides++: A rapid prototyping tool for mould design. In *Proceedings of the Rapid Product Development Event RDP 2004*, Marinha Grande, Portugal. Ed. Cefamol, Portuguese Association for the Mould Industry, support in CD-ROM.

[Jota et al., 2010] Jota, R., B. Araújo, L. Bruno, J. A. Madeiras Pereira, and J. A. Jorge (2010). Immiview - a multi-user solution for design review in real-time. *Journal of Real-Time Image Processing: Special Issue on Improving Display and Rendering Technology* 5(2): 91–107.

[Jota and Benko, 2011] Jota, Ricardo and Hrvoje Benko (2011). Constructing virtual 3D models with physical building blocks. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pp. 2173–2178, NY, USA. ACM.

[Kallio, 2005] Kallio, Kiia (2005). 3d6b editor: Projective 3d sketching with line-based rendering. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pp. 73–80.

[Kallmann and Thalmann, 1999] Kallmann, Marcelo and Daniel Thalmann (1999). Direct 3d interaction with smart objects. In *VRST '99: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 124–130, New York, NY, USA. ACM.

[Kaltenbrunner et al., 2005] Kaltenbrunner, M., T. Bovermann, R. Bencina, and E. Costanza (2005). Tuio a protocol for table-top tangible user interfaces.

[Kara, D'Eramo, and Shimada, 2006] Kara, Levent Burak, Chris M. D'Eramo, and Kenji Shimada (2006). Pen-based styling design of 3d geometry using concept sketches and template models. In *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pp. 149–160, New York, NY, USA. ACM.

[Kara and Shimada, 2006] Kara, Levent Burak and Kenji Shimada (2006). Construction and modification of 3d geometry using a sketch-based interface. In *EuroGraphics Workshop on Sketch-Based Interfaces and Modeling (SBIM'06)*.

[Karpenko and Hughes, 2006] Karpenko, Olga A. and John F. Hughes (2006). Smoothsketch: 3d free-form shapes from complex sketches. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pp. 589–598, New York, NY, USA. ACM.

[Karpenko, Hughes, and Raskar, 2002] Karpenko, Olga A., John F. Hughes, and Ramesh Raskar (2002). Free-form sketching with variational implicit surfaces. *Comput. Graph. Forum* 21(3).

[Kasik, 2011] Kasik, David J. (2011). The third wave in computer graphics and interactive techniques. *IEEE Comput. Graph. Appl.* 31(4): 89–93.

[Kato and Billinghurst, 1999] Kato, Hirokazu and Mark Billinghurst (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, p. 85, Washington, DC, USA. IEEE Computer Society.

[Kato et al., 2003] Kato, Hirokazu, Keihachiro Tachibana, Masaaki Tanabe, Takeaki Nakajima, and Yumiko Fukuda (2003). A city-planning system based on augmented reality with a tangible interface. In *ISMAR '03: Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, p. 340, Washington, DC, USA. IEEE Computer Society.

[Kaufmann and Schmalstieg, 2006] Kaufmann, Hannes and Dieter Schmalstieg (2006). Designing immersive virtual reality for geometry education. In *VR '06: Proceedings of the IEEE conference on Virtual Reality*, pp. 51–58, Washington, DC, USA. IEEE Computer Society.

[Keefe, Zeleznik, and Laidlaw, 2007] Keefe, Daniel, Robert Zeleznik, and David Laidlaw (2007). Drawing on air: Input techniques for controlled 3d line illustration. *IEEE Transactions on Visualization and Computer Graphics* 13: 1067–1081.

[Kim et al., 2005] Kim, Hyosun, Georgia Albuquerque, Sven Havemann, and Dieter W. Fellner (2005). Tangible 3d: Hand gesture interaction for immersive 3d modeling. In Kjems, Erik and Roland Blach, editors, *Eurographics Symposium on Virtual Environments*, pp. 191–199, Aalborg, Denmark. Eurographics Association.

[Kim and Fellner, 2004] Kim, Hyosun and Dieter W. Fellner (2004). Interaction with hand gesture for a back-projection wall. In *CGI '04: Proceedings of the Computer Graphics International*, pp. 395–402, Washington, DC, USA. IEEE Computer Society.

[Kitamura, Itoh, and Kishino, 2001] Kitamura, Yoshifumi, Yuichi Itoh, and Fumio Kishino (2001). Real-time 3d interaction with activecube. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pp. 355–356, New York, NY, USA. ACM.

[Kiyokawa, Takemura, and Yokoya, 2000] Kiyokawa, Kiyoshi, Haruo Takemura, and Naokazu Yokoya (2000). Seamlessdesign for 3d object creation. *IEEE MultiMedia* 7(1): 22–33.

[Krause et al., 2004] Krause, Frank-Lothar, Martin Göbel, Alain Roger Biahmou Tchebetchou, and Gerold Wesche (2004). A three-stage conceptual design process using virtual environments. In *WSCG (Posters)*, pp. 85–88.

[Kulik, 2009] Kulik, Alexander (2009). Building on realism and magic for designing 3d interaction techniques. *IEEE Computer Graphics and Applications* 29: 22–33.

[Lapides et al., 2006] Lapides, Paul, Ehud Sharlin, Mario Costa Sousa, and Lisa Streit (2006). The 3d tractus: A three-dimensional drawing board. *Horizontal Interactive Human-Computer Systems, International Workshop on* 0: 169–176.

[Larive, Dupuy, and Gaildrat, 2005] Larive, Mathieu, Yann Dupuy, and Véronique Gaildrat (2005). Automatic generation of urban zones. In *WSCG (Short Papers)*, pp. 9–12.

[Larive and Gaildrat, 2006] Larive, Mathieu and Veronique Gaildrat (2006). Wall grammar for building generation. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pp. 429–437, New York, NY, USA. ACM.

[Laycock and Day, 2003] Laycock, R. G. and A. M. Day (2003). Automatically generating large urban environments based on the footprint data of buildings. In *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, pp. 346–351, New York, NY, USA. ACM.

[Lee and Ishii, 2010] Lee, Jinha and Hiroshi Ishii (2010). Beyond: collapsible tools and gestures for computational design. In *CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, pp. 3931–3936, New York, NY, USA. ACM.

[Lee, Feng, and Gooch, 2008] Lee, Sangwon, David Feng, and Bruce Gooch (2008). Automatic construction of 3d models from architectural line drawings. In *I3D '08: Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pp. 123–130, New York, NY, USA. ACM.

[Lee et al., 2008] Lee, Sangwon, David Feng, Cindy Grimm, and Bruce Gooch (2008). A sketch-based user interface for reconstructing architectural drawings. *Computer Graphics Forum* 27(1): 81–90.

[Lindenmayer, 1968] Lindenmayer, Aristid (1968). Mathematical models for cellular interaction in development: Parts i and ii. *Journal of Theoretical Biology* 18.

[Lipp, Wonka, and Wimmer, 2008] Lipp, Markus, Peter Wonka, and Michael Wimmer (2008). Interactive visual editing of grammars for procedural architecture. *ACM Transactions on Graphics* 27(3): 102:1–10. Article No. 102.

[Liu et al., 2006] Liu, Yang, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang (2006). Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph.* 25(3): 681–689.

[Llamas et al., 2003] Llamas, Ignacio, Byungmoon Kim, Joshua Gargus, Jarek Rossignac, and Chris D. Shaw (2003). Twister: a space-warp operator for the two-handed editing of 3d shapes. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pp. 663–668, New York, NY, USA. ACM.

[Llamas et al., 2005] Llamas, Ignacio, Alexander Powell, Jarek Rossignac, and Chris D. Shaw (2005). Bender: a virtual ribbon for deforming 3d shapes in biomedical and styling applications. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pp. 89–99, New York, NY, USA. ACM.

[Lopes et al., 2011] Lopes, Pedro, Daniel Mendes, Bruno Araújo, and Joaquim A. Jorge (2011). Combining bimanual manipulation and pen-based input for 3D modelling. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pp. 15–22, NY, USA. ACM.

[MacKenzie, 2003] MacKenzie, Scott (2003). Motor behaviour models for human-computer interaction. In Carroll, John, editor, *HCI models, theories, and frameworks: Toward a multidisciplinary science*, pp. 27–54. Morgan Kaufmann, San Francisco, USA.

[Marquardt et al., 2011] Marquardt, Nicolai, Ricardo Jota, Saul Greenberg, and Joaquim A. Jorge (2011). The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III*, INTERACT'11, pp. 461–476, Berlin, Heidelberg. Springer-Verlag.

[Martin, 2006] Martin, Jess (2006). Procedural house generation: A method for dynamically generating floor plans. In *Symposium on Interactive 3D Graphics and Games*.

[Marvie, Perret, and Bouatouch, 2005] Marvie, Jean-Eudes, Julien Perret, and Kadi Bouatouch (2005). The fl-system: a functional l-system for procedural geometric modeling. *The Visual Computer* 21(5): 329–339.

[Masry, Kang, and Lipson, 2005] Masry, M., D. Kang, and H. Lipson (2005). A freehand sketching interface for progressive construction of 3d objects. *Comput. Graph.* 29(4): 563–575.

[Merrell, Schkufza, and Koltun, 2010] Merrell, Paul, Eric Schkufza, and Vladlen Koltun (2010). Computer-generated residential building layouts. *ACM Transactions on Graphics* 29(5): (to appear). Proceedings of SIGGRAPH ASIA 2010.

[Milgram et al., 1995] Milgram, Paul, Haruo Takemura, Akira Utsumi, and Fumio Kishino (1995). Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum 2351: 282–292.

[Minority Report, 2002] Minority Report (2002). [Motion picture]. Steven Spielberg (Director), Gerald R. Molen, Bonnie Curtis, Walter F. Parkes and Jan de Bont (Producers), 20th Century Fox and DreamWorks Pictures (Distributors).

[Mistry and Maes, 2009] Mistry, Pranav and Pattie Maes (2009). Sixthsense: a wearable gestural interface. In *ACM SIGGRAPH ASIA 2009 Sketches*, SIGGRAPH ASIA '09, pp. 11:1–11:1, New York, NY, USA. ACM.

[Mitani, Suzuki, and Kimura, 2002] Mitani, Jun, Hiromasa Suzuki, and Fumihiko Kimura (2002). 3d sketch: sketch-based model reconstruction and rendering. *From geometric modeling to shape modeling* pp. 85–98.

[Mitra, Guibas, and Pauly, 2006] Mitra, Niloy J., Leonidas J. Guibas, and Mark Pauly (2006). Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph.* 25(3): 560–568.

[Moustakas et al., 2006] Moustakas, Konstantinos, Dimitrios Tzovaras, Michael G. Strintzis, Sebastien Carbini, Olivier Bernier, Jean Emmanuel Viallet, Stephan Raidt, Matei Mancas, Mariella Dimiccoli, Enver Yagci, Serdar Balci, and Eloisa Ibanez Leon (2006). Masterpiece: Physical interaction and 3d content-based search in vr applications. *IEEE MultiMedia* 13(3): 92–100.

[Müller et al., 2005] Müller, P., T. Vereenooghe, A. Ulmer, and L. Van Gool (2005). Automatic reconstruction of roman housing architecture. In et al., Baltsavias, editor, *Recording, Modeling and Visualization of Cultural Heritage*, pp. 287–298. Balkema Publishers (Taylor & Francis group).

[Müller et al., 2006a] Müller, Pascal, Tijl Vereenooghe, Peter Wonka, Iken Paap, and Luc Van Gool (2006a). Procedural 3d reconstruction of puuc buildings in xkipché. In *VAST 2006: 7th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, pp. 139–146.

[Müller et al., 2006b] Müller, Pascal, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool (2006b). Procedural modeling of buildings. *ACM Transactions on Graphics* 25(3): 614–623. Proceedings of ACM SIGGRAPH 2006.

[Müller et al., 2007] Müller, Pascal, Gang Zeng, Peter Wonka, and Luc Van Gool (2007). Image-based procedural modeling of facades. *ACM Transactions on Graphics* 26(3). Proceedings of ACM SIGGRAPH 2007.

[Müller-Tomfelde et al., 2010] Müller-Tomfelde, Christian, Otmar Hilliges, Andreas Butz, Shahram Izadi, and Andy Wilson (2010). Interaction on the tabletop: Bringing the physical to the digital. In Müller-Tomfelde, Christian, editor, *Tabletops - Horizontal Interactive Displays*, Human-Computer Interaction Series, pp. 189–221. Springer London.

[Nishino, Utsumiya, and Korida, 1998] Nishino, Hiroaki, Kouichi Utsumiya, and Kazuyoshi Korida (1998). 3d object modeling using spatial and pictographic gestures. In *VRST '98: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 51–58, New York, NY, USA. ACM.

[Nishino et al., 1997] Nishino, Hiroaki, Kouichi Utsumiya, Daisuke Kuraoka, Kenji Yoshioka, and Kazuyoshi Korida (1997). Interactive two-handed gesture interface in 3d virtual environments. In *VRST '97: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 1–8, New York, NY, USA. ACM.

[Oh, Stuerzlinger, and Danahy, 2006a] Oh, Ji-Young, Wolfgang Stuerzlinger, and John Danahy (2006a). Sesame: towards better 3d conceptual design systems. In *DIS '06: Proceedings of the 6th conference on Designing Interactive systems*, pp. 80–89, New York, NY, USA. ACM.

[Oh, Stuerzlinger, and Danahy, 2006b] Oh, Ji-Young, Wolfgang Stuerzlinger, and John Danahy (2006b). Sesame: towards better 3d conceptual design systems. In *DIS '06: Proceedings of the 6th conference on Designing Interactive systems*, pp. 80–89, New York, NY, USA. ACM.

[Olsen et al., 2008] Olsen, L., F.F. Samavati, M.C. Sousa, and J. Jorge (2008). A taxonomy of modeling techniques using sketch-based interfaces. In *Eurographics 2008 State-of-the-Art Report (EG'08 STAR)*.

[Olsen et al., 2009] Olsen, L., F.F. Samavati, M.C. Sousa, and J. Jorge (2009). Sketch-based modeling: A survey. *Computers & Graphics* 33: 85–103.

[Orbay and Kara, 2011] Orbay, Günay and Levent Burak Kara (2011). Sketch-based modeling of smooth surfaces using adaptive curve networks. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pp. 71–78, New York, NY, USA. ACM.

[Pan, Reitmayr, and Drummond, 2009] Pan, Qi, Gerhard Reitmayr, and Tom W. Drummond (2009). Interactive model reconstruction with user guidance. In *ISMAR '09: Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pp. 209–210, Washington, DC, USA. IEEE Computer Society.

[Parish and Müller, 2001] Parish, Yoav I. H. and Pascal Müller (2001). Procedural modeling of cities. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 301–308, New York, NY, USA. ACM.

[Pauly et al., 2008] Pauly, Mark, Niloy J. Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J. Guibas (2008). Discovering structural regularity in 3d geometry. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pp. 1–11, New York, NY, USA. ACM.

[Penn et al., 2004] Penn, A., C. Mottram, A. Schieck, M. Wittkämper, M. Störring, O. Romell, A. Strothmann, and F. Aish (2004). Augmented reality meeting table: a novel multi-user interface for architectural design. *Recent Advances in Design and Decision Support Systems in Architecture and Urban Planning* 4.

[Pereira et al., 2004] Pereira, J. P., V. A. Branco, J. A. Jorge, N. F. Silva, T. D. Cardoso, and F. N. Ferreira (2004). Cascading recognizers for ambiguous calligraphic interaction. In *Eurographics Workshop on Sketch-Based Modeling (SBM'04)*.

[Perkunder, Israel, and Alexa, 2010] Perkunder, Helen, Johann Habakuk Israel, and Marc Alexa (2010). Shape modeling with sketched feature lines in immersive 3D environments. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM '10, pp. 127–134, Aire-la-Ville, Switzerland. Eurographics Association.

[Piper, Ratti, and Ishii, 2002] Piper, Ben, Carlo Ratti, and Hiroshi Ishii (2002). Illuminating clay: a 3-d tangible interface for landscape analysis. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 355–362, New York, NY, USA. ACM.

[Pottmann, Grohs, and B., 2009] Pottmann, H., P. Grohs, and Blaschitz B. (2009). Edge offset meshes in laguerre geometry. *Adv. Comp. Math.* . to appear.

[Pottmann et al., 2007] Pottmann, Helmut, Yang Liu, Johannes Wallner, Alexander Bobenko, and Wenping Wang (2007). Geometry of multi-layer freeform structures for architecture. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, p. 65, New York, NY, USA. ACM.

[Pottmann et al., 2008] Pottmann, Helmut, Alexander Schiftner, Pengbo Bo, Heinz Schmiedhofer, Wenping Wang, Niccolo Baldassini, and Johannes Wallner (2008). Freeform surfaces from single curved panels. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pp. 1–10, New York, NY, USA. ACM.

[Pottmann, Schiftner, and Wallner, 2008] Pottmann, Helmut, Alexander Schiftner, and Johannes Wallner (2008). Geometry of architectural freeform structures. *Int. Math. Nachr.* 209: 15–28.

[Pottmann and Wallner, 2008] Pottmann, Helmut and Johannes Wallner (2008). The focal geometry of circular and conical meshes. *Adv. Comp. Math* 29: 249–268.

[Prusinkiewicz and Lindenmayer, 1990] Prusinkiewicz, P. and Aristid Lindenmayer (1990). *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA.

[Regenbrecht, Wagner, and Baratoff, 2002] Regenbrecht, H.T., M. Wagner, and G. Baratoff (2002). Magicmeeting: A collaborative tangible augmented reality system. *Virtual Reality* 6: 151–166. 10.1007/s100550200016.

[Reznik and Mayer, 2007] Reznik, S. and H. Mayer (2007). Implicit shape models, model selection, and plane sweeping for 3d facade interpretation. In *Photogrammetric Image Analysis 2007*, p. 173.

[Ricard, Royan, and Aubault, 2008] Ricard, Julien, Jerome Royan, and Oliver Aubault (2008). Visualization of real cities based on procedural modeling. In *Workshop on Virtual Cityscapes*.

[Sachs, Roberts, and Stoops, 1991] Sachs, Emanuel, Andrew Roberts, and David Stoops (1991). 3-draw: A tool for designing 3d shapes. *IEEE Comput. Graph. Appl.* 11(6): 18–26.

[Santos et al., 2003] Santos, P., H. Graf, T. Fleisch, and A. Stork (2003). 3D interactive augmented reality in early stages of product design. In *Proceedings of the Tenth International Conference on Human-Computer Interaction*, Vol. 4, pp. 1203–1207.

[Santos et al., 2005] Santos, Pedro, André Stork, Joaquim Jorge, and Rodrigo Filipe (2005). An integrated approach to virtual tape drawing for automotive design. In *Proceedings of HCI International 2005*, pp. 1203–1207, Las Vegas, NV, USA.

[Schkolne, 2006] Schkolne, Steven (2006). Making digital shapes by hand. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pp. 84–93, New York, NY, USA. ACM.

[Schkolne, Ishii, and Schroder, 2004] Schkolne, Steven, Hiroshi Ishii, and Peter Schroder (2004). Immersive design of dna molecules with a tangible interface. In *VIS '04: Proceedings of the conference on Visualization '04*, pp. 227–234, Washington, DC, USA. IEEE Computer Society.

[Schkolne, Pruett, and Schröder, 2001] Schkolne, Steven, Michael Pruett, and Peter Schröder (2001). Surface drawing: creating organic 3d shapes with the hand and tangible tools. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 261–268, New York, NY, USA. ACM.

[Schmalstieg, Encarnação, and Szalavári, 1999] Schmalstieg, Dieter, L. Miguel Encarnação, and Zsolt Szalavári (1999). Using transparent props for interaction with the virtual table. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pp. 147–153, New York, NY, USA. ACM.

[Schmidt et al., 2009] Schmidt, Ryan, Azam Khan, Karan Singh, and Gord Kurtenbach (2009). Analytic drawing of 3d scaffolds pp. 1–10.

[Schmidt et al., 2005] Schmidt, Ryan, Brian Wyvill, Mario Costa Sousa, and Joaquim Jorge (2005). Shapeshop: Sketch-based solid modeling with blobtrees. In *2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pp. 53–62.

[Schmitz, Endres, and Butz, 2007] Schmitz, Michael, Christoph Endres, and Andreas Butz (2007). A survey of human-computer interaction design in science fiction movies. In *Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment*, INTETAIN '08, pp. 7:1–7:10, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[Schneider, 1990] Schneider, Philip J. (1990). Graphics gems. chapter An algorithm for automatically fitting digitized curves, pp. 612–626. Academic Press Professional, Inc., San Diego, CA, USA.

[See what you mean, 2005] See what you mean (2005). [Short Film]. Stylewar (Director), Les Telecreateurs (Producer), Swiss.se (Post production), Dassault Systemes (Client).

[See what you mean V2.0, 2007] See what you mean V2.0 (2007). [Short Film]. June Twenty First (Director), June Twenty First (Producer), Dassault Systemes (Client).

[Seichter, 2007] Seichter, Hartmut (2007). Augmented reality and tangible user interfaces in collaborative urban design. In *CAAD futures*, pp. 3–16, Sydney, Australia. Springer, Springer.

[Sharlin et al., 2002] Sharlin, Ehud, Yuichi Itoh, Benjamin Watson, Yoshifumi Kitamura, Steve Sutphen, and Lili Liu (2002). Cognitive cubes: a tangible user interface for cognitive assessment. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 347–354, New York, NY, USA. ACM.

[Shedroff and Noessel, 2012a] Shedroff, Nathan and Christopher Noessel (2012a). *Make it So: Interface Design Lessons from Sci-Fi*. Rosenfeld Media.

[Shedroff and Noessel, 2012b] Shedroff, Nathan and Christopher Noessel (2012b). Make it so: learning from sci-fi interfaces. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pp. 7–8, New York, NY, USA. ACM.

[Sheng, Balakrishnan, and Singh, 2006] Sheng, Jia, Ravin Balakrishnan, and Karan Singh (2006). An interface for virtual 3d sculpting via physical proxy. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pp. 213–220, New York, NY, USA. ACM.

[Shesh and Chen, 2004] Shesh, Amit and Baoquan Chen (2004). Smartpaper: An interactive and user friendly sketching system. *Comput. Graph. Forum* 23(3): 301–310.

[Shin and Igarashi, 2007] Shin, HyoJong and Takeo Igarashi (2007). Magic canvas: interactive design of a 3-d scene prototype from freehand sketches. In *GI '07: Proceedings of Graphics Interface 2007*, pp. 63–70, New York, NY, USA. ACM.

[Shpitalni and Lipson, 1996] Shpitalni, Moshe and Hod Lipson (1996). Optimization-based reconstruction of a 3d object from a single freehand line drawing. *Computer-Aided Design* 28(8): 651–663.

[Singh and Fiume, 1998] Singh, Karan and Eugene Fiume (1998). Wires: a geometric deformation technique. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pp. 405–414, New York, NY, USA. ACM.

[Sinha et al., 2008] Sinha, Sudipta N., Drew Steedly, Richard Szeliski, Maneesh Agrawala, and Marc Pollefeys (2008). Interactive 3d architectural modeling from unordered photo collections. *ACM Trans. Graph.* 27(5): 1–10.

[Smith et al., 2002] Smith, Jeffrey, Jessica Hodgins, Irving Oppenheim, and Andrew Witkin (2002). Creating models of truss structures with optimization. *ACM Trans. Graph.* 21(3): 295–301.

[Smith, Thomas, and Piekarski, 2008] Smith, R.T., B.H. Thomas, and W. Piekarski (2008). Tech note: Digital foam. pp. 35 –38.

[Stava et al., 2010] Stava, O., B. Bene, R. Mech, D. Aliaga, and P. Kristof (2010). Inverse procedural modeling by automatic generation of l-systems. *Computer Graphics Forum* 29(2). To appear.

[Steinicke, Hinrichs, and Ropinski, 2006] Steinicke, Frank, Klaus H. Hinrichs, and Timo Ropinski (2006). A hybrid decision support system for 3d city planning. In *Proceedings of International Commission II Symposium (ISPRS)*, pp. 103–108, Vienna.

[Steinicke et al., 2006] Steinicke, Frank, Timo Ropinski, Klaus H. Hinrichs, and Jörg Mensmann (2006). Urban city planning in semi-immersive virtual reality. In *Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP2006)*, pp. 192–199, Setúbal.

[Stiny and Gips, 1972] Stiny, G. and J. Gips (1972). Shape grammars and the generative specification of painting and sculpture. In Friedman, C. V., editor, *Information Processing '71*, pp. 1460–1465, Amsterdam.

[Stiny, 1975] Stiny, George Nicholas (1975). Pictorial and formal aspects of shape and shape grammars and aesthetic systems. Ph.D. diss.

[Stroud, 2006] Stroud, I. (2006). *Boundary Representation Modelling Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Sutherland, 1963] Sutherland, Ivan E. (1963). Sketchpad: A Man-Machine Graphical Communication System. In Johnson, E. Calvin, editor, *Proceedings of the 1963 Spring Joint Computer Conference*, Vol. 23 of *AFIPS Conference Proceedings*, pp. 329–346, Baltimore, MD. American Federation of Information Processing Societies, Spartan Books Inc.

[Szalavári and Gervautz, 1997] Szalavári, Z and M. Gervautz (1997). The personal interaction panel : a two-handed interface for augmented reality. *Computer Graphics Forum* 16: 335–346.

[Tsang et al., 2004] Tsang, Steve, Ravin Balakrishnan, Karan Singh, and Abhishek Ranjan (2004). A suggestive interface for image guided 3d sketching. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 591–598, New York, NY, USA. ACM.

[Underkoffler and Ishii, 1999] Underkoffler, John and Hiroshi Ishii (1999). Urp: a luminous-tangible workbench for urban planning and design. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 386–393, New York, NY, USA. ACM.

[Van Gool et al., 2007] Van Gool, L.J., G. Zeng, F. Borre, and P. Muller (2007). Invited paper: Towards mass-produced building models. In *Photogrammetric Image Analysis 2007*, p. 209.

[Vogel and Casiez, 2011] Vogel, Daniel and Géry Casiez (2011). Conté: multimodal input inspired by an artist's crayon. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 357–366, New York, NY, USA. ACM.

[Wang, Paris, and Popović, 2011] Wang, Robert, Sylvain Paris, and Jovan Popović (2011). 6D hands: markerless hand-tracking for computer aided design. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pp. 549–558, NY, USA. ACM.

[Weiler, 1985] Weiler, Kevin (1985). Edge-based data structures for solid modeling in curved-surface environments. *IEEE Computer Graphics and Applications* 5(1): 21–40.

[Wesche, 2003] Wesche, Gerold (2003). The toolfinger: supporting complex direct manipulation in virtual environments. In *EGVE '03: Proceedings of the workshop on Virtual environments 2003*, pp. 39–45, New York, NY, USA. ACM.

[Wesche, 2004] Wesche, Gerold (2004). A user interface for free-form shape design at the responsive workbench. *Journal of Computing and Information Science in Engineering* 4(3): 178–185.

[Wesche and Droske, 2000] Wesche, Gerold and Marc Droske (2000). Conceptual free-form styling on the responsive workbench. In *VRST '00: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 83–91, New York, NY, USA. ACM.

[Wesche and Seidel, 2001] Wesche, Gerold and Hans-Peter Seidel (2001). Freedrawer: a free-form sketching system on the responsive workbench. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 167–174, New York, NY, USA. ACM.

[Willmott et al., 2001] Willmott, J., L. I. Wright, D. B. Arnold, and A. M. Day (2001). Rendering of large and complex urban environments for real time heritage reconstructions. In *VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*, pp. 111–120, New York, NY, USA. ACM.

[Wonka et al., 2003] Wonka, Peter, Michael Wimmer, François Sillion, and William Ribarsky (2003). Instant architecture. *ACM Transactions on Graphics* 22(3): 669–677.

[World Builder, 2007] World Builder (2007). [Short Film]. Bruce Branit (Director), Bruce Branit (Producers), Branit VFX (Distributors).

[Xin, Sharlin, and Sousa, 2008] Xin, Min, Ehud Sharlin, and Mario Costa Sousa (2008). Napkin sketch: handheld mixed reality 3d sketching. In *VRST '08: Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pp. 223–226, New York, NY, USA. ACM.

[Yeh and Mech, 2009] Yeh, Yi-Ting and Radomír Mech (2009). Detecting symmetries and curvilinear arrangements in vector art. *Comput. Graph. Forum* 28(2): 707–716.

[Yi, Qin, and Kang, 2009] Yi, Xiao, Shengfeng Qin, and Jinsheng Kang (2009). Generating 3d architectural models based on hand motion and gesture. *Comput. Ind.* 60(9): 677–685.

[Yu and Zhang, 2007] Yu, Jeff J. and Hao Zhang (2007). A prototype sketch-based architectural design system with behavior mode. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling 2007*.

[Zeleznik, Herndon, and Hughes, 1996] Zeleznik, Robert C., Kenneth P. Herndon, and John F. Hughes (1996). Sketch: an interface for sketching 3d scenes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 163–170, New York, NY, USA. ACM.

A

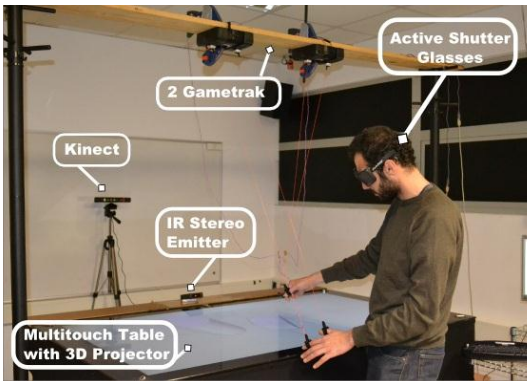
Mockup Builder Short User Guide

Mockup Builder User Guide

1. Introducing the Mockup Builder User Interface

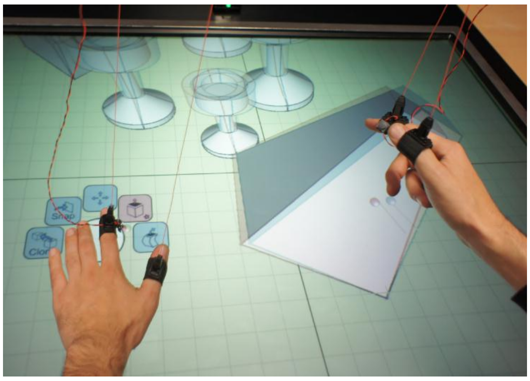
Mockup Builder is a 3D modeling system using a stereoscopic visualization with active shutter glasses. The working space relies on an interactive surface where users can draw and create 3D shapes. Instead of using the mouse and the keyboard, Mockup Builder allows to interact using finger touches on the surface and using hand gestures above the surface in space using two rings on the thumb and index finger of each hand. The thumb ring offers a button to confirm a selection and perform modeling actions.

Overview of the Mockup Builder Modeling environment. The **working space** is defined by an **interactive** surface and the **space above** it.

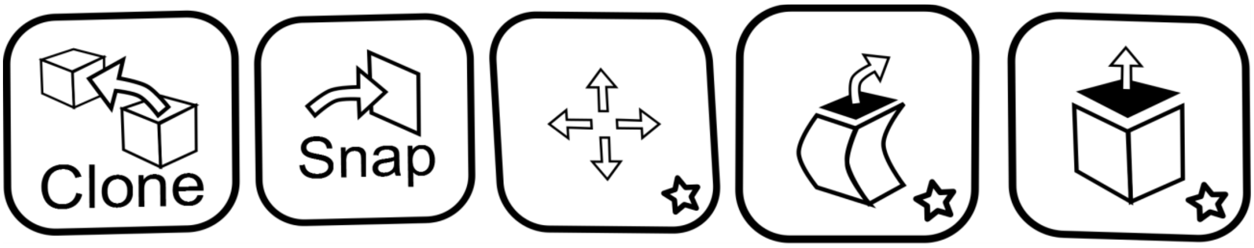


In each hand, the **ring devices** should be worn on thumb and index finger. The ring with button should be used on **the thumb finger** allowing to be **pressed by the index finger**.

To confirm an action on the surface, just touch the multi-touch surface. In the space above it, you just need to **press the button to start** an action and **release it to end the command**.



Main icons of the contextual Menu depending on the active selection.



Object Cloning, Face Snapping, Moving a Face, Curvilinear Extrusion, Linear Extrusion

How to use Mockup Builder?

Mockup Builder proposed an interactive tabletop where all the space is considered as the working space, there is no windows, no toolbars and menus are only visible if invoked by the user when they are needed.
Mockup Builder mainly uses hand and finger gestures as natural as possible such as it is done when we use both hands to interact with pencil and paper. To do so, **we assign different roles to each hand**. While one hand behaves like a tool, the other hand controls the spatial manipulation. Mockup Builder will behave **differently if you are left or right handed adapting itself to your skills.**

If you are Right Handed: (You usually write or draw with your right hand)

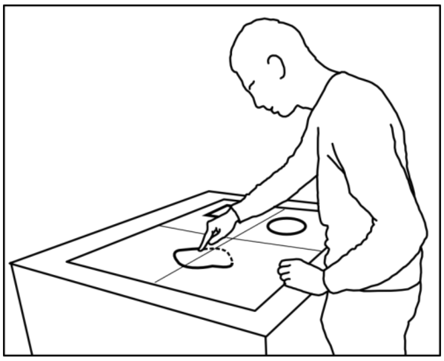
With your Left Hand	With your Right Hand
You can **Translate, Rotate and Scale** Objects You can **Translate, Rotate and Scale** the view of the scene You can **invoke Menus** if something is highlighted by the other hand	You can **Draw** You can **Select** and **Move** faces, edges and vertexes of your 3D model You can **Extrude** a face along its normal direction or defining a curve using gestures in space

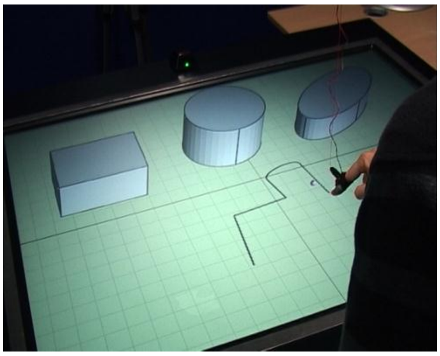
If you are Left Handed: (You usually write or draw with your left hand)

With your Left Hand	With your Right Hand
You can **Draw** You can **Select** and **Move** faces, edges and vertexes of your 3D model You can **Extrude** a face along its normal direction or defining a curve using gestures in space	You can **Translate, Rotate and Scale** Objects You can **Translate, Rotate and Scale** the view of the scene You can **invoke Menus** if something is highlighted by the other hand

2. How to sketch shapes?

Using you finger on the surface
Or Pressing the thumb button, moving and releasing it in the air.

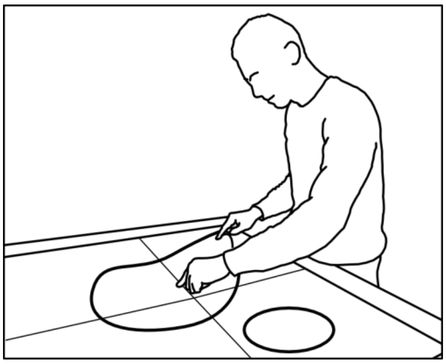


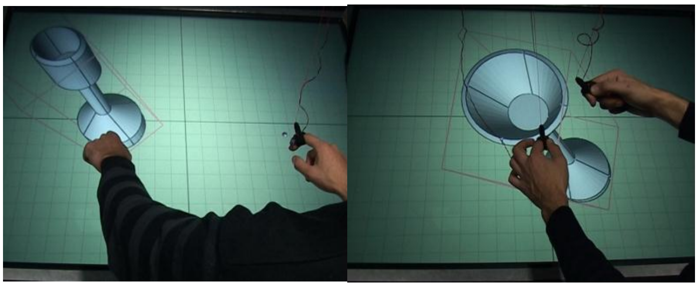


3. How to Translate, Rotate and Scale Objects or Control the View?

Using your finger(s) on the surface **Or** Pressing the thumb button, moving and releasing it in the air.

If an object is below you fingers **the object will be transforms**. **Otherwise** the transformation is applied to the view and the **entire scene is transformed**.

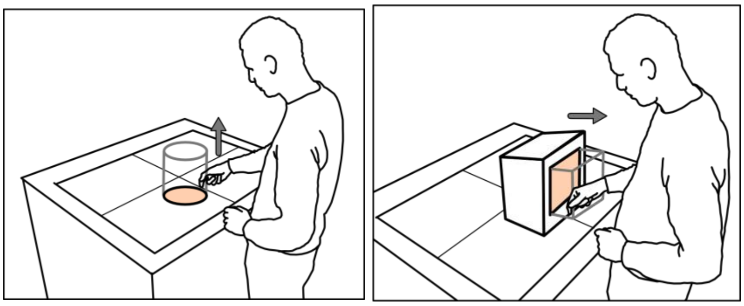


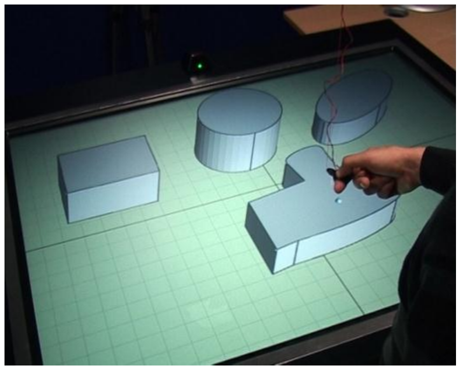


1 finger will Translate, 2 fingers (from 1 ou 2 Hands) will Rotate or Scale like the gestures used by multi-touch mobile phones or tablet devices.

4. **How to extrude a face along its normal direction?**

Pressing the thumb button while highlighting a face, then move your hand, and release the button in space when the face is in the wished position.

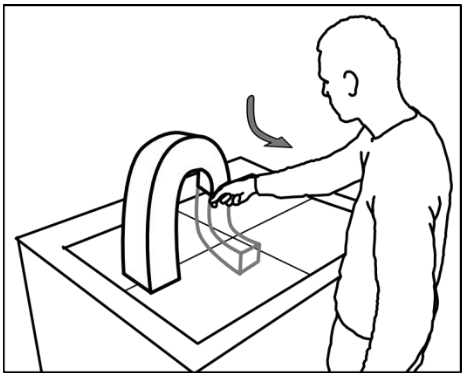


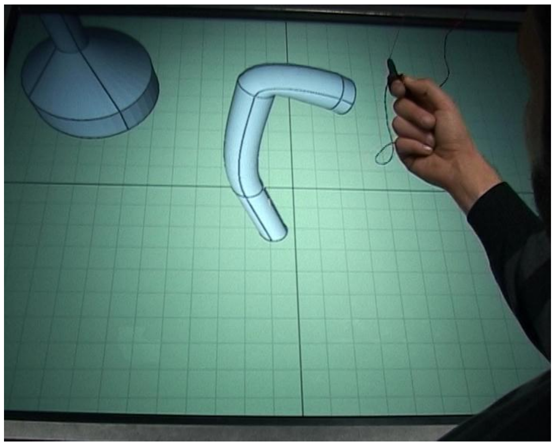


If you want to create **a new extrusion after an extrusion**: invoke the Menu and select the **Linear Extrusion option**.

5. **How to extrude a face along a curved trajectory?**

Invoke the Menu and select the Curvilinear Extrusion option **Then** press the button while highlighting the face, move your hand to define the trajectory and release the button to end.





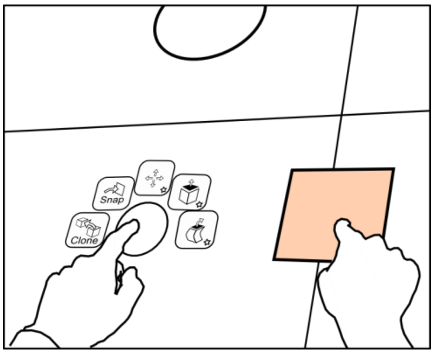
6. **How to Select/Highlight a Face, an Edge or a Vertex or even the entire Object?**

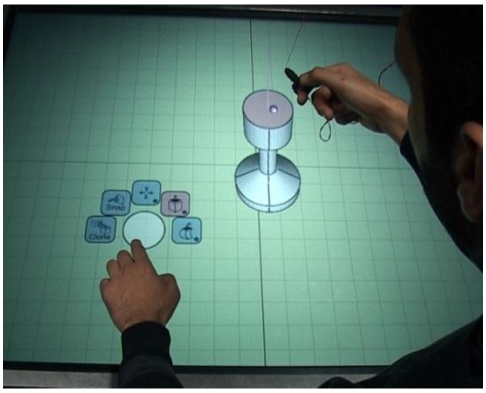
If the face, edge or vertex is on the surface of the multi-touch table **just touch it using your fingers**. If the entity to be select is in the air, just **approach our index finger to** the face, edge or vertex. **Automatically the color of the face, edge or vertex is changed, highlighting part of the object.** **On the surface**, manipulation actions are **performed automatically when touching** the table with your fingers. **In the space**, modeling or manipulation action will only occur if you confirm the selection **using the thumb button** and will be valid or active until you release it. **To select or highlight an entire object:** approach you finger to the object without touching any face, edge or vertex**, a red bounding box will appear around the entire shape**.

7. How to invoke Menus?

First you need to highlight a shape attribute using the drawing hand, and then you need to approach the other hand to the table without touching it. The menu will appear automatically below the moving hand. The menu will disappear if your hand is more distant to the table or if the feature is not highlighted anymore invalidating the selection.

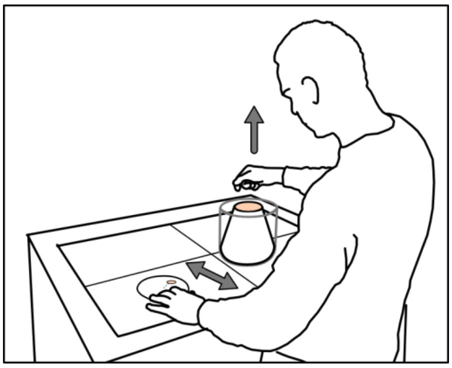
To select an option of the menu, just touch the option with our finger on the surface.

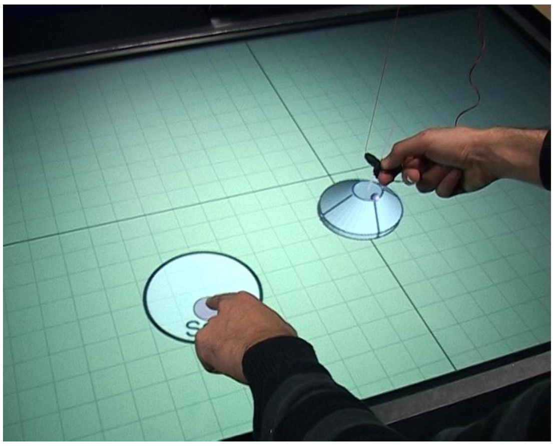




8. How to scale the profile of an object that I am extruding?

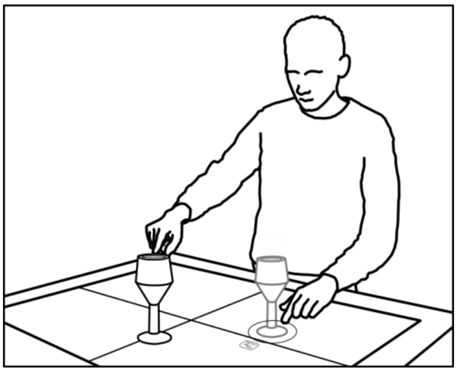
Use the hand which is not extruding to invoke a menu by approaching this hand while extruding with the other. The menu will present a circular button on the surface to control the scaling factor.

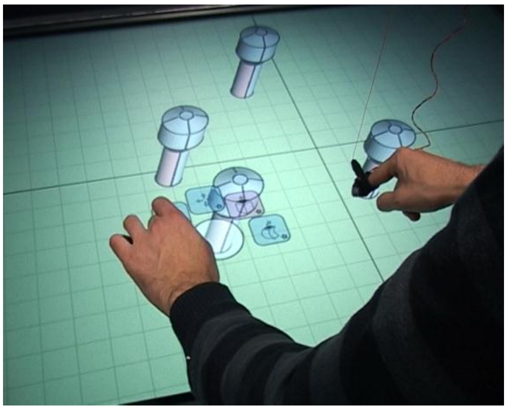




9. How to copy an object?

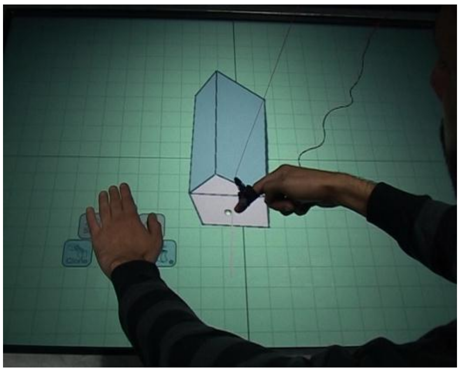
Select the object with the drawing hand and invoke the menu with the other hand. Then, select the cloning option on the Menu.



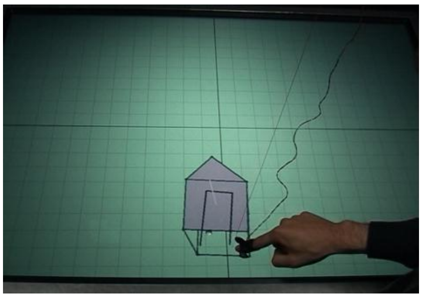


10. I can't draw over a face because it in the air instead of being in the surface!

Select the face with the drawing hand, invoke the menu and select the snapping option.



The face will be automatically aligned with the surface; you can draw on it now.
To get back to the previous orientation, invoke the menu by selecting an empty space on the surface, the menu below the other hand will propose the snapping option to get back.



B

Sketchup Short User Guide

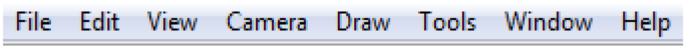
Sketchup User Guide

1. Introduction to the User Interface

The main components of the Sketchup User interface are toolbars, the top menu and the drawing area.

1.1 Top Menu

All Sketchup functionality, commands and definitions are available on the top menu. The following options are available on the top menu: File, Edit, View, Camera, Draw, Tools, Window, and Help.



1.2 Toolbar

The toolbar contains all the items to model or sketch new shapes.

1.2.3 View Control Toolbar

This tool bar allows to Orbit, Translate or Zoom.

1.2.3 Drawing Toolbar

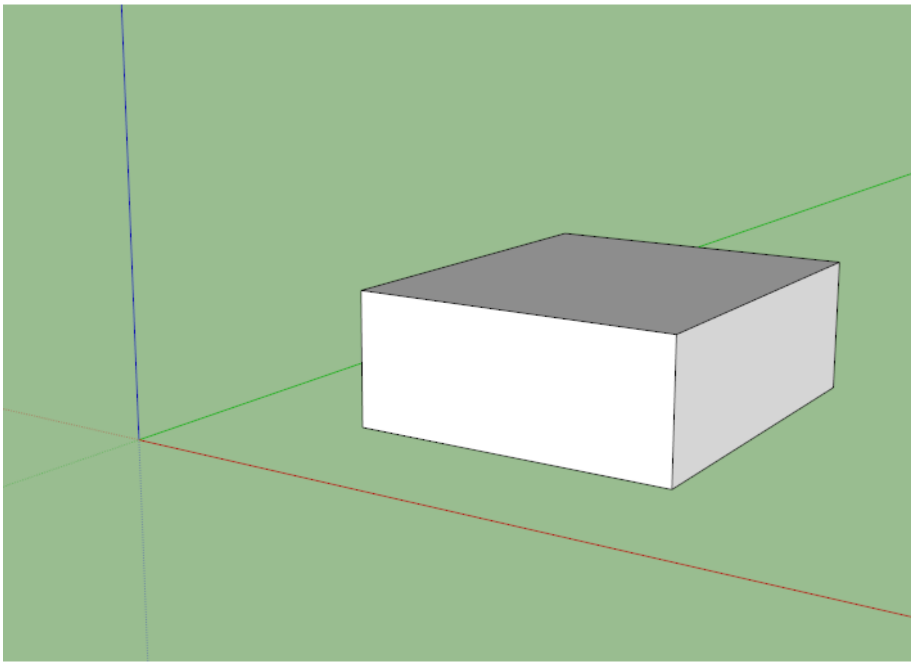
The options of this toolbar allow to activate the line drawing tool, the rectangular tool, the circle tool and the arc tool.

1.2.4 The Geometry Modifier Toolbar

This toolbar prpose options to extrude (push and pull), move, modify the scale or offset a surface.

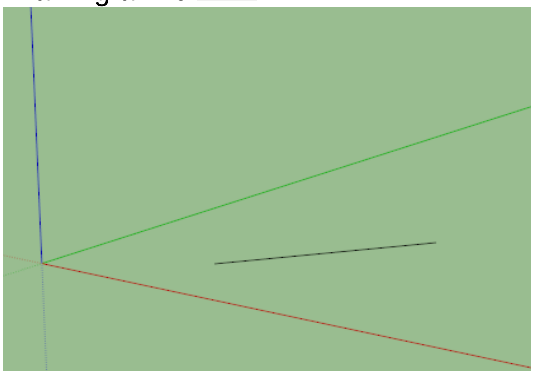
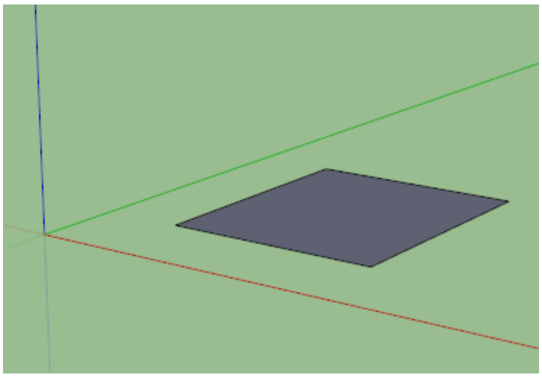
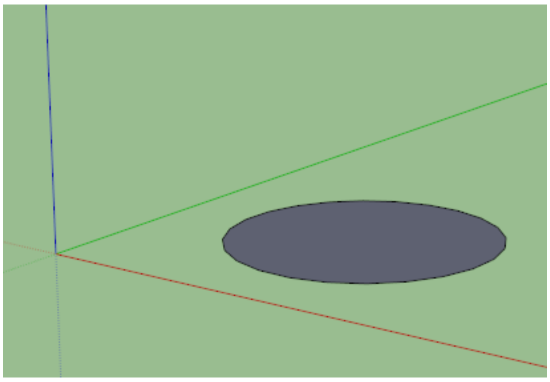
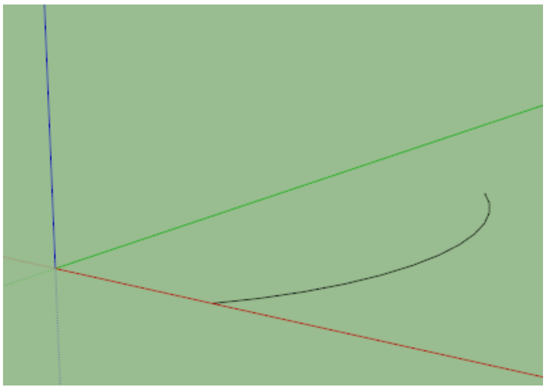
1.3 The Drawing Area

The drawing area is used to create 3D models in 3D. The tridimensional space is identified by its three axes using a different color.
On the drawing area, it is possible to select shapes or features of the shape such as faces, edges and vertexes using the  tool. Selected elements can be copied and pasted in other locations using Cut, Copy and Paste tools.

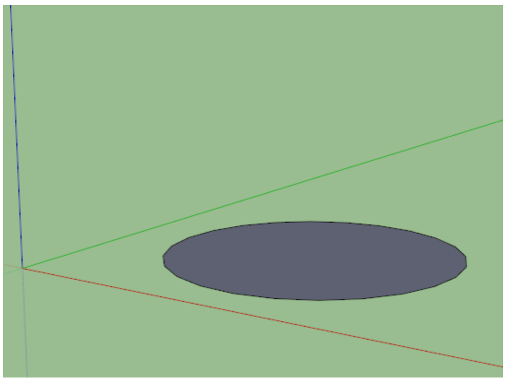
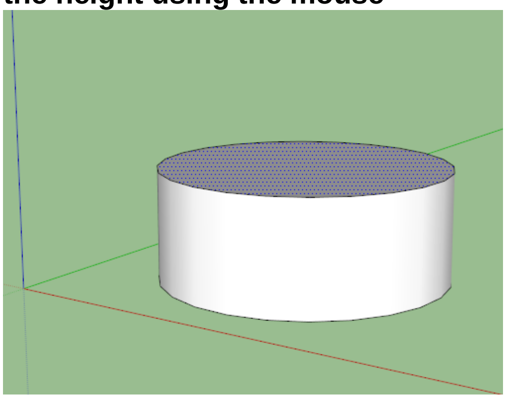


You can draw anywhere on the Drawing Area or even on top of an existing model or its faces. When drawing or manipulating faces, measures are presented or the color changes according to the reference axis. Extremities of the sketch can be readjusted to be align with important features such as the midpoint or to guaranty that two extremities are coincident.

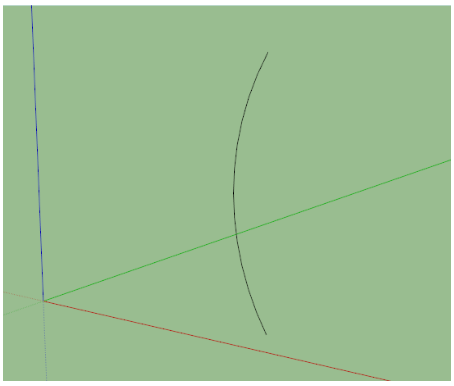
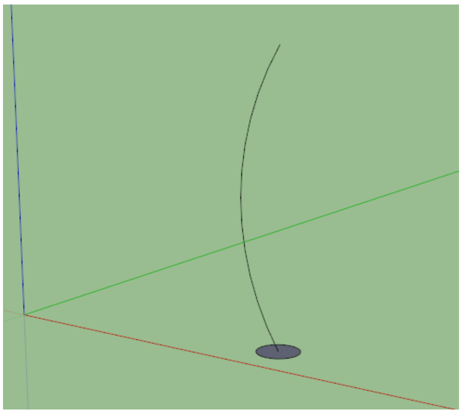
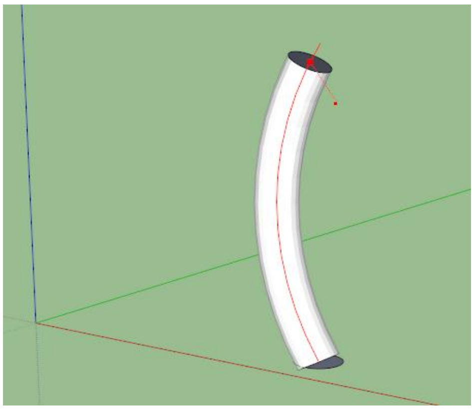
2. Drawing Tool

Drawing a line  	Drawing a rectangle  
Drawing a circle  	Drawing an arc  

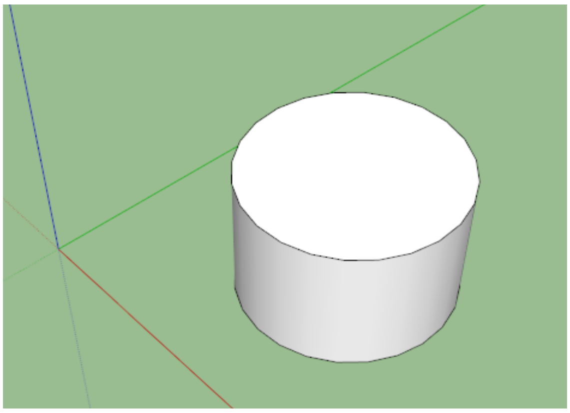
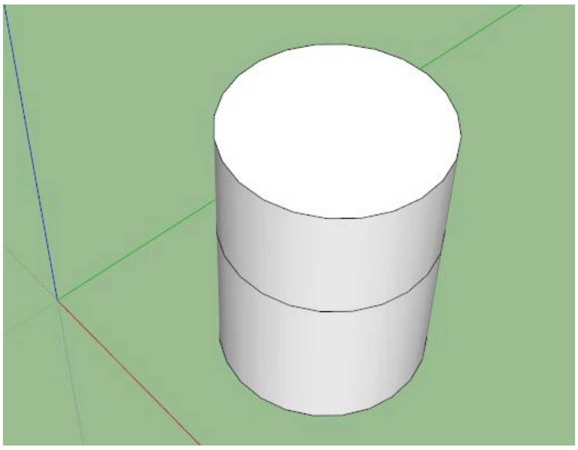
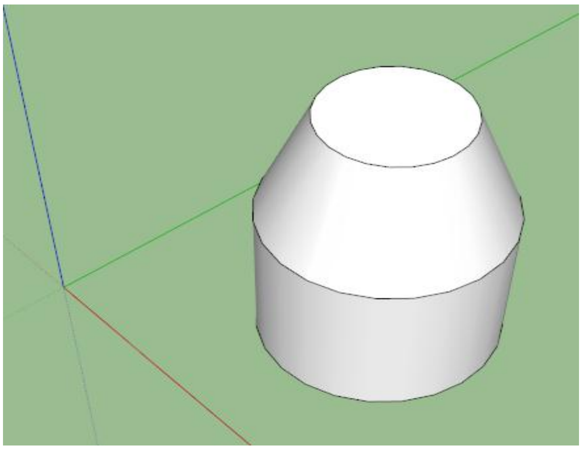
2.5 Extruding 

Start by drawing a closed polygon or shape using the drawing tool 	Apply the  **tool on the face and adjust the height using the mouse** 

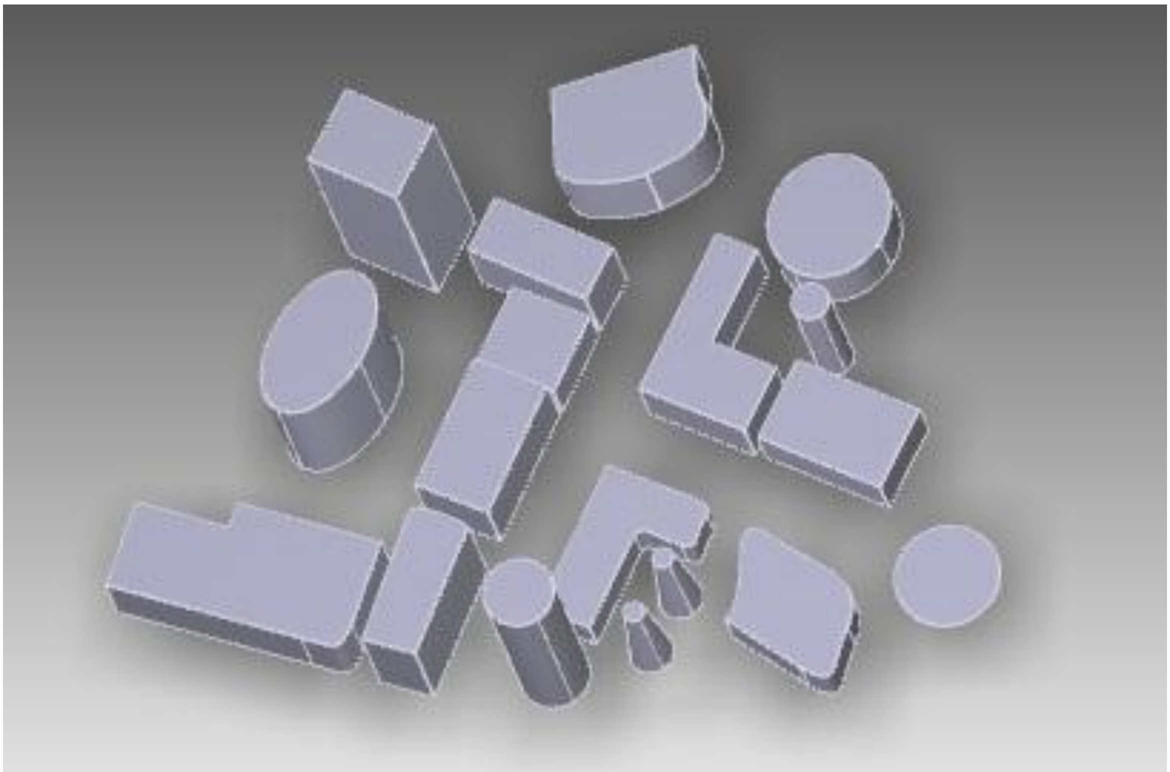
2.6 Extruding along curves using the Follow Me tool (Available on menu Tool)

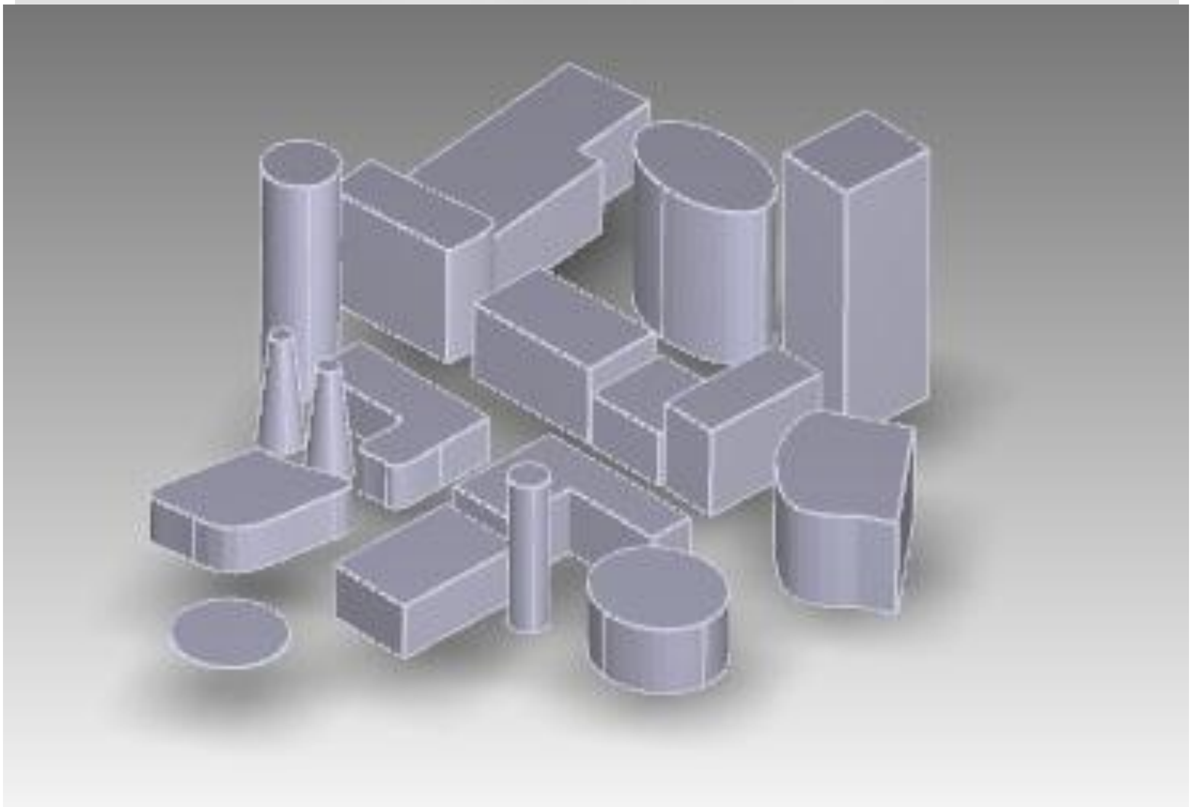
Draw a line or a curve using any tool of Draw menu.	
Create a face using the drawing tool  close to one of the extremities of the path	
Apply the Follow Me tool (Tools > Follow Me) to extrude the face along the path:	

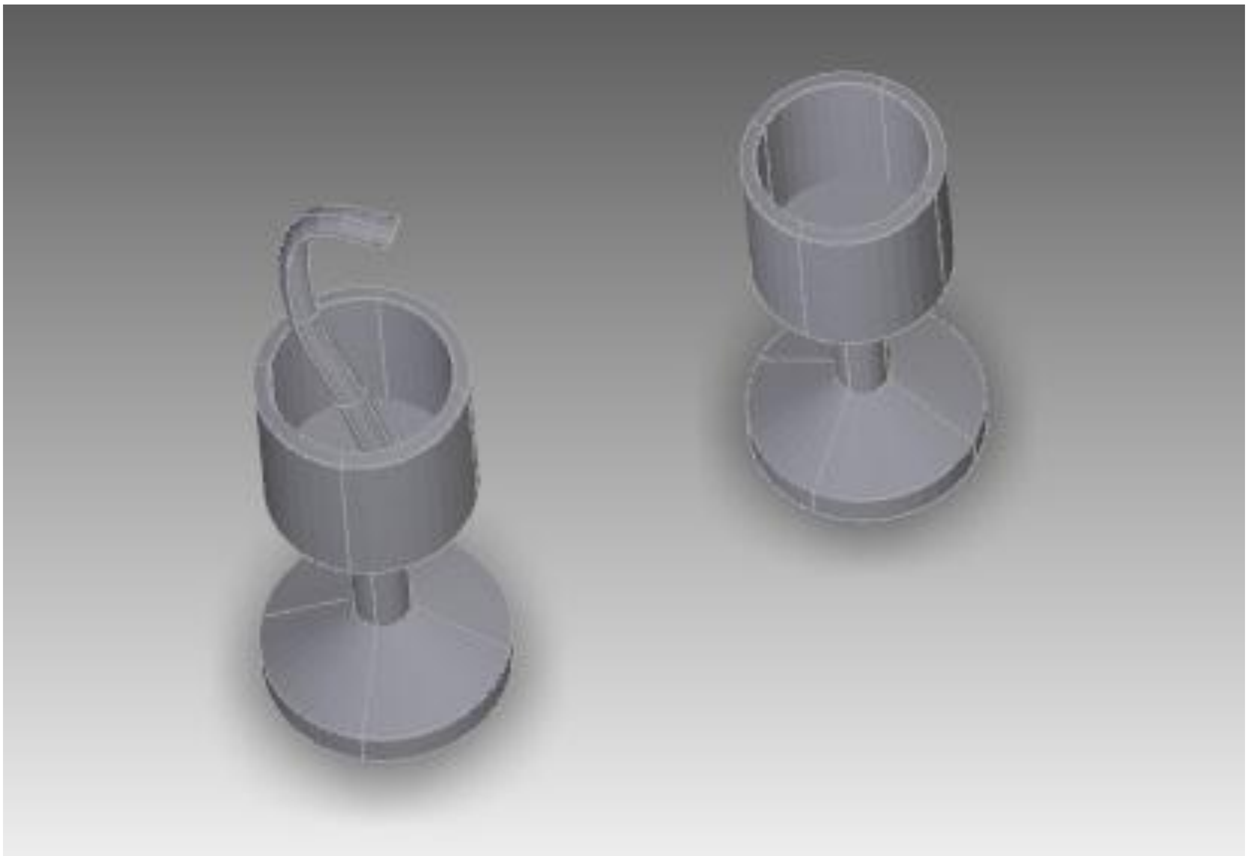
2.6 Start a new extrusion using the  option and pressing the Ctrl key:

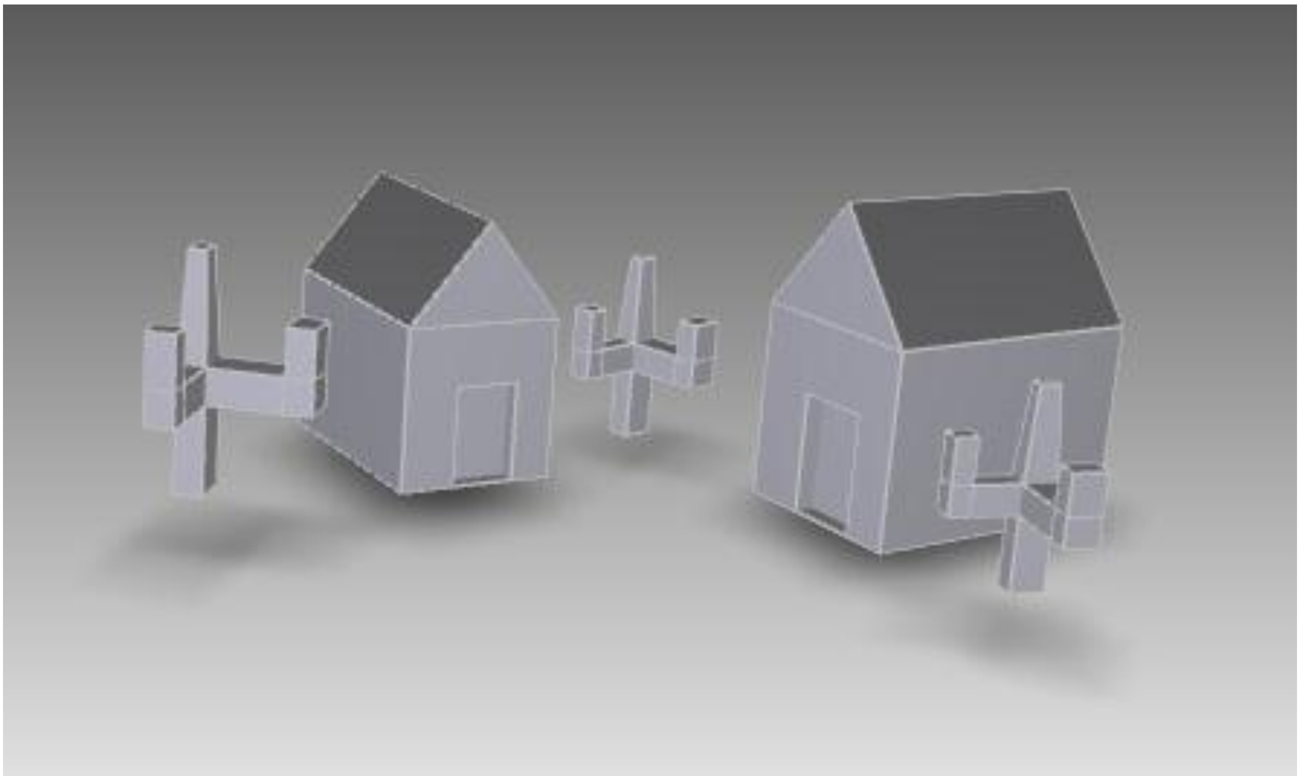
Apply the  **tool** on a 2D object to create an a volumetric shape by using the extrusion.	
While applying the  tool, press the **Ctrl** key to create a new extrusion on top of the previous one.	
Now, the second extrusion can be manipulated leaving the properties of the first extrusion unchanged. Manipulation tools can be apply on the second extrusion. For example, the  tool or *Tools > Move menu option,* can be used on the edge of the top cylinder to increase or reduce its radius.	

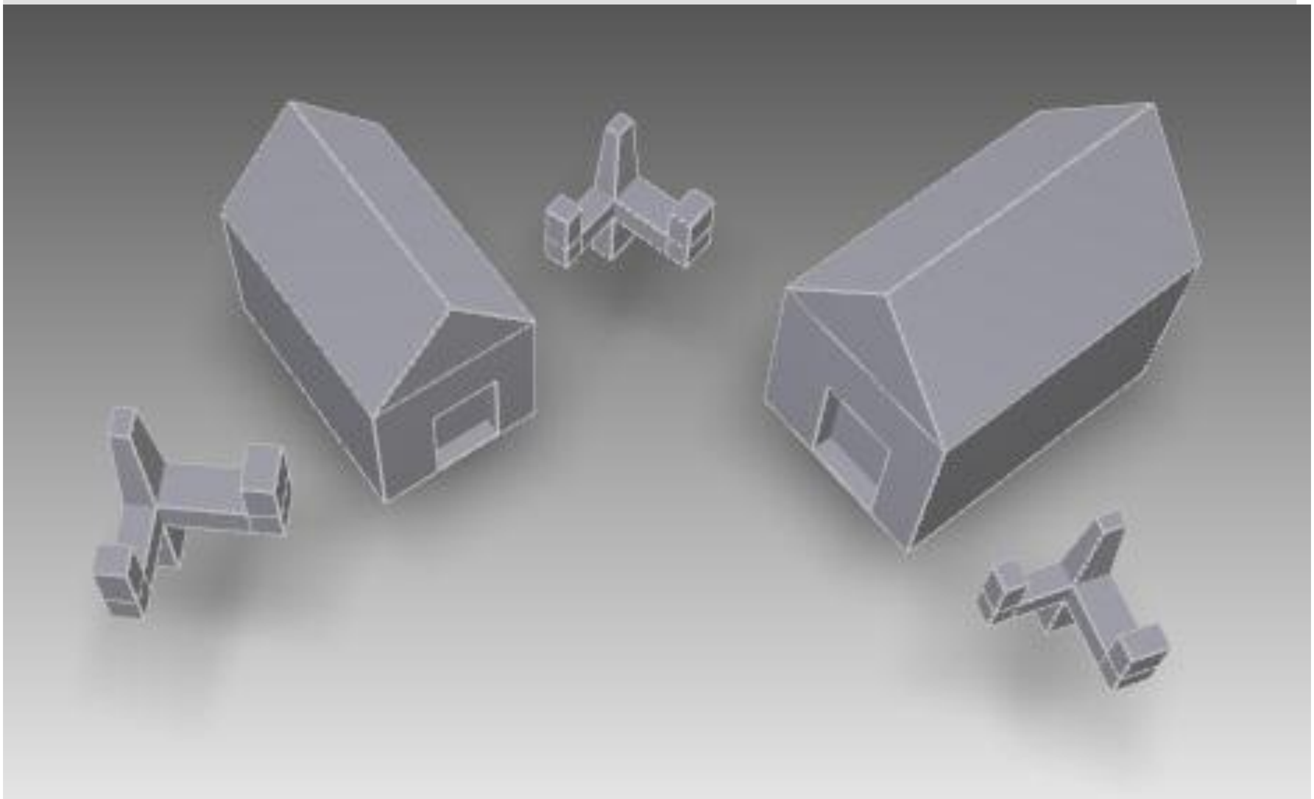
C
User Test Task Screenshots











D
User Test Questionnaire

Direct 3D Modeling User Evaluation

Date: / /2012

Preliminary Questionnaire

Name (optional): ____

Email (optional): ____

Work or Education: ____

Age: ___ Gender: Male ☐ Female ☐ Handedness: Right Hand ☐ Left Hand ☐

Note: Select only one option if not said otherwise

Please specify your experience on the following topics:
Do you have experience on using 3D modeling tools? Never or a little 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ Strongly Skilled
Which 3D modeling tools did you tried before this test? **(multiple selection allowed)** 3Ds Max ☐ Maya ☐ Sketchup ☐ AutoCAD ☐ Blender ☐ Rhino ☐ Unity 3D ☐ Other:______________
Did you have experience on sketching using computers or on the paper? Never or a little 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ Strongly Skilled
Did you already experience any 3D stereoscopic visualization system? Never 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ Almost Daily
Which 3D Visualization setup did you experience before this test? **(multiple selection allowed)** Red/Cyan Anaglyph ☐ 3DTV ☐ 3DMovies ☐ 3DGames ☐ Head Mount Display ☐ Glass Free Stereo ☐
Do you have experience on PC or Console Gaming? Never or a little 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ Almost Daily
Which console did you already try? **(multiple selection allowed)** XBOX, PS3, Wii ☐ Arcade System ☐ Portable Devices ☐ Older Consoles ☐
Did you already try any of the following console accessories? **(multiple selection allowed)** Nintendo Wiimote ☐ Microsoft Kinect ☐ Playstation Move ☐ Playstation EyeToy ☐ None ☐
Do you have experience on Multitouch devices (tablets, phones or other) Never or a little 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ Almost Daily
Which multitouch device did you already used? **(multiple selection allowed)** Phone/SmartPhone ☐ Tablet ☐ Multitouch screens (15-24") ☐ Large Multitouch screens (>24") ☐ None ☐

The following part of the questionnaire presents questions relative to your experience while performing tasks and experimenting freely each modeling system (Mockup Builder and Sketchup). For both systems the questions are exactly the same, please try to differentiate them on your classification. The first part will be relative to the first system you tested and the following relative to the second system. Please identify which system (MockupBuilder or Sketchup) was used at the beginning of each part.

1st System tested: Mockup Builder ☐ Sketchup ☐

General Questions	Strongly disagree / Fully agree
I liked this system	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I find the system fun to use	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I think this system is easy to use	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I felt sickness or fatigue	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I had difficulties to perceive depth	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

Easiness Questions	Strongly disagree / Fully agree
Creation	
Easy to create 2D regular shapes	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create 2D shapes generally	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create 3D simple Shapes	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create curves and polylines	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to define shapes in 3D space	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create or add details on shapes on plane	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create details on space	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Extrusion	
Easy to extrude 2D shapes	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to extrude a shape several times	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to extrude along a curve	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Manipulation	
Easy to move parts of the model such as edge or a vertex	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to place object inside or on top of existing objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to manipulate on scene plane	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to manipulate object in space	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to control size and orientation of an object	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to rotate, move, scale objects as I wanted	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to control scene visualization	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Selection	
Easy to select menu options when needed	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to select modeling operation	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to select faces	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to select edges	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
User Interface	
Easy to use the menu when needed	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to switch between manipulation and shape modeling	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to copy objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to recover from an error or erase unwanted features	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

Perception and Feedback Questions	Strongly disagree / Fully agree
I perceive 3D objects well	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I was difficult to identify what was selected	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
The menu or the toolbar disturbed me when modeling	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
My hands or the cursor disturbed me to visualize of 3D objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
My hands or the cursor disturbed me to select 3D objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
My hands or the cursor disturbed me to visualize 2D Widgets	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

It was easy to perceive relative position between objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
It was easy to perceive relative size between objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
The system is precise enough to create the shape I wanted	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
The system is precise enough to place the object as I wanted	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Other Questions	Strongly disagree / Fully agree
The system allows me to experiment several design possibility	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
The system was interactive or fast enough	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
It was easy to design what I imagine	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
What did you like most on the system? ____________________________ ____________________________ ____________________________	
What did you dislike most on the system? ____________________________ ____________________________ ____________________________	
What would you change on the current system? ____________________________ ____________________________ ____________________________	
What would you add to the system which is not currently available? ____________________________ ____________________________ ____________________________	
Would you like to remove any feature of the current system? ____________________________ ____________________________ ____________________________	
Would you like to leave any suggestion or comment regarding the experience with this 1st system? ____________________________ ____________________________ ____________________________ ____________________________ ____________________________ ____________________________	

The questionnaire relative to the first system tested is finished. Be careful that the following questions are exactly the same relative to the system used on the second test.

2nd System tested: Mockup Builder ☐ Sketchup ☐

General Questions	Strongly disagree / Fully agree
I liked this system	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I find the system fun to use	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I think this system is easy to use	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I felt sickness or fatigue	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I had difficulties to perceive depth	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

Easiness Questions	Strongly disagree / Fully agree
Creation	
Easy to create 2D regular shapes	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create 2D shapes generally	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create 3D simple Shapes	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create curves and polylines	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to define shapes in 3D space	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create or add details on shapes on plane	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to create details on space	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Extrusion	
Easy to extrude 2D shapes	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to extrude a shape several times	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to extrude along a curve	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Manipulation	
Easy to move parts of the model such as edge or a vertex	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to place object inside or on top of existing objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to manipulate on scene plane	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to manipulate object in space	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to control size and orientation of an object	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to rotate, move, scale objects as I wanted	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to control scene visualization	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Selection	
Easy to select menu options when needed	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to select modeling operation	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to select faces	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to select edges	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
User Interface	
Easy to use the menu when needed	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to switch between manipulation and shape modeling	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to copy objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Easy to recover from an error or erase unwanted features	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

Perception and Feedback Questions	Strongly disagree / Fully agree
I perceive 3D objects well	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
I was difficult to identify what was selected	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
The menu or the toolbar disturbed me when modeling	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
My hands or the cursor disturbed me to visualize of 3D objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
My hands or the cursor disturbed me to select 3D objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
My hands or the cursor disturbed me to visualize 2D Widgets	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

It was easy to perceive relative position between objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
It was easy to perceive relative size between objects	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
The system is precise enough to create the shape I wanted	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
The system is precise enough to place the object as I wanted	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Other Questions	Strongly disagree / Fully agree
The system allows me to experiment several design possibility	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
The system was interactive or fast enough	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
It was easy to design what I imagine	1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

What did you like most on the system?

____

____

____

What did you dislike most on the system?

____

____

____

What would you change on the current system?

____

____

____

What would you add to the system which is not currently available?

____

____

____

Would you like to remove any feature of the current system?

____

____

____

Would you like to leave any suggestion or comment regarding the experience with this 1st system?

____

____

____

____

____

____

Thanks you for your valuable contribution.

Bruno Rodrigues De Araujo

(bdearaujo@gmail.com)

E
Initial Questionnaire Data

Name	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
Work or Edu	Bsc CS Student	Architect (Msc)	Bsc CS Student	Msc Archi Student	Phd CS Student	Archtect (Phd Student)	Architect (Phd Student)	Designer (Msc)	Architect (Prof)	Arquitect (Msc)	Bsc CS StudentLEIC	Architect (Msc)	Bsc Archi Student	Bsc CS Student
Age	21	24	23	29	26	32	41	27	48	27	22	26	22	21
Gender	F	F	M	M	M	M	M	M	F	M	M	F	M	M
Handedness	R	R	R	R	R	R	R	R	R	R	R	R	L	R
Which 3D modeling tools did y	NNNNNNY	YYYYNNN	NNNNNNY	YNNYYYY	NNYNNNNY	YYYYNYN	YNNYYYN	NYYNNNN	NNNYYYN	YYYYNNN	NNNNNNN	YNYYNYN	YNYYNNY	NNNNNNY
3Ds Max	0	1	0	1	0	1	1	0	0	1	0	1	1	0
Maya	0	1	0	0	0	1	0	1	0	1	0	0	0	0
Sketchup	0	1	0	0	1	1	0	1	0	1	0	1	1	0
AutoCAD	0	1	0	1	0	1	1	0	1	1	0	1	1	0
Blender	0	0	0	1	0	0	1	0	1	0	0	0	0	0
Rhino	0	0	0	1	0	1	1	0	1	0	0	1	0	0
Unity3D	1	0	1	1	0	0	0	0	0	0	0	0	1	1
Other	0	0	0	0	1	0	0	1	0	0	0	0	0	0
# of Systems	1	4	1	5	2	5	4	3	3	4	0	4	4	1
Which 3D Visualization setup d	NYYYNN	NNYYN	NYNNNN	YNNYYYY	NNNNNN	NNNNNN	NNNNNN	NNYNNN	NNNNNY	NNNNNN	NYYNNN	NNYNNN	NNYNNN	NNYNNN
Red/Cyan Anaglyph	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3DTV	1	0	1	0	0	0	0	0	0	0	1	0	0	0
3DMovies	1	1	0	1	0	0	0	1	0	0	1	1	1	1
3DGames	1	1	0	1	0	0	0	0	0	0	0	0	0	0
Head Mount Displays	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Glass Free Stereo	0	0	0	1	0	0	0	0	1	0	0	0	0	0
# of 3D	3	2	1	5	0	0	0	1	1	0	2	1	1	1
Which console did you already	YNYN	YNYN	YNNY	YYYN	YNNN	YYYY	YNNY	YYYN	NNYN	YYYY	YNYY	YNYN	YNYN	YYYY
XBOX, PS3, Wii	1	1	1	1	1	1	1	1	0	1	1	1	1	1
Arcade System	0	0	0	1	0	1	0	1	0	1	0	0	0	1
Portable Devices	1	1	0	1	0	1	0	1	1	1	1	1	1	1
Older Consoles	0	0	1	0	0	1	1	0	0	1	1	0	0	1
Did you already try any of the	YYYNN	YNNNN	NNYNN	YNNNN	YNYNNN	YNNYN	NYNNN	YNNNN	NNNNNY	YNYYN	NYNNN	NNYNNN	NYNYN	YYYYN
Nintendo Wiimote	1	1	0	1	1	1	0	1	0	1	0	0	0	1
Microsoft Kinect	1	0	0	0	0	0	1	0	0	0	1	0	1	1
Playstation Move	1	0	1	0	1	0	0	0	0	1	0	1	0	1
Playstation EyeToy	0	0	0	0	0	1	0	0	0	1	0	0	1	1
None	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	3	1	1	1	2	2	1	1	1	3	1	1	2	4
Which multitouch device did y	YYYNN	YYYYN	YNNN	NYNNN	YYNYN	YYYYN	YYNNN	YNNNN	NNYNN	YYNNN	YYNNN	YNNNN	YNNNN	YNNNN
Phone	1	1	1	0	1	1	1	1	0	1	1	1	1	1
Tablet	1	1	0	1	1	1	1	0	0	1	1	0	0	0
15"-24" MT	1	1	0	0	0	1	0	0	1	0	0	0	0	0
>24" MT	0	1	0	0	1	1	0	0	0	0	0	0	0	0
	1	1	0	0	1	1	0	0	1	0	0	0	0	0
None	0	0	0	0	0	0	0	0	0	0	0	0	0	0

F

First System Questionnaire Data

Name	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
1st System	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP	MOCKUP	SKETCHUP	MOCKUP	SKETCHUP	SKETCHUP	MOCKUP	MOCKUP	MOCKUP	MOCKUP	SKETCHUP
1	3	5	3	4	5	5	5	5	2	4	4	5	4	4
2	2	5	3	4	5	5	3	5	4	4	4	4	4	5
3	4	5	3	4	4	4	3	5	4	3	3	3	3	3
4	3	1	4	1	3	1	3	1	3	3	1	1	1	1
5	2	1	4	1	3	3	1	1	2	2	1	1	1	2
6	5	5	4	5	4	5	5	5	5	4	4	5	5	5
7	5	4	4	5	4	5	5	4	5	3	4	4	4	5
8	4	5	4	5	5	4	3	4	5	3	4	4	4	4
9	4	4	3	5	5	3	5	3	5	3	5	3	4	3
10	4	4	3	5	2	3	3	4	4	3	4	3	4	3
11	3	4	2	5	4	2	3	4	5	2	3	2	3	2
12	3	4	2	5	1	2	2	3	3	1	3	3	3	2
13	5	5	4	5	5	5	3	4	5	4	5	5	5	5
14	4	5	4	5	4	4	2	4	5	4	4	4	4	4
15	4	5	3	5	4	2	3	2	4	4	4	3	4	3
16	4	5	3	5	4	4	3	4	4	3	4	3	4	3
17	3	5	4	5	5	3	3	3	3	4	3	3	2	3
18	4	5	4	5	5	3	3	4	3	4	4	4	3	4
19	3	5	4	5	3	3	3	4	3	3	4	4	4	4
20	2	5	3	5	5	3	2	3	4	3	4	4	4	3
21	2	5	3	5	4	3	2	3	3	3	4	4	4	3
22	3	5	4	5	3	3	2	2	5	4	4	5	4	4
23	4	5	4	5	4	4	2	4	4	4	3	3	4	4
24	4	5	4	5	5	4	2	4	4	4	3	4	4	4
25	5	5	5	5	5	4	2	5	5	3	2	3	3	4
26	3	5	4	5	4	4	1	5	5	2	2	3	3	4
27	4	5	3	5	4	5	2	4	4	4	3	3	4	4
28	4	5	4	5	3	3	2	4	4	5	4	3	3	4
29	5	5	4	5	5	5	3	5	4	4	4	4	4	4
30	4	5	4	5	4	5	1	5	4	1	4	2	2	4
31	4	5	3	5	5	5	5	5	4	5	5	4	5	4
32	2	1	2	1	1	2	3	1	2	2	3	3	2	3
33	1	1	3	1	1	1	2	2	2	1	4	1	2	2
34	2	1	3	1	1	1	2	1	3	3	2	4	2	2
35	2	1	2	1	1	1	2	1	3	3	4	4	2	2
36	2	1	2	1	3	1	2	1	3	3	2	2	2	2
37	3	4	2	5	5	4	2	5	4	5	5	4	4	2
38	3	4	3	5	5	4	2	5	4	5	5	4	4	3
39	3	4	3	5	4	4	2	5	2	3	4	3	3	3
40	3	3	4	5	4	4	2	3	2	2	5	4	3	3
41	4	4	3	5	5	4	5	3	4	3	4	4	4	4
42	4	4	3	2	5	4	3	4	4	4	4	4	4	4
43	3	4	2	4	4	4	3	4	3	4	4	3	3	3

G
Second System Questionnaire Data

Name	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
2nd Syster	MOCKUP	MOCKUP	MOCKUP	MOCKUP	SKETCHUP	MOCKUP	SKETCHUP	MOCKUP	MOCKUP	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP	MOCKUP
1	5	3	4	4	3	5	5	3	5	4	4	5	5	4
2	5	3	4	5	1	5	3	5	5	4	3	3	5	5
3	4	3	3	4	2	4	3	3	3	4	3	5	5	3
4	2	1	5	2	4	1	2	1	1	1	1	1	1	3
5	1	1	3	1	5	1	1	2	2	2	1	1	1	2
6	5	3	4	3	5	4	5	4	3	5	3	5	5	4
7	5	3	4	5	1	4	5	3	5	2	3	5	5	4
8	5	4	4	4	5	5	5	3	5	5	4	5	5	3
9	4	4	4	4	1	4	3	3	5	2	2	3	5	3
10	5	3	3	1	1	4	5	2	4	3	4	3	5	4
11	4	4	4	4	3	3	4	2	4	3	4	4	5	3
12	3	4	2	3	3	4	4	1	4	2	4	3	5	4
13	5	4	4	5	5	5	4	5	4	5	4	5	5	3
14	5	4	4	5	5	4	4	4	5	4	4	5	5	3
15	5	3	3	1	2	5	4	3	4	4	3	3	5	3
16	4	4	3	4	1	5	2	2	4	4	3	4	4	3
17	3	2	2	3	1	2	2	2	4	3	3	3	5	4
18	4	3	3	4	1	5	2	5	4	4	4	4	5	5
19	4	3	4	4	1	5	2	5	5	3	4	4	5	5
20	4	3	4	4	3	3	2	5	5	3	4	4	5	5
21	5	4	4	4	2	4	2	5	5	3	4	4	5	5
22	5	3	3	4	1	5	4	5	5	3	4	5	5	5
23	5	4	4	5	4	5	3	3	4	4	4	4	5	3
24	5	4	4	5	4	5	3	3	4	4	4	5	5	3
25	4	4	3	4	5	4	3	3	5	5	4	5	5	3
26	3	4	2	3	5	3	3	2	4	4	4	5	5	2
27	4	3	4	4	3	5	1	2	4	4	5	4	5	3
28	4	3	4	4	2	3	2	2	5	5	5	4	5	4
29	5	4	5	5	5	3	3	5	5	4	4	5	5	3
30	3	1	3	1	5	2	1	1	1	4	4	5	5	1
31	5	4	4	5	2	5	5	4	5	5	5	3	5	5
32	4	2	3	4	1	3	3	1	2	2	2	1	1	2
33	1	1	2	1	1	2	3	1	3	1	2	1	1	1
34	3	1	5	4	1	2	3	2	3	1	2	1	1	5
35	3	4	5	3	1	3	3	1	3	1	2	1	1	5
36	3	1	3	3	1	3	3	3	4	1	2	1	1	3
37	5	2	3	5	1	5	5	4	3	3	4	5	5	4
38	5	3	3	5	3	5	5	4	5	3	4	4	5	5
39	4	2	2	4	1	3	5	3	4	4	2	4	4	4
40	4	2	3	4	1	3	5	5	4	4	3	4	4	4
41	5	4	4	3	3	4	5	3	5	3	3	4	5	5
42	5	2	3	4	4	3	5	3	5	4	2	4	5	4
43	5	3	3	4	1	3	3	3	3	4	2	4	5	3

H
Final Textual Questionnaire Data

Name	P1	P2	P3	P4	P5	P6	P7
System	MOCKUP	MOCKUP	MOCKUP	MOCKUP	MOCKUP	MOCKUP	MOCKUP
What Like Most	Easiness in drawing and Modelling	The concept	Perceive 3D	Manipulation and 3D perception	Extrusion,3D View, 3D Manipulations	3D experience allows to have a more realitic and accurated perception of 3D.	The emersive and scale capabilities
What Dislike Most	The objects move a little and could make people nauseas	NA	Fatigue with 3D view	Lack of rigour when creating drawings	Usage of left hand for move, Object Rotation on Space (table ok)	3D Glasses a bit disturbing especially when looking away from the table	Menu interface interaction, should be all time present, selecting objects is dificult
What Would Change	Some accuracy on selecting 3D objects	NA	Grab faces and edges	NA	NA	NA	Menus, almost everything should be cpntroled with the predominand hand, select the
New Features Suggestion	NA	NA	NA	3D snapping, Booleans, parametric objects (primtives)	Less restrictive splitting and shape inside shapes	Undo, and back to previous view	Undo and Optimize Erase
Remove any feature	No, it is just fine	NA	NA	No	No	No	NA
Additional Suggestion Comments	NA	NA	NA	NA	NA	NA	More time to learn the setings following a diferent strategie
System	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP
What Like Most	The ability of making shapes as I wanted	NA	Easy to draw	More Practical	Extrusion, Lines, Shape Primitives	It is intuitive even when I tried somethibng that I did not read on the manual	NA
What Dislike Most	Dificult to manipulate shapes and make extrusions to move	NA	Perspective Camera	Operation access using the toolbar	Hard to create custom shapes, Rotate, Move, Control View	the navigation is not intuitive	NA
What Would Change	Tips when we move they go all the other way around	NA	3D perception	A better toolbar	NA	The navigation system	NA
New Features Suggestion	NA	NA	NA	Other commands existing on other CAD systems	Desactivate usage of Constraints	freeform tool	NA
Remove any feature	No, I think all iof them are usefull	NA	NA	No	NA	No	NA
Additional Suggestion Comments	3D shapes is not evolving and for me does not give a very good vision of depth	NA	NA	NA	NA	No	NA

Name	P8	P9	P10	P11	P12	P13	P14
System	MOCKUP	MOCKUP	MOCKUP	MOCKUP	MOCKUP	MOCKUP	MOCKUP
What Like Most	Move and scale objects	Freedom to draw (in movement)	The interactivity and easy mapping of ideas to commands	3D View, Interactivity, Realism	Fun to work, could benefit 3D modeling in future with improvements	NA	The way you can manipulate objects with your fingers is awesome
What Dislike Most	lack of precision of the system	Nothing	Lack of precision on view control	Selection of edges or faces and menu opening	Sometime Dificult to coordinate both hands due to misidentification problems	NA	Aligning the 3D cursor with objects so you can use snap
What Would Change	Snapping system to draw on faces	Add rigor and add the measures o fthe extrusions	More Controls, betterway to erase (faster)	Face Manipulation on 3D not so intuitive (usage of "snapping")	NA	NA	Change the way an object is erased, the current method does not work everytime, may be a new tool like an erasor should be adequate
New Features Suggestion	Undo	Visualization types to understand if I am manipulating the interior of an object	Undo, very simple menus including basic primitives (squares, circles)	Erase button on the menu	Undo and usage of measures	NA	Undo Option
Remove any feature	Reducing Menu size and make always visible	Erase entities from space and undo oprations	NA	No all are necessary	No	NA	No
Additional Suggestion Comments	More direct menus than sketchup and too much menus, the lack o precision make the experience less direct and fun	Add more tutorials to orient the usage of comands, and a second monitor for the experience	NA	NA	Very Interesting Experience since it was possible to compare easyness and dynamism between two diferent 3D modelling System	NA	No
System	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP	SKETCHUP
What Like Most	Precise Selection of faces, edges and vertices	fast to model	Fast to test simple ideas	3D Manipulations, Menu, Change between Commands	Easy to create shapes, Very intuitive	NA	The extrude system
What Dislike Most	Creative Limitation regarding curves	Not knowing how it is structured and organized, what is inside each menu and the name of the commands are not normalized	Very simple and not precise	Rotate is hard and curve creation and manipulation in space is not intuitive	NA	NA	It is too hard to crate tiny details on objects
What Would Change	To be able to create more original shapes	I do not think I would use sketchup for my daily activities	Adequate for simple tasks, need more precision and more creative freedom for more complex projects	Object Manipulation regarding predefine axis (rotation for example) should be change may be using keyboard keys	Rendering Limitations	NA	NA
New Features Suggestion	Animation system	Entity edition	NA	NA	NA	NA	A spectactor like view that you can control with "WASD" and the mouse, using the tools to rotate and move feels too clumsy
Remove any feature	No	NA	NA	No	NA	NA	No
Additional Suggestion Comments	It was direct tho use the system since the commands are natural to me	NA	NA	NA	Being experienced on Sketchup the exercise was very easy than MockupBuilder. Although I believe that in the future with improvements and less flaws, it will be a very dynamic system and will better convey 3D dimensionality notions	NA	No

I

Task Timing Data

TASK DURATION IN SECONDS

Training	Sketchup	Mockup
P1	268.59	418.75
P2	641.14	946.52
P3	314.03	674.19
P4	1078.01	1118.63
P5	374.26	1314.64
P6	168.54	1030.33
P7	540.51	207.53
P8	181.79	1132.43
P9	720.05	1164.39
P10	723.35	1188.53
Avg.	501.03	919.59
SD	292.31	365.80
95%CI	181.18	226.73

Task1	Sketchup	Mockup
P1	793.71	511.19
P2	1079.69	335.09
P3	762.11	483.78
P4	339.19	460.18
P5	657.89	411.34
P6	566.13	286.75
P7	269.63	393.68
P8	490.80	504.48
P9	612.61	688.70
P10	322.53	402.96
Avg.	502.30	447.81
SD	247.59	111.36
95%CI	153.46	69.02

Task2	Sketchup	Mockup
P1	434.31	374.00
P2	220.21	404.11
P3	472.71	443.38
P4	360.93	321.00
P5	413.73	344.41
P6	498.67	379.68
P7	1095.42	673.50
P8	230.74	324.96
P9	905.11	461.74
P10	245.08	353.88
Avg.	487.69	408.07
SD	291.50	104.37
95%CI	180.67	64.69

Task3	Sketchup	Mockup
P1	495.15	448.31
P2	309.14	465.54
P3	348.90	1089.22
P4	300.00	368.85
P5	250.15	358.31
P6	314.61	549.41
P7	521.81	1150.89
P8	260.60	427.77
P9	737.94	998.70
P10	305.95	500.69
Avg.	384.42	635.77
SD	154.40	313.40
95%CI	95.70	194.25