Mixed Physical and Virtual Design Environments for Digital Fabrication

Christian Weichel
B.Sc.
Hochschule Furtwangen University

This dissertation is submitted for the degree of
Doctor of Philosophy

School of Computing and Communications
Lancaster University

September 2015

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as listed below.

All of this work was done under the supervision of Hans Gellersen. Chapter 3 contains work done in collaboration with Manfred Lau. Chapter 4 contains work done in collaboration with Jason Alexander and Abhijit Karnik. Chapter 5 contains work done in collaboration with Manfred Lau and David Kim. Chapter 6 contains work done in collaboration with John Hardy and Jason Alexander.

Christian Weichel
September 2015

Es ist nicht nur der Künstler, der alles Größte in seinem Leben der Phantasie verdankt, sondern überhaupt jeder schöpferische Mensch. Das dynamische Prinzip der Phantasie ist das Spielerische, das auch dem Kinde eignet, und als solches ebenfalls unvereinbar mit dem Prinzip ernster Arbeit erscheint. Aber ohne dieses Spiel mit Phantasien ist noch nie ein schöpferisches Werk geboren worden. Wir verdanken dem Imagionationsspiel unabsehbar viel.

It is not only the artist, who owes all great things in his life to fantasy, but indeed every creative person. The dynamic principle of fantasy is play, which also suits the child, and as such seems irreconcilable with the principle of serious work. However, without this play with fantasies, no creative work has ever been born. We owe the play with imagination incalculably.

– Carl Jung [1, p. 65]

Acknowledgements

This thesis has one author, yet it is an achievement of many. Without any one of the below mentioned – and the countless many who remain unnamed – none of this would exist.

Firstly, I would like to thank my supervisor *Hans Gellersen* for giving me the opportunity to pursue this research endeavor in the first place; for giving me the freedom to engage in a topic of my own choosing. Thank you for your support, supervision and patience throughout the past four years. Second, I am indebted to my second supervisor *Jason Alexander* for supporting me throughout this thesis, especially during the final phase. Thank you very much for you invaluable feedback and advice. Third, thank you *Manfred Lau* for helping me get started in this topic. I have found all our discussions invaluable.

Secondly, I am grateful to all members of the EIS group (and beyond) who I am honored to call my friends. One would be hard pressed to find an environment more welcoming, friendly, stimulating and supportive than this group. Thank you for the countless discussions, encouragement and support. *Pauline Anthonysamy* and *Mélodie Vidal*, thank you for becoming close friends, for the many conversations and hugs. Mélodie and *Ken Pfeuffer*, thank you for sharing many great hours and our house(s). *John Hardy*, thank you for making this stressful last build a pleasant experience through your open cordiality. Thanks to *Steven Houben* for introducing me to HCI theory and for our philosophy of science discussions. Also thanks to all iCareNet members who made this journey even more enjoyable.

Thirdly, I am forever indebted to my family: to my sisters *Agnes* and *Stefanie*, to my partner *Ioana* and my parents *Ruth* and *Bernhard*. The latter two who have been, and continue to be, more influential to me than anyone else. Thank you for your steady support, and believing in me. I can not overstate my gratitude. Ioana, you manage to keep me grounded when I need it most; it will be my greatest joy to spend a life with you.

To my parents.
To Ioana.

Abstract

Digital Fabrication (3D printing, laser-cutting or CNC milling) enables the automated fabrication of physical objects from digital models. This technology is becoming more readily available and ubiquitous, as digital fabrication machines become more capable and affordable. When it comes to designing the objects that are to be fabricated however, there are still barriers for novices and inconveniences for experts.

Through digital fabrication, physical objects are created from digital models. The digital models are currently designed in virtual design environments, which separates the world we design in from the world we design for. This separation hampers design processes of experienced users and presents barriers to novices. For example, manipulating objects in virtual spaces is difficult, but comes naturally in the physical world. Further, in a virtual environment, we cannot easily integrate existing physical objects or experience the object we are designing in its future context (e.g., try out a game controller during design). This lack of reflection impedes designer's spatial understanding in virtual design environments. To enable our virtual creations to become physical reality, we have to posses an ample amount of design and engineering knowledge, which further steepens the learning curve for novices. Lastly, as we are physically separated from our creation – until it is fabricated – we loose direct engagement with the material and object itself, impacting creativity.

We follow a research through design approach, in which we take up the role as interaction designers and engineers. Based on four novel interaction concepts, we explore how the physical world and design environments can be brought closer together, and address the problems caused their prior separation. As engineers, we implement each of these concepts in a prototype system, demonstrating that they can be implemented. Using the systems, we evaluate the concepts and how the concepts alleviate the aforementioned problems, and that the design systems we create are capable of producing useful objects.

In this thesis, we make four main contributions to the body of digital fabrication related

Human-Computer Interaction (HCI) knowledge. Each contribution consists of an interaction concept which addresses a subset of the problems, caused by the separation of virtual design environment, and physical target world. We evaluate the concepts through prototype implementations, example walkthroughs and where appropriate user-studies, demonstrating how the concepts alleviate the problems they address. For each concept and system, we describe the design rationale, and present technical contributions towards their implementation.

The results of this thesis have implications for different user audiences, design processes, the artifacts users design and domains outside of digital fabrication. Through our concepts and systems, we lower barriers for novices to utilize digital fabrication. For experienced designers, we make existing design processes more convenient and efficient. We ease the design of artifacts that reuse existing objects, or that combine organic and geometrically structured design. Lastly, the novel interaction concepts (and on a technical level, the systems) we present, which blur the lines between physical and virtual space, can serve as basis for future interaction design and HCI research.

Publications

- *Christian Weichel*, John Hardy, Jason Alexander and Hans Gellersen. 2015. ReForm: Integrating Physical and Digital Design through Bidirectional Fabrication. In Proceedings of the 28th annual ACM symposium on User interface software and technology (UIST '15).

- *Christian Weichel*, Jason Alexander, Abhijit Karnik and Hans Gellersen. 2015. SPATA: Spatio-Tangible Tools for Fabrication-Aware Design. In Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '14).

- *Christian Weichel*, Manfred Lau, David Kim, Nicolas Villar and Hans Gellersen. 2014. MixFab: A Mixed-Reality Environment for Personal Fabrication. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14).

- *Christian Weichel*, Manfred Lau and Hans Gellersen. 2013. Enclosed: A Component-Centric Interface for Designing Prototype Enclosures. In Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13).

- *Christian Weichel*, Jason Alexander, Abhijit Karnik and Hans Gellersen. 2015. Connected Tools in Digital Design. IEEE Pervasive Computing 14, 2 (April 2015), 18-21.

- *Christian Weichel*, Jason Alexander and John Hardy. 2015. Shape Display Shader Language (SDSL): A New Programming Model for Shape Changing Displays. In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15).

- John Hardy, *Christian Weichel*, Faisal Taher, John Vidler and Jason Alexander. 2015. ShapeClip: Towards Rapid Prototyping with Shape-Changing Displays for Designers.

In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15).

- Faisal Taher, John Hardy, Abhijit Karnik, *Christian Weichel*, Yvonne Jansen, Kasper Hornbæk, and Jason Alexander. 2015. Exploring Interactions with Physically Dynamic Bar Charts. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15).

- Manfred Lau, *Christian Weichel*, and Nicolas Villar. 2013. Workshop on personal and pervasive fabrication (PerFab 2013). In Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication (UbiComp '13 Adjunct).

- Steven Houben and *Christian Weichel*. 2013. Overcoming interaction blindness through curiosity objects. In CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13).

- Thomas Kubitza, Norman Pohl, Tilman Dingler, Stefan Schneegaß, *Christian Weichel*, and Albrecht Schmidt. 2013. Ingredients for a New Wave of Ubicomp Products. IEEE Pervasive Computing 12, 3 (July 2013), 5-8.

Table of contents

List of figures

List of tables

1 Introduction

Digital Fabrication is a process that produces physical objects from digital 3D models (see Figure 1.1 a, b). This takes place by virtue of digital fabrication machines (see Figure 1.1, f), which can either accumulate material to form a physical shape, or carve an object out of a block of material. The former, called additive fabrication, is also referred to as 3D printing; the latter, subtractive fabrication, is often implemented through laser cutters and CNC milling machines. A key property of the digital fabrication process is that the quality of its physical result depends solely on the machine properties and not on users' ability to shape material. Thus, users only have to digitally describe/model the object they want to create, and no longer need know how to physically fabricate it. This is a paradigm shift, as it enables non-fabrication-experts to create complex physical artifacts.

Now that the digital description of a physical object is enough to reproduce it, we can virtually share physical objects online. Hobbyists have taken to this practice and offer a wide variety of 3D models for download[1], which can be 3D printed or laser-cut, respectively. In the commercial space, companies such as Shapeways[2] enable non-engineers to create physical objects through customization and 3D printing, in a variety of materials ranging from plastics to ceramic and metal. Domain experts can create physical artifacts based on previously acquired measurement data. For example, doctors can use digitally fabricated objects

[1]http://thingiverse.com
[2]http://shapeways.com

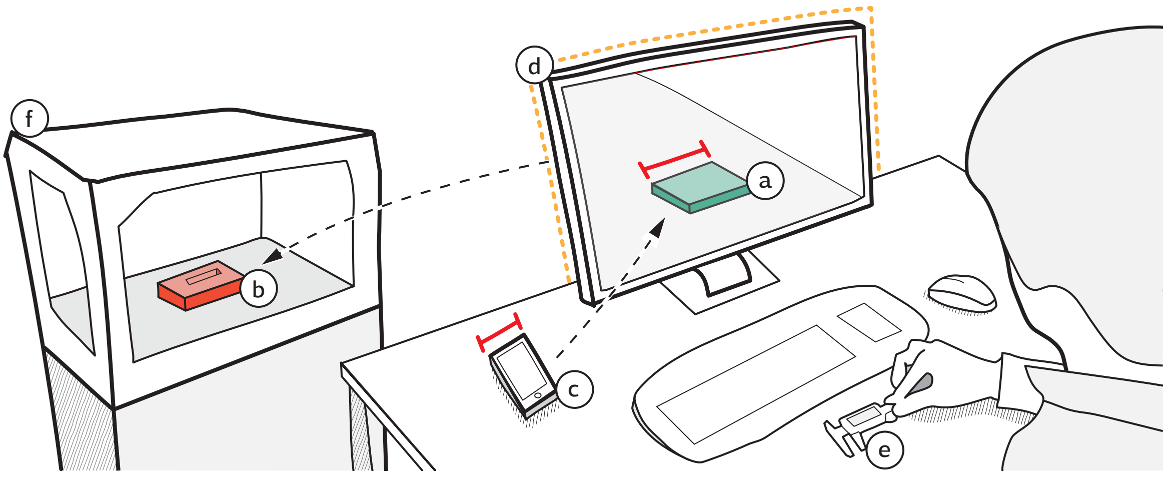


Fig. 1.1 Illustration of digital design and fabrication. (a) digital model being designed, (b) physical object that was fabricated from a digital model, (c) physical object that influenced the design of *b*, (d) the virtual design environment, (e) analog tool used to integrate the physical object into the design, (f) digital fabrication machine (3D printer).

to plan complex surgical procedures [2]. Digital fabrication is well suited for supporting device prototyping, e.g. for human-computer interaction research, as device prototypes can be fabricated from digitally designed blueprints.

Besides the benefits offered by this process, it also has drawbacks. To create the digital blueprints necessary to fabricate physical objects, we need to design in virtual environments typically bound to flat graphical user interfaces (see Figure 1.1, d). Thus the space we design in is separated from the space we design for. This has a range of disadvantages. Compared to physical space, we need to learn how to interact with, modify and manipulate objects in such virtual spaces. For example, grabbing and moving an object from A to B in physical space is easy, but not an obvious operation in a virtual 3D environment. Making informed design decisions is also difficult. Judging size and future interactions with the physical world requires experience and training. Compared to the physical space our spatial understanding of objects suffers in virtual environments [3]. Integrating existing objects or their properties (such as the width of the phone in Figure 1.1, c) is cumbersome. It requires the use of analog, disconnected tools (see Figure 1.1, e). Lastly, because we no longer interact with and shape physical material directly, we lose direct engagement with the artifact, possibly impeding on creativity.

Our thesis is that we can overcome these disadvantages by integrating the physical world closer into the virtual design environment. To this end, we develop four different concepts, each exploring a different form of such integration. Employing a design research method-

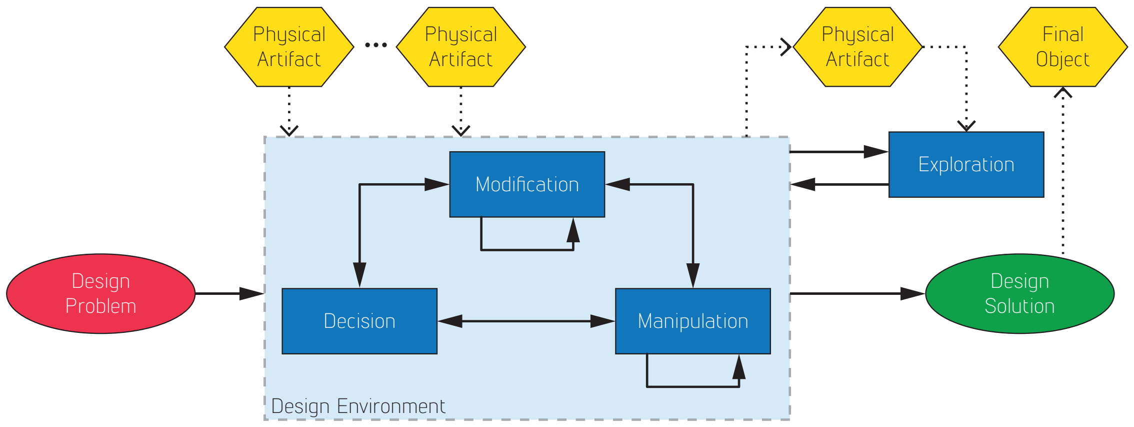


Fig. 1.2 A fabrication-aware design process model. Users, starting with a design problem, perform a various actions involving physical artifacts and the design environment, to create the final object (solution).

ology, we implement each concept prototypically, and study its benefits and properties. *Enclosed* integrates virtual models of physical reference objects into a prototype enclosure design environment. *SPATA* integrates active tangible measurement tools into common design environments, such as parametric Computer Aided Design (CAD). *MixFab* situates the design environment in a mixed-reality space, enabling novices to design objects by mixing existing ones and through gestural interaction. *ReForm* synchronizes the digital model with its physical object, so that users can perform modeling operations in physical or virtual space alike.

1.1 Problems and Aim

Currently, users design physical objects in virtual design environments. This imposes a separation between the space users design in, and the space they design for. Activities of the design process (for example, integration of physical artifacts or exploratory actions) are not well supported. More generally, this separation poses diverse issues presenting difficulties for skilled digital fabrication users and barriers to novices. In the following we first introduce a coarse *fabrication-aware design process* model, then list the problems that stem from the physical/virtual separation, and conclude with the aim of this dissertation.

The goal of design processes we are concerned with, is to create physical objects using digital fabrication. As such, these design processes (see Figure 1.2) are fabrication-aware: either because the designers themselves will make decisions while being aware of the subsequent fabrication, or because the design environment used, explicitly supports fabrication specific aspects. Throughout the processes, users perform a series of actions to bring about a solution to their design problem[3] [4]. These actions (see Figure 1.2, blue boxes) primarily involve making *decisions* about the shape of the final object and are influenced by existing physical artifacts e.g., when deciding how big to make a phone dock, the size of the phone the dock is for, will be important. To implement their decisions, users *modify* the object-under-design through operations offered by the design environment e.g., add a hole or conversely more material. To support and make their decisions, users *manipulate* objects (including the object-under-design), for example rotate or move them within the virtual environment. The outcome of these actions is scrutinized through *exploration*, where users create a temporary physical prototype of the object they are designing and place that prototype in its target environment e.g., when designing a game controller, print that game controller and place it in users hands to see if the controller is comfortable to use.

Fabrication-aware design is currently inaccessible to novices and inconvenient for experts, as design environments are situated entirely in virtual space. Design being situated in virtual space makes interaction with the physical world, the world we design for, difficult. Integrating existing objects, or evaluating the object we are currently designing in the physical realm is cumbersome. We can not directly interact with the design material or object-under-design. Further can the interaction with entirely virtual environments be more challenging than interaction with physical space. In more detail, the problems, caused by the disconnection between virtual design environment and physical target space, are:

(P1) ***Difficult interaction with virtual spaces*** As the design of objects is situated in virtual environments, we have to learn how to interact with these virtual spaces [6]. Manipulating objects, a task that is intuitive in physical space, becomes a hurdle. For example, moving an object from one place to another, is a task easily performed in the physical world. In virtual spaces, a multitude of interaction techniques exists to perform this simple task. More complex or abstract operations require new interaction concepts, many of which have to be learned by novices.

(P2) ***Spatial understanding is hampered*** In virtual environments users, particularly novices, struggle with the judgment of spatial relationships between objects [3]. In such environments, it is hard to perceive depth and the order of objects along this dimension correctly. Moreover, correctly perceiving size and distance in virtual environments

[3]Accurately modeling design processes, all their steps, components, influences and relationships is a difficult task and far beyond the scope of this thesis. We have thus opted for this simplistic model inspired by Simon [4]. Note that we do not describe a process itself, but its components relevant to this thesis. For a more comprehensive overview of design process models, please refer to Wynn and Clarkson [5].

proves difficult [7]. For fabrication-related applications that results in objects which are not sized as intended by their designers, as their size was judged incorrectly.

(P3) ***Lack of physical artifact integration*** When designing new objects, the decisions we make are often influenced by existing artifacts. For example, when designing a phone dock, we will have to consider the size and shape of the phone we are designing the dock for. More generally, throughout any fabrication-aware design process, designers have to interact with objects of the physical world to make informed design decisions or validate previous ones. Physical artifacts, such as dimensions, shapes, weight, balance, colors – materiality in general – of existing physical objects, currently lack integration into the design environments.

(P4) ***In-context exploration is limited*** Rather than taking physical objects into the design environment, we might want to take the object-under-design out of design space and into the target environment, either to validate or to make design decisions. For example, when creating interactive artifacts, e.g. a new game-controller, being able to place that game controller in someone's hands aids its ergonomic design. When designing within a virtual design environment, that is removed from the physical world, such explorations are not possible. We can not place the object in its target context, prior to fabricating it.

(P5) ***Lack of direct engagement with the material*** Crafting a new object is in part about the experience of shaping the material itself [8], and the creative opportunities that can arise from intimate engagement with the material and tools. Due to the separated spaces we lack this direct engagement with the object-under-design. After all, we can not directly manipulate and modify the object-under-design with our hands or physical tools.

(P6) ***Digital-Fabrication and engineering knowledge required*** Despite the fabrication quality being a direct result of the machines fabrication ability and not of the user's, designers still need to acquire and maintain knowledge about the fabrication technology itself. Digital fabrication machines have tolerances that need to be considered, the material they fabricate in has certain properties (e.g., minimal wall thickness) that need to be taken into account. Constructing complex assemblies requires mechanical knowledge about material strengths, different types of connectors and various mechanisms.

These problems, are grounded in literature, design practice and anecdotal evidence. That it is difficult to interact in virtual environments (P1) (independent of the application), is a long standing problem in the HCI community [6]. In literature [3, 7] we find evidence that, in such virtual environments, spatial understanding is impeded (P2). We know through anecdotal reports that integrating existing objects and their properties (P3) in design processes is important. To ground this problem further, we present evidence that such integration would be valuable, in Chapter 5. In a larger context, the role of physicality in design is well established [8, 9, 10], which motivates problems P4 and P5). P6, the need to know about fabrication properties, becomes clear when one considers how digital fabrication machines operate (see Section 2.2 for an introduction). These machines implement mechanical processes and are thus subject to tolerances and material properties.

The fabrication-aware design processes of experts are hampered by the aforementioned problems caused by a lack of integration of physical space into their tools and design environments. Novice users are presented with a significant learning curve, as they have to learn not only about fabrication technologies and their properties, but also how to interact with digital design environments. In this thesis, we target a broad spectrum of experience: from novices to experts. Our concepts are independent of the experience level, but our implementations are tailored towards inexperienced users. In the discussion, we draw out a path of how to adapt the concepts (and implementations) for varying skill levels. For example, in Chapter 6, we present the concept of bidirectional fabrication, which is beneficial for novices and experts alike. Our implementation offers a simplified user interface (UI), but as we elaborate in the chapter (see 6.3.2), could be adopted for experts.

Our aim in this thesis is to connect the physical world and digital design environments in which we create new artifacts, to alleviate the problems outlined above (P1 - P6). We integrate physical artifacts more closely design processes to increase their influence on design decisions (P3), ease spatial understanding (P2) and enable the early exploration of the interaction between the object-under-design in their target context (P4). We further situate design environments closer to the physical realm to ease interaction with them (e.g., moving an object from A to B, P1) and foster more direct engagement with materiality (P5). By doing so, we seek to lower the barriers for digital fabrication novices, enabling new user groups to participate in the creation of physical artifacts. For skilled users, we want to enrich their design experience by making it more convenient and effective.

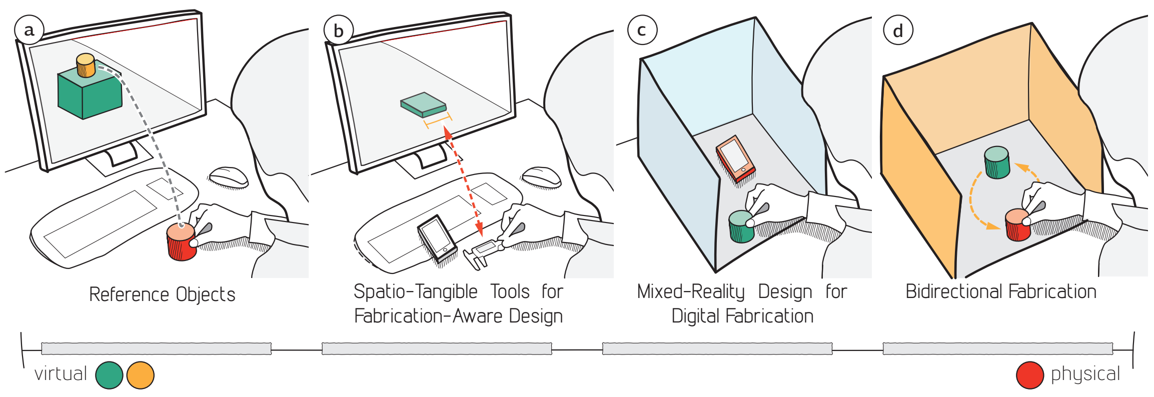

a
b
c
d
Reference Objects
Spatio-Tangible Tools for Fabrication-Aware Design
Mixed-Reality Design for Digital Fabrication
Bidirectional Fabrication
virtual
physical

Fig. 1.3 Interaction concepts along the virtuality continuum [11]. *Reference Objects* relate physical objects to digital model. *Spatio-Tangible Tools for Fabrication-Aware Design* integrates physical measurements into virtual design environments. *Mixed-Reality Design for Digital Fabrication* creates a space where virtual and physical objects co-exist. *Bidirectional Fabrication* contentiously synchronizes digital model with its physical rendition.

1.2 Concepts

Towards the aim outlined above, we develop four unique concepts, each of which targets a subset of the aforementioned problems. These four concepts are situated along the *virtuality continuum* [11] (see Figure 1.3). The design environments we create occupy a space along said continuum, and each concept employs physical features of different nature. For example, the *mixed-reality design for digital fabrication* approach addresses problems P1, P3, P5; is centered on the virtuality continuum; and integrates physical shapes. The four concepts we develop in this thesis are:

1. **Reference Objects** are objects at the designer's immediate disposal, whose virtual 3D model are used during design. As designers have the physical *reference objects* at hand, they can pick them up and compare them with the virtual models they used in their design. Through this immediate representation of known physical objects in the virtual environment, we enable designers to physically recreate the spatial arrangement they see on the screen, in an attempt to further spatial judgment (P2).

2. **Active Spatio-Tangible Measurement Tools** connect the physical and virtual world through measurement tools which seamlessly work in both worlds. Dimensions and angles, as simple as those features may be, are prominent design decisions to make when designing a new object (P3). Thus situating these decisions in physical space will enable designers to make more informed decisions, for example by measuring an

existing artifact. This concept ought to hold in reverse: reflecting upon the physical properties of the virtual 3D model enables us to judge the physical presence an object will have subsequent to its fabrication (P2, P4). By making this activity a tangible matter, we hope to foster such reflection.

3. **Mixed-Reality Design for Digital Fabrication** situates the design environment in a space where virtual and physical things coexist. In such a space entities are not bound by the laws of physics, yet can be manipulated as if they were physically present (P1). Existing objects can effortlessly be integrated into new designs as they occupy the same space as objects under design (P3). We can, for example, place a physical thing inside the virtual object being designed, in order to scale said object to enclose the physical thing. In such an environment, we hope to foster intuitive and direct engagement with object being designed (P5), while lowering barriers for novices.

4. **Bidirectional Fabrication** maintains a synchronized representation of the object-under-design in physical and virtual space. Users can modify and manipulate both as they see fit. As one is altered, the other one is synchronized. For example, if a hole is made in the physical representation, the virtual counterpart is updated so that it too has the same hole; and vice versa. The physical rendition can be scrutinized in its target context (P4), offers the intuitive manipulation and modification of physical objects (P1), and by occupying physical space can be modified with respect to existing objects (P3). The virtual model affords the benefits of digital entities e.g., it can be shared, previous versions can be restored and we conveniently apply repetitive or complex operations to it. With this concept we aim to combine the benefits of physical and virtual design environments.

1.3 Methodology

We employ a *research through design* methodology developing novel systems that by implementing the aforementioned concepts, demonstrate alleviation of the problems (P1 to P6) caused by the prior physical space/design space disconnection. We take up the roles of interaction designers and engineers [12, 13]. As interaction designers we develop novel interaction concepts; as engineers we build the technological base and implement prototype systems that illustrate/evaluate the novel interaction concepts. Our engineering work is guided by the requirements of the interaction design and underlying approach. It follows a systems-engineering approach, where we integrate (often pre-existing) components in novel ways, creating technology that is more than the sum of its parts. We implement each of our concepts in a system, showing that the idea can be made a reality. For each system (and thus concept), we show through examples and walk-throughs that it is *useful* (can produce non-primitive objects) [14], and how it alleviates physical/digital divide problems.

We evaluate our interaction design work according to Zimmerman et al. [12]. For each concept, we describe the *process* that led to the respective designs, including design choices and decision rationales. We demonstrate how these concepts (and their implementations) constitute a significant *invention*, in that they produce novel technical solutions. The *relevance* of this work is underpinned by the relevance of digital fabrication research as demonstrated by related work (see Chapter 2.3). Lastly, the *extensibility* of this work is shown by it opening future research opportunities (see Chapter 8).

Our engineering work will be evaluated through a characterization of the implementations properties, and their suitability as research vehicles for investigating the underlying interaction concepts. First, we technically characterize our solutions (e.g., in chapter 6, the speed at which the machine can operate), so that the reader can judge if our implementation is fit for its intended purpose. Second, we use the systems to evaluate the underlying concepts, thus evaluating the implementation quality itself. For example, in chapter 5 we implement a system to show – through a user-study – that its underlying concept enables novices to create meaningful fabricated artifacts.

1.4 Contributions

This thesis makes several original contributions to the field of Human-Computer Interaction (HCI). We contribute new knowledge regarding the interaction with digital fabrication systems, specifically their fabrication-aware design environments. Our work provides a broader understanding of how to design such environments, through a series of four novel systems.

All systems developed in this thesis are contributions in their own right. Through each system, we develop the interaction concept and show that it can be implemented. To this end we solve numerous technical challenges, contributing knowledge as to how to build such systems. The systems themselves serve as vehicles to investigate the four interaction design concepts they are based on. Different forms of evaluation are brought to bear, depending on the system at hand (e.g., lab studies to show that a system – hence its concept – lowers barriers for novice users). In summary, we contribute four design-for-fabrication concepts (outlined in the previous sections) and their system implementations:

1. **The concept of *reference objects* and its implementation *Enclosed*** which is an

enclosure design system for creating device prototypes using laser-cutters. We utilize the Microsoft .NET Gadgeteer landscape, and use the components that constitute the prototype as reference objects. Users can place components on an enclosure which resizes to ensure the components fit in the resulting shape. Through our UI design we ease the interaction with the design environment (P1). To free designers from fabrication-specific knowledge requirements (P6), we contribute a new algorithm for generating laser-cut enclosure outlines from a part-graph representation, including a heuristic for choosing appropriate part connectors. We evaluate this system through designing and fabricating real-world examples.

2. **The concept and implementation of *Active Spatio-Tangible Measurement Tools* (SPATA tools)** which integrate two novel active, physical measurement tools (calipers and protractor) in three commonly used design environments (based on parametric, direct and mesh-based modeling – see Section 2.3). The calipers can measure and output length, the protractor can measure and output angle, enabling in-context exploration (P4). Both devices are tightly integrated into the design environments, supporting workflows and tasks within them. They are also bi-directional in the sense that they can measure, as well as physically demonstrate their respective dimension (P2, P3). The tools can serve as tangible props for the object-under-design, easing navigation within the design environments (P1). We contribute the design of the devices, an architecture for the integration of such tools, the interaction techniques enabled through them and an evaluation through application scenarios and walkthroughs.
3. **The concept of *mixed-reality design for digital fabrication* and its implementation *MixFab*.** We construct a novel hardware system that creates a mixed-reality interaction volume, supports gesture recognition and has 3D scanning capabilities. Based on this hardware, we implement a fabrication-aware design system that users can interact with through gestures (P1, P5), and which supports the seamless integration of existing objects into new designs (P3). Next to various technical contributions, we present a user-defined gesture set that forms the basis of interaction with this system. Through a user-study, we provide evidence that this approach indeed lowers barriers for novice users to design meaningful objects utilizing digital fabrication.
4. **The concept and implementation of *bidirectional fabrication***, which continuously synchronizes a digital model and physical object. On top of this core concept, we build ReForm, a system to design objects using manual physical modification, as well as precise digital operations. We contribute the first *bidirectional fabrication* imple-

mentation, which combines additive and subtractive fabrication, a structured-light 3D scanner, a custom-built computer-numerically controlled (CNC) five-axis motion platform actuating a custom clay mill and extruder, and a purpose-built, back-projected Augmented Reality (AR) display. We make numerous technical contributions, specifically the use of two-state polymer clay for interactive fabrication systems and a novel toolpath generation algorithm to update the physical object. Lastly, we demonstrate the benefits afforded by ReForm through application examples.

1.5 Structure

This thesis is structured along the different systems we develop. It is organized as follows:

Chapter 2 *Related Work* first highlights the relationship between HCI and the physical space, as illustrated by Tangible User Interface (TUI) and prototyping toolkits. We then focus on digital fabrication, introducing fundamental production technologies and research that aims to extend what these methods can produce. Following we present prominent categories of existing virtual design environments, predominantly from the CAD community. Subsequent to this more fundamental introduction, we shift towards research in fabrication-related design environments, including sketch-based, domain-specific, tangible, and interactive fabrication UIs. Lastly, we give an overview of fabrication-related computational design work, that treats production-specific questions from an algorithmic standpoint. We conclude with a summary of this section where we embed our own contributions in the surveyed work.

Chapter 3 - 6 describe the concepts (see Section 1.2) and subsequent system implementations (see Section 1.4) as detailed above. We start by investigating the integration of physicality into purely virtual design environments. First, through *reference objects* that bundle physical dimensions in 3D models (see Chapter 3). Following this, we introduce active tangible UI components into virtual design spaces, enabling the bi-directional transfer of length and angle (see Chapter 4). After exploring virtual systems, we move along the virtuallity continuum towards a mixed-reality environment, where users can integrate existing physical objects all-together (see Chapter 5). Adding a tangible component to such environments, we then move to the physical end of the aforementioned continuum and synchronize virtual and physical space to enable design in both spaces (see Chapter 6). Each of these chapters introduces the concept, describes the design and user-interface of the system, its implementation, and a form of evaluation (predominantly through application examples). We conclude with a discussion of system-specific matters and a summary that embeds the chapters work in the thesis by relating it to the problems and approaches introduced above.

Chapter 7 *Discussion* considers aspects that bestride the individual chapters. We discuss the implications of mixed physical/virtual design environments for different user-groups, for design processes, for the resulting artifacts, and how these concepts generalize to other domains. This is followed by a reflection on the research methodology we employed. We relate our own approach to more empirical methods and broader design exploration. This leads us to a discussion of our evaluation methods. We then elaborate how our systems could be combined, and on the effect future technological developments will have on the contributions made in this thesis.

Chapter 8 Lastly, the *Conclusion* summarizes the contributions of this thesis in a systems and design process view. We relate back to the problems introduced in the introduction, and show how our work has contributed towards their mitigation. We conclude by offering hints at future avenues of fabrication-related research: combining shape-changing devices with digital fabrication, and investigating the interaction aspects of computational design.

2 Related Work

Digital Fabrication produces physical objects from digital data. Thus, it is an important question how to design these physical objects in a virtual world. Considering that this connection between physical and digital world has long played a key role in HCI research, we review prototyping and physical computing work our community produced. Then, we introduce digital fabrication processes and their various implementations; throughout this thesis we will refer to these concepts and technologies.

The question of how to design for fabrication is not new, but has seen thorough treatment in the CAD community. In this thesis we are interested in the interaction specific aspects, thus we introduce key modeling concepts the CAD community produced and that have found widespread adoption. Our own community (HCI) has also contributed a body of work that explores various aspects of fabrication-aware design systems – work that we structure along these aspects.

Lastly, we look at work that is related to fabrication in a broader sense. That includes the algorithmic treatment of 3D shapes in a fabrication context e.g., answering questions to the effect of "how can we 3D print this model?". Other work is concerned with extending the range of objects that can be produced with digital fabrication. Going away from utilitarian design contexts, digital fabrication has also been put to use for public engagement in creative activities.

2.1 HCI and Physical Computing

Prototyping new interaction devices is a core activity of Human-Computer Interaction [15, 16]. As we develop new interactions, devices, and ideas we build physical artifacts of varying fidelity to explore, evaluate and test our ideas. This practice has lead to many physical computing toolkits which aim to make building physical prototypes easier by providing basic interaction components e.g., buttons or displays. In a similar vain are construction kits: they are building blocks for constructing physical objects, often with some kinematic component, construction kits for shape input are surveyed in section 2.3.4.

It has been a long-standing goal to make interaction with computers as "natural" as interaction with the physical world is. *Tangible Computing* [17] promotes this idea clearly by associating digital data with physical proxies. Here, physical, low-fidelity objects are not only prototypes but the final interaction artifacts themselves. For example, Hinckley et al. use a rubber ball for neurosurgeons to navigate a 3D model of the brain before surgery [18]. They did not use a realistic prop, but a rubber ball that can be more comfortably held. Rotating the physical prop directly rotates the model on the screen. The Cubic Mouse [19] also lets users change the models orientation using a spatially tracked prop, but adds a button for clutching. This way, the model can be rotated to any position while the prop can still be held comfortably. With active TUIs users can not only manipulate digital artifacts through physical interaction, but physical objects can be altered programmatically. For example, the Actuated Workbench [20] is a top-projected surface on which passive, but magnetic pucks can move autonomously. Used for example to decide cell-phone tower locations, the system always maintains consistency between the projected and physical state. ZeroN [21] shows an active tangible suspended in mid-air. The commercially available Phantom Chess [22] game uses autonomous tangible chess pieces to interact with the user in two dimensions. Nowacka et al. [23] give an overview of self-actuated autonomous tangible user interfaces. Our own SPATA tools (chapter 4) build on those ideas, as they can autonomously maintain a consistent state of the virtual model and its measurement in the physical world.

Recent concepts involving computational interaction between matter and data envision an even more fluid and radical connection between both worlds, where matter can be altered programatically [24].

2.1.1 Toolkits

To ease the prototyping of TUIs and other interactive devices, hardware toolkits encapsulate interaction primitives (e.g., buttons, sliders or displays) as reusable components. *Phidgets* first introduced this idea of encapsulating interaction primitives, focusing on device prototyping as target application for such physical widgets [25]. *iStuff* [26] extended this idea by introducing a virtual patch panel, through which the data and events sent between the physical components was routed through a software-defined mechanism, thus promoting component and assembly reuse. Hartman and colleagues identify iterating with evolving prototypes as key task during design. Their *d.tools* [27] toolkit thus integrates the reusable hardware components with an integrated design and test environment. Promoting a more material approach to physical prototyping, Hudson and Mankoff [28] use cardboard, thumbtacks and capacitive touch sensing hardware to support *interactive physical sketching*.

These early, yet sophisticated, hardware toolkits have given rise to new, powerful and widespread physical computing platforms. More in the spirit of device prototyping, supported by a sophisticated integrated development environment is *.NET Gadgeteer* [29]. This platform encapsulates powerful hardware components (e.g., cameras, displays, networking and radio devices) through an object oriented library. By standardizing the connectors between components, they become easy to connect, making powerful hardware available to novices. A more fundamental approach is offered by the widespread, open-source *Arduino* platform[1]. The platform consists of various microncontroller boards and a standardized API, giving programmers access to the various functions offered by the microcontroller. Through *Arduino shields*, users can add various hardware components. Compared to other prototyping platforms however, there is no standardized set of elementary components (e.g. buttons). Users are encouraged to familiarize themselves with basic electronics, in order to use such components. A host of contributed libraries eases their integration. With Fritzing [30], an integrated development environment, that is particularly focused on the hardware design side, is available.

Creating a device enclosure is an important step during prototyping. Yet, the presented toolkits do not offer means for the fabrication-aware design of a physical enclosure. Instead, some rely on cardboard [28], others make no recommendation. The .NET Gadgeteer project [29] offers 3D models for a subset of their components, so that experienced users can use CAD software to design enclosures. In chapter 3 we present a design environment that supports users in designing laser-cut prototype enclosures.

[1]https://www.arduino.cc/

2.2 Digital Fabrication

Digital Fabrication is the fabrication of physical artifacts from digital data. Two different approaches to fabrication are subsumed under this term: additive manufacturing where material is added to form an object, and subtractive manufacturing where material is removed to the same end. Both approaches can be implemented through different processes, each offering unique benefits and disadvantages. While a full survey of these approaches and currently available machines is outside of the scope of this thesis (for an in-depth analysis please refer to Harrop and Gordon [31]), we present a short historical account and give an overview of the available processes, technologies and machines.

Gershenfeld traces the first computer-numerically controlled (CNC) fabrication machine to an MIT research system built in 1952 [32]; a milling machine that was capable of producing aviation metal parts. The first CNC machines were indeed milling machines, implementing the subtractive fabrication process. It was not until 1980 that additive fabrication machines became available. Such machines are also referred to as rapid prototyping or freeform fabrication. Compared to subtractive processes, additive manufacturing is more versatile, as it can create complex internal structures what can not be reached by a milling tool. Initially very expensive machines – in 1998 a fusion-deposition modeling 3D printer would cost in the order of 100k GBP [33] – the cost of such devices has been reduced by a factor of 100. It is this development that has spurred the *digital fabrication revolution* [32]: the availability of advanced automated manufacturing methods for the masses.

Motivated by this promise of making fabrication available to a broader user group [34, 35], we quickly identified the step before the actual fabrication as an interesting field. To create informed systems, we need to understand the various digital fabrication technologies, their abilities and trade-offs. Throughout this thesis we will make references to the technologies presented here, denoted by their common abbreviations. For ReForm (see Chapter 6), we design and build our own five-axis CNC platform, combining additive and subtractive manufacturing.

2.2.1 Additive Fabrication

Additive Fabrication (more commonly known as 3D printing) builds up objects by adding material layer by layer. Various implementations of this process exist, varying in cost, as well as spatial resolution (and thus tolerances), material and structures they can produce. Probably, the most common implementation is *Fused Deposition Modeling (FDM)*.

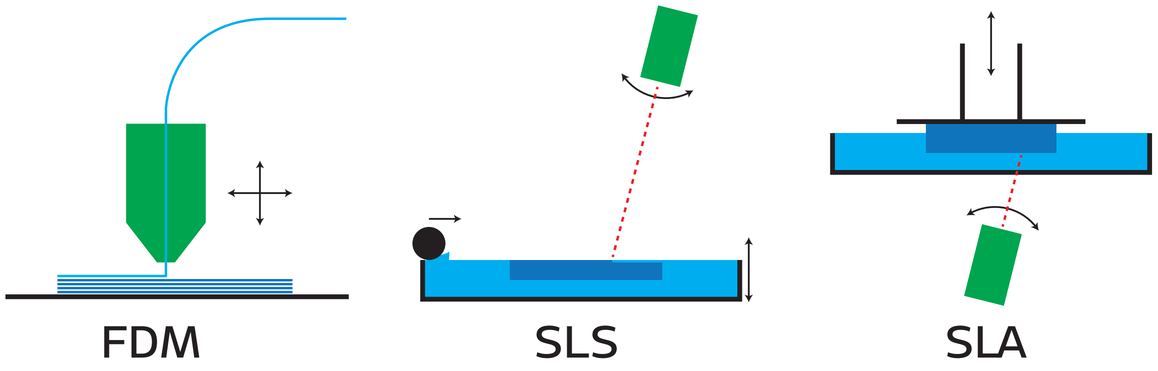


Fig. 2.1 Different additive fabrication processes. FDM is *fusion deposition modeling* where the extruder (green) deposits material (blue) onto a build platform (black). SLS is *selective laser sintering* where a laser (red) fuses a material powder (blue) in a container. A rolling bar (black) deposits a new layer of powder for each successive layer. SLA is *stereolithography* where a light source (red) hardens an ultraviolet curable resin layer by layer.

Consumer-grade devices, such as the ones produced by Makerbot [2] or Ultimaker [3] employ this process. This process deposits molten plastic – typically Acrylonitrile Butadiene Styrene (ABS) or Polylactic Acid (PLA) – extruded through a hot nozzle layer by layer, thus forming the object (Figure 2.1, FDM). Objects produced this way are characterized by a rough surface finish and low tensile strength; both caused by the layering. Yet, this process can be implemented with simple means, making it a prime candidate for widespread adoption. Notably, the RepRap project [36], produced one of the first open-source implementations of this principle. Through their efforts, 3D printing became available to new users groups and found widespread adoption in the hobbyist and open-source communities.

Selective Laser Sintering (SLS) is a similar process which produces objects from a fine powder (Figure 2.1, SLS). The powder is deposited inside a build volume that can change its height. For each layer, a new powder coating is produced by moving a bar with a small amount of powder in front of it over the volume. Depending on the material being used, a laser, heat source or binder material is applied to fuse the powder particles to a solid object. This process is very versatile with respect to the materials it supports. Common materials used include Nylon, Aluminium and other metals. However, more unusual materials have been used, for example sugar to create large-scale objects [37].

Stereolithohraphy (SLA) produces objects by hardening a resin that can be cured using ultraviolet light. Utilizing a high-powered light source, e.g. a laser, a build platform is lowered into a bath of said resin, where a laser then successively hardens the material to

[2]http://www.makerbot.com/
[3]https://ultimaker.com/

form the object (Figure 2.1, SLA). Compared to FDM and SLS, this process can achieve very low tolerances as its resolution is determined by an optical system, as compared to a mechanical one. Multiple commercial systems employing this process are available, notably the Objet™ offered by Stratsys [4]. Their implementation of this process differs slightly, in that they apply the UV curable resin layer by layer through a jet-printing process, and harden it in a second pass with a high-powered ultraviolet light source. This method enables multi-material printing, where different materials offering different properties can be mixed within the same part. This way, one can produce stiffness gradients or bespoke optical properties within the same part.

Other additive fabrication methods include *sheet lamination* and *laser cladding*. The former binds sheets of material together (by virtue of glue, heat or pressure), and cuts the topmost one to form the current layer. Mcor [5] implement this process using A4 sheets of paper. Combined with an inkjet color printer, these machines can produce full-color objects. The latter process is close selective laser sintering and predominantly used for metal objects. A fine metal powder is blown into the focal point of a laser which fuses and solidifies the powder in that point. Mid-air 3D structures can be produced on existing surfaces without the need for support structures.

2.2.2 Subtractive Fabrication

Subtractive Fabrication shapes material through stock removal: given a sheet or block of material, so much of it is removed until the desired shape has been carved out. This process can be characterized by the cutting method and the degrees of freedom with which the workpiece (or cutting tool) can be articulated. The former determines the materials that can be machined, the latter determines what shapes can be produced.

Prominent implementations of subtractive fabrication are CNC milling, laser cutting, plasma cutting, water jet cutting and vinyl cutting. Within this list, CNC milling and laser-cutting maintain special roles: both are prominently used in hobbyist and industrial environments, differing mainly in their sophistication. CNC mills are most versatile with respect to the materials (material hardness) that can be machined. Depending on their construction, they can machine anything from light wood to steel. Using different end-mills, different surface finishes can be achieved. Laser-cutters offer similar material versatility, but at higher speeds. For prototyping and in hobbyist/maker spaces, 40 - 80 Watt laser cutters are com-

[4] http://www.stratasys.com/
[5] http://mcortechnologies.com/

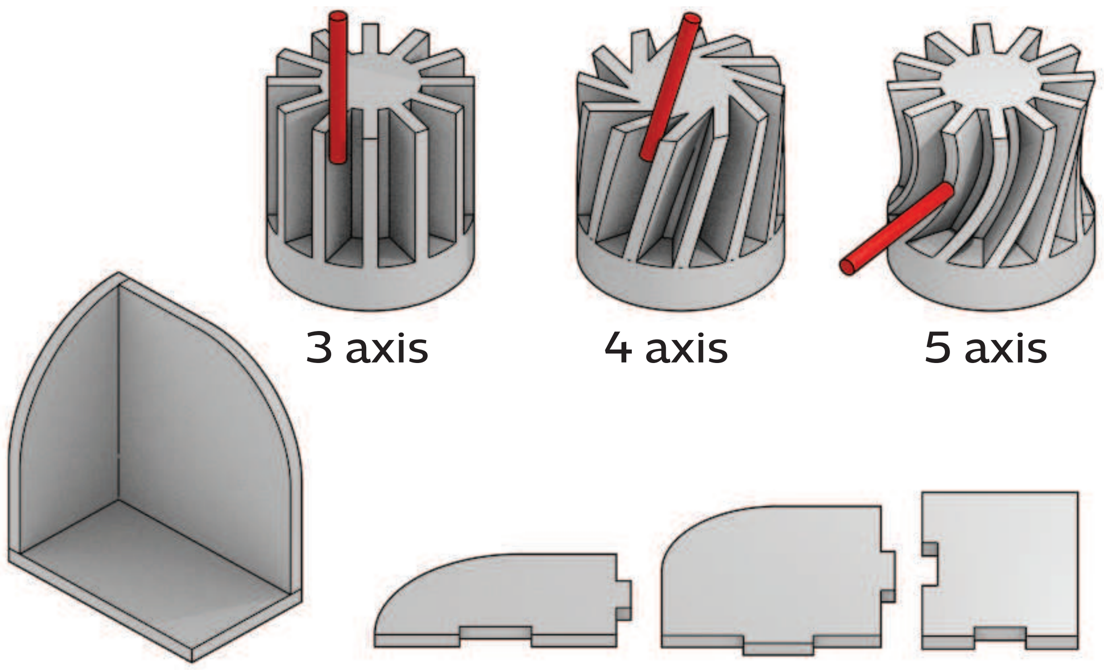


Fig. 2.2 Various objects produced using subtractive fabrication. The upper half shows three objects as they could be produced with three, four and five axis CNC milling machines, respectively. The red cylinder demonstrates tool position. Note that the five-axis object could not be produced with four axis alone; and the four axis object could not be produced with three axis, respectively. The lower half shows a 3D object assembled from three planar pieces; a construction method often used with laser-cutting.

mon; often produced by Epilog[6] or Universal[7]. Such machines can cut acrylic, paper, wood and fabrics of up to 2 cm thickness. For commercial manufacturing processes laser cutters providing multiple kilowatts of cutting energy are available, e.g. from TUMPF[8]. Such machines can very precisely and speedily cut various metals.

The amount of axis (degrees of freedom) employed to move the milling device, or the workpiece respectively, defines what shapes can be produced. Three-axis machines are mechanically simple and have thus found widespread adoption. For laser-cutters (most of which use 2.5 axis, although five-axis models can be bound) designers have to decompose three-dimensional objects into two-dimensional planar parts (Figure 2.2, lower half); the Z-axis is used solely for focusing the laser. Three-axis milling machines can produce simple 3D parts. Adding a fourth axis greatly improves the range of shapes that can be produced, as the cutting tool no longer has to be perpendicular to the XY plane of the part (Figure

[6]https://www.epiloglaser.com/
[7]http://www.ulsinc.com/
[8]http://www.trumpf-laser.com/

2.2, upper half). Employing a fifth axis further increases the range of objects that can be produced, and reduces machining time [38]. In order to produce objects with such machines, one needs to algorithmically compute the toolpath the machine needs to execute. The more axis are used, the more complex this toolpath generation problem gets. For an overview of this well-studied problem, see Dragomatz and Mann [39].

2.2.3 Advancing Fabrication

The range of what can be produced using digital fabrication is continuously extended, in part by the HCI community. In this regard our community is concerned with easing device prototyping and novel uses for digital fabrication. *LaserOrigami* makes creative use of material properties of acrylic to create 3D structures with laser-cutters. By selectively heating parts of a sheet of acrylic, the material is softened and subsequently deformed through gravity. This enables the creation of almost self-assembling structures [40]. More novel uses of laser-cutters include the preparation of food. Less for cullinary experience, more for communication researchers have engraved messages and dietary information on the food itself [41, 42]. Constructing a chocolate 3D printer (based on the FDM principle) Khot et al. produce chocolate tokens as reward for physical exercise [43].

Extending the capabilities of 3D printers Hudson presents a new 3D printer that can fabricate soft plush toys [44]. It uses a lamination process (see Section 2.2.1) where multiple sheets of felt are stacked on top of each other, and cut with with a built-in laser cutter to form the object. Using conductive fabric in-between the felt, Peng et al. [45] extend this method to produce interactive artifacts. The conductive fabric serves as bend and capacitive touch sensor and can be hidden within the object. This work enables a new class of rapidly fabricated, interactive soft toys and objects.

Fabricating Interactivity

Digitally fabricating sensors or integrating them into fabricated artifacts is primarily used for prototyping interactive devices. Recent development however, moves these technologies out of the prototyping space towards more production ready artifacts.

When prototyping interactive devices, flexibility is important as it reduces the cost of exploring different options. Our own work presented in chapter 3 is in this vain, as it frees users from having to painstakingly add finger joints and other fabrication-specific ornamentation to their design. What holds for physical shape, also holds for the interactivity enabling sensors themselves. *Midas* uses a vinyl cutter (see Section 2.2.2) to cut conductive cooper foil, to rapidly prototype capacitive touch interfaces. Extending the range of input controls, Vazquez et al. [46] introduce a series of 3D printed, pneumatic devices which can be integrated into new devices. Their pneumatic properties are not only used to sense user input, but also to provide tactile feedback. Another approach for 3D printing interactive devices is using computer vision. By embedding a single camera into the device prototype, *Sauron* [47] can sense user input. The location of the camera, and if necessary multiple mirrors, is algorithmically determined. Each interaction element (e.g., buttons or joysticks) has a unique pattern embedded which is used to sense the widgets actuation.

Moving from the prototyping space towards novel interaction capabilities, we can use multi-material 3D printing to embed optical elements in very confined spaces. By virtue of integrated, and thus 3D printed optics (so called *light pipes*), information from elsewhere located displays is guided to a desired location [48]. This way we can create interactive objects that do not require their own display, or when space is too constraint to add such. Brockmeyer et al. extend this method to printing curved displays, and add optical touch sensing capabilities [49]. A similar concept, albeit not optical, is to embed *A Series of Tubes* [50] in 3D printed objects. Filling these conduits with conductive ink, electroluminescent wire or using them as pneumatic components, enables the fabrication of interactive artifacts.

Hybrid Fabrication

The fabrication process itself can become interactive. Zoran [51] introduces *Hybrid Fabrication* to describe a process where human and machine interact to create a unique artifact. Similar to our own motivation, he bemoans the loss of engagement with the object being created. To this end he introduces a fabrication system where the user, under active guidance of the system, carves a 3D model out of a block of material [10]. At any point can users decide to overrule the guidance, thus to alter the model and object. Rivers et al. present a similar approach for clay sculpting, but uses projection mapping (rather than an instrumented active milling tool) to guide users in removing and adding material [52].

Following a more utilitarian motivation, users can be made part of the fabrication process to extend the range (or size) of objects that can be fabricated. Large CNC milling machines are expensive, and thus inaccessible to many users. By actuating a desktop routers milling bit and guiding users along a prescribed path, we can utilize human dexterity and reach to fabricate larger parts [53]. This raises the question of agency within the fabrication process: how does it feel to become part of the fabrication machinery? Devendorf and Ryokai explore this question in a series of experiements where they have study participants execute machine instructions for 3D printers. Their findings suggest that fabricating objects jointly with a

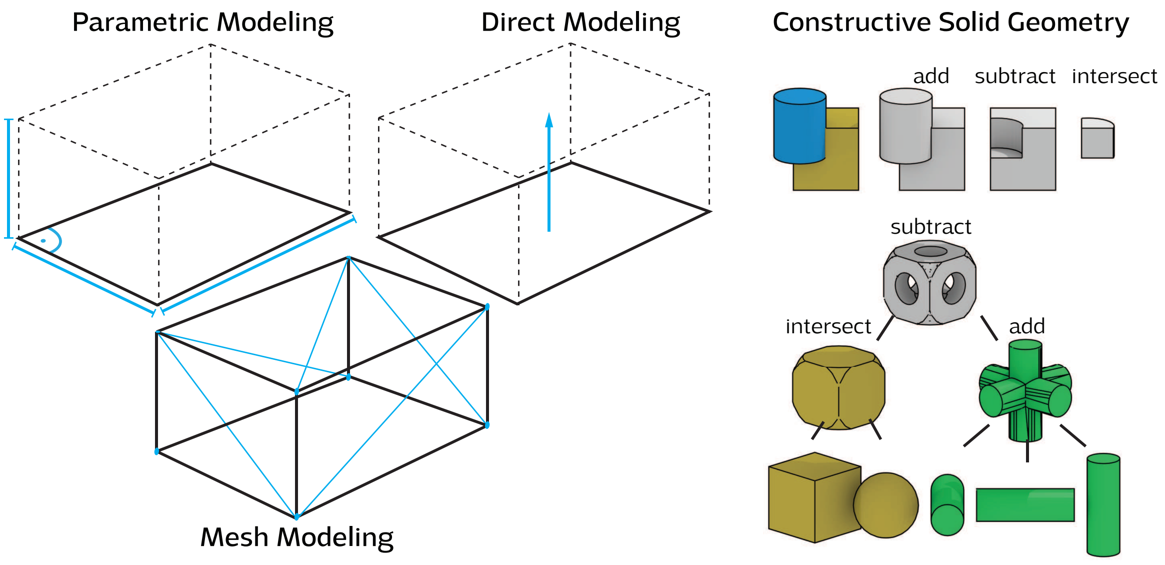


Fig. 2.3 Four different 3D modeling techniques. *Parametric Modeling* describes objects trough dimensions and other parametric constraints (blue). With *Direct Modeling* users can directly manipulate a models features - for example drag a rectangle into 3D space to form a cuboid. *Mesh Modeling* directly manipulates a graph-based datastructure representing 3D shapes. *Constructive Solid Geometry* combines primitive objects using Boolean operations.

machine can foster meditation and reflection upon the artifact being fabricated.

2.3 Design for Digital Fabrication

Interactive design systems, techniques and modeling metaphors have been developed in the CAD, HCI and graphics communities. In this section, we give an overview of these techniques and their relation to digital fabrication. After briefly introducing key CAD concepts, we review sketch-based design systems, domain specific tools that reduce the design space to ease the design process. We continue with design environments that are situated in physical space i.e. tangible design environments and interactive fabrication systems. Continued with the computational capture of physical properties we move on to algorithms that compute design aspects (e.g., if 3D model can be fabricated). We finishing off with fabrication process aspects.

2.3.1 Computer-Aided Design

Computer-Aided Design research has produced various data models for representing solid 3D shapes, and modeling metaphors to manipulate these data structures; for an overview see

Requicha and Rossignac [54]. While these advances are not digital fabrication specific, they have been developed for engineering and product design applications. From an interaction standpoint the developed modeling techniques can roughly be classified into four categories:

Parametric Modeling describes form and shape by constraining it with a number of parameters. For example, to model a cube one would draw a rectangle, fix its width and length, and extrude it in 3D constraining the extrusion height (Figure 2.3). Features created this way are often semantic in nature (e.g., this extrusion is the display mount). Thereby a key concept is to capture users *design intent* accurately [55]. This assumes that users have a fixed design intent already, making this method less suitable for early stage design exploration. This method is also called historic modeling, as each subsequent modification depends on the previous one. Prominent parametric modeling applications are *CATIA*[9], *SolidWorks*[10] and *Autodesk Inventor*[11].

Direct Modeling alleviates users from the need of a fixed design intent. Users directly modify the geometry by dragging it in 3D space or through precise numerical input (e.g., to move a rectangle by 10 mm along the Z axis, see Figure 2.3). The underlying representation of the solid geometry is the similar as with parametric modelers, so that data easily can be exchanged between the two. Some applications combine parametric and direct modeling, e.g. *SolidWorks*. Other commercial offerings include *PTC Creo Elements*[12].

Mesh Modeling modifies the underlying datastructure directly, but compared to direct modeling without semantic description. Mesh-models represent 3D form as a graph of vertices, edges and faces which are modified directly (and are technically speaking a surface representation, not solid). Graph elements are not semantically annotated (e.g., part of the model marked as mounting hole). Due to its semantic freedom, this is a very versatile modeling technique and often used for more organic shapes. However, it can also make certain operations more difficult (e.g., creating a hole) and unlike the other techniques can result in non-watertight models, for example through unconnected edges. Non-watertight (technically referred to as non 2-manifold) meshes can not be fabricated as they do not unambiguously define a solid object. Note that for 3D models to be fabricated they typically have to be converted to mesh models, as com-

[9]http://www.3ds.com/products-services/catia/
[10]http://www.solidworks.co.uk/
[11]http://www.autodesk.co.uk/products/inventor/overview
[12]http://www.ptc.com/product/creo/elements

mon exchange formats are mesh-based. Example mesh modeling tools are *Blender*[13] and MeshMixer[14].

Constructive Solid Geometry describes shapes by combining geometric primitives using Boolean operations. Primitives are objects of simple shapes such as cuboids, cylinders, spheres or cones. The Boolean operations are Union (adding two objects), Difference (subtracting one object from the other), Intersection (part shared by both objects); see Figure 2.3). It is up to the user to decompose the shape they want to model into primitives and operations; what might seem like a daunting task at first, presents itself as a problem that is easily understood by users (see Chapter 5). The primitives and operations that make up a shape can be described textually (e.g., *OpenSCAD*[15]), or be part of other modeling paradigms e.g., *SolidWorks*.

2.3.2 Sketch-based interfaces

In contrast to precisely capturing a designers intent (as aspired to by CAD systems), sketch-based user interfaces aim to lower the barriers for novice users – a goal that we share in this thesis. Sketching is an intuitive form of describing shape. Starting from hasty freehand sketches to supporting ideation of new designs, sketches can turn into drawings created with care and precision. In the context of 3D modeling, sketching is used create 3D models in the first place, add detail to models or to deform them. In this section we are going to survey specifically fabrication-related sketching work. For a broader overview, please refer to Olsen et al. [56].

The freedom offered by sketching can be utilized for explorative design, that is driven by curiosity and playfulness rather than purposeful problem solving. Embodying the sketching process through full arm movement, *Spatial Sketch* [57] encourages users to explore shape by drawing in mid-air. The system turns the so drawn shapes into fabricable artifacts (e.g., lamp-shades) by producing planar parts on a laser-cutter. Supporting novices in goal-driven design, *Sketch it, make it* is a system for sketching laser-cut parts [58]. It utilizes sketch recognition to straiten lines and infer constraints such as perpendicularlity between lines or concentricity between arcs and circles. This way users can use imprecise hand-drawn input, and the system will produce fabricable outlines.

Hildebrand and Alexa [59] use sketching to design new objects using existing 3D models. Users draw a sketch based on which a 3D model is retrieved from a database. Using

[13]http://www.blender.org/
[14]http://www.meshmixer.com/
[15]http://www.openscad.org/

further sketches detail and features can be added to the model. This frees users from having to design the whole object; instead they choose a starting point and amend it to their liking.

2.3.3 Tools that Reduce the Design Space

We have to observe many constraints when we design things. Constraints can be imposed by the fabrication technology we use (e.g., laser-cutters produce planar parts only), by the target application/domain we are designing for (e.g., a chair must not collapse under the weight of its user) and the physical world (e.g., our object will have to withstand gravity). By reducing the design space within the design environment already, we can make the design easier for novice users as they themselves will not have to be aware of all the constraints in existence. Gross describes this concept as *Code as the carrier for design expertise* [60].

Domain-specific systems will encode design knowledge about that domain, and sometimes offer simulation facilities specific to that domain. *SketchChair* lets users design chairs by drawing their shape and test them by seating virtual manikins [61]. Through physics simulation users can then see if their chair tips over. Using such physics simulations in virtual environments, users can design stuffed animals [62], other inflatable structures [63] and even complex mechanical arrangements [64]. Generalizing the concept of domain expert knowledge, Schulz et al. extract parts and connections from existing models and use this domain specific knowledge to enable novices to design complex physical arrangements such as furniture [65].

Rather than integrating domain specific aspects into the design process, one can also focus on material specific properties. Considering that objects will be fabricated in one or more materials, that have certain properties, we can ease the design process by optimizing the design operations to what can be fabricated in a given material. Saul et al. support novice users in designing interactive paper devices, through solving paper fabrication constraints using evolutionary algoritms [66]. *FlatFitFab* [67] follows a similar optimization approach for generic planar materials, e.g. acrylic or plywood. Expanding to more flexible materials, Garg et al. present a system to design and fabricate 3D shapes from wire-mesh materials [67]; a material that exhibits globally coupled deformation behavior and is thus unintuitive and difficult to work with.

In chapter 3 we present *Enclosed*, which also falls in this category. While it primarily investigates the *reference object* strategy (see Section 1.3), we also reduce available design choices to what can be fabricated. Users place objects on planar surfaces and our system ensures fabricability by computing the laser-cut outlines.

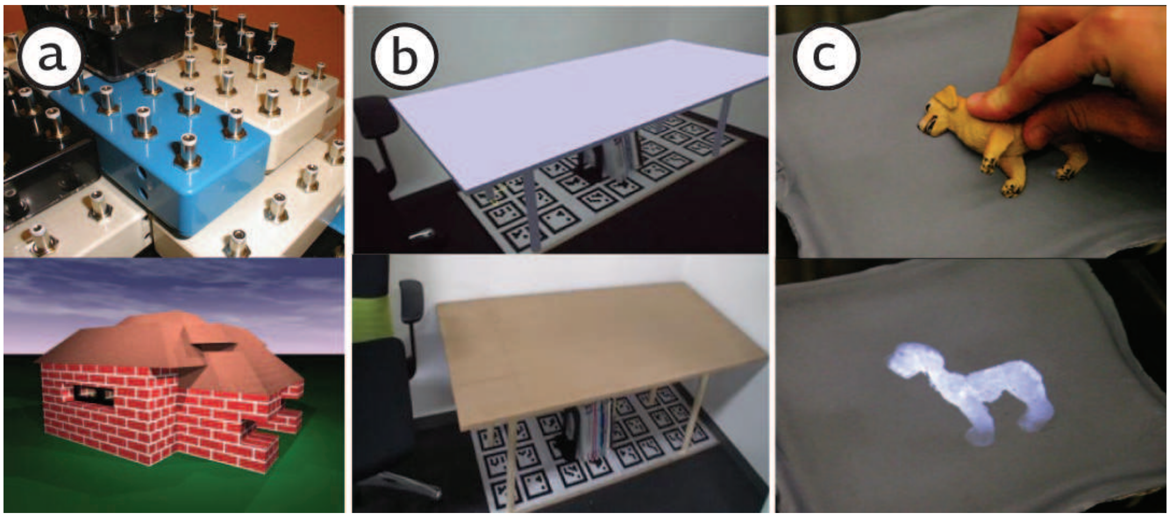


Fig. 2.4 Various tangible design environments. (a) uses LEGO™-like building blocks [73], (b) uses primitive geometric shapes stamped in place [72], (c) combines the 2.5D shape and texture of existing objects [74].

2.3.4 Tangible Design Environments

Tangible design environments for shapes and objects can be separated by their physical material: on one hand are discrete building blocks which can be connected to form shape, on the other hand are continuous materials e.g., clay [68]. Among the first building blocks (also called construction kits) were LEGO™ like objects that can sense the physical shape of their assembly (Figure 2.4, a). This idea has been extended to different shapes [69] and more flexible, hinged assemblies [70]. Rather than using building blocks, tangible tools can also be used to manipulate 3D shape. Schkolne et al. combine hand-motion with physical tools in an augmented-reality environment for users to create large-scale objects [71]. By sweeping their hand within an interaction volume, new surfaces are created; using physical tools, these surfaces are then modified and assembled. To make such an environment more precise, Lau et al. do away with hand gestures. *Situated Modeling* [72] uses geometric primitives which are stamped in place to create shapes. Unlike former systems, here users are not bound to a fixed location but can model situated in the target environment. For example, users can create a table fitting a corner by virtually placing the appropriate cylinders and boxes where the table will be (Figure 2.4, b).

Similar to the construction kit concept is the idea of creating new designs based on existing objects. CopyCAD [75] enables users to copy 2D shapes, manipulate and fabricate them inside a computer-numerically controlled (CNC) milling machine. Extending this concept to 2.5D shapes, *KidCAD* [74] enables children to "‘remix"’ their toys and design new ones (Figure 2.4, c). Unlike many other systems presented here, *KidCAD* can also capture the texture of objects, not just their shape.

Clay as a continuous medium has been used for shape-design; either as material itself, or as interaction metaphor. When modeling with clay directly, one needs to capture the modeled shape. This can be done either through external sensors (not part of the clay), i.e. 3D scanners [68] or by embedding sensors into the material itself. Reed et al. add wireless position tracker modules into clay, and recover an approximation of the clay shape from that data [76]. Clay-modeling can also be used as a metaphor, so that one does not have to solve the sensing problem. Different aspects of clay-modeling can serve as interaction metaphor: the clay-material manipulation itself, or established clay shaping tools. Using a foam-block as clay-material proxy, Sheng et al. [77] turn pinch, pull and shear interactions on this proxy into modeling operations. Instead of manipulating a tangible clay proxy, *Turn* uses the pottery wheel as metaphor [78]. Users can turn the physical pottery wheel and use mid-air gestures to shape the virtual lump of clay they see on a display situated in front of the wheel. This form of interaction is not yet generalized beyond designing pottery, yet could serve as more generic 3D modeling paradigm.

There are other continuous tangible materials that can be used for the design of physical objects: i.e., our skin when designing body-worn objects. Leveraging the elasticity of this material, *Tactum* uses shear and twist operations to design shapes that fit the human body. These shapes are then 3D printed [79].

It is important to note that the aforementioned systems have no active tactile output capabilities; the tactility they offer is purely passive in nature. As such, the integration of the physical world is a one-way street. We can manipulate physical objects, but do not get feedback on the same channel. This is very unlike manipulating physical material, which can collapse or deform. We explore the effects of providing tangible feedback in chapter 4 through active tangible measurement tools.

2.3.5 Integrating Fabrication

Interactive fabrication combines the act of designing an object with its fabrication. In this sense, it identifies digital fabrication with more traditional handicraft processes. The entire design process is situated in physical space, and every design decision made is immediately fabricated. Interactive design environments produce the fabricated artifact itself, as well as digital model of that artifact. The concept was first proposed by Willis et al. [80] and demonstrated through three example implementations in an artistic context. Shaper uses a transparent touch-screen situated above a CNC foam depositioning head. When users touch a point on the touch-screen, foam is deposited underneath that touch-point. Cutter is a subtractive system where a hot wire is used to cut a foam block. The third system, Speaker, uses sound as an ephemeral input modality. As users speak into it a wire is bent by the machine to match the waveform of the spoken sound.

This concept of interactive fabrication has been applied to other contexts, particularly by making it more precise. *Interactive construction* [81] uses a laser-cutter as drawing table. Users draw with laser-pointers directly onto the planar material. There are different "pens" for different functions, e.g. to create finger joints, or precise holes. Users can also copy the 2D outline of objects by placing them inside the laser-cutter, similar to CopyCAD [75]. Although not explicitly framed as interactive fabrication, *ModelCraft* enables users to manipulate 3D models by annotating folded paper models using a pen. Upon manipulation the paper model is reproduced [82]. Both systems target engineering applications (mechanical systems and architecture, respectively).

Going away from drawing on planar materials, *D-Coil* uses a wax extruder and removal tool for 3D modeling. The extruder is actuated and aids in the exact creation of form by compensating for hand jitter. Both tools are tracked in 3D space, so that their modifications can be recorded and a 3D model be created accordingly [83].

As interactive fabrication systems are situated entirely in physical space, they yield intuitive interfaces and direct engagement with the material. However, we must also give up some of the benefits of the virtual world, e.g. undo. Once a piece of material has been cut, or some wax has been removed, it is difficult to repair that cut or add the material back. Also, we are bound by the laws of physics. Objects can not be suspended in mid-air, and must at any point be self-sustaining (or be supported externally).

Rather than designing in physical space entirely, *Low-Fi Fabrication* aims to reduce fabrication times to a level so that numerous protoypes can be fabricated. This way, digital fabrication is not at the end of the design process, but enables fast iteration. Low-Fi Fabrication makes the trade-off between level of detail and fabrication time explicit. Depending on the stage in the design process, users can produce the appropriate physical manifestation of their model. Two approaches can be identified: part substitution and fabrication changes. The part substitution does not fabricate the whole object using 3D printing, but instead replaces parts with building blocks [84] or faster-to-produce laser-cut parts [85]. By changing what gets fabricated in full detail, we can also save fabrication time. *WirePrint* [86] produces functional aspects of a 3D model in full resolution, and non-functional aspects as wire-frame model which requires less material and thus fabrication time. In a similar vain, Koo and colleagues [87] fabricate low-fidelity prototypes that maintain their mechanical workings (e.g., hinges), but are greatly simplified in shape.

The *bi-directional fabrication* concept introduced in this thesis (section 1.3 and chapter 6) explores this space further. We integrate fabrication into the design process in order to synchronize a physical representation of the digital model under design. Compared to interactive fabrication however, the objects produced in bi-directional fabrication are not the end product, but merely an intermediary.

2.3.6 3D Scanning / Capturing Physical Properties

Capturing an existing objects properties i.e., its shape is another way to integrate physicality into the design process [88]. Various technologies exist to capture 3D shape, see Blais [89] for an overview. The most common method of capturing shape is a structured light approach. A known light pattern is projected into the environment, and captured by one or multiple cameras. Based on the distortion or other factors of the recorded light pattern, we can reconstruct depth information. To capture the shape of an object entirely, we need to see the object from different viewpoints and integrate the depth information captured for each of these viewpoints – Taubin et al. provide an overview of the available methods [90]. One particularly noteworthy implementation of this principle is *Kinect Fusion* [91] which can capture 3D geometry using consumer depth sensors (i.e., Microsoft Kinect™) in real-time.

Passive methods also exist. To reconstruct 3D geometry from passive sensors, multiple such sensors or different views on the same scene are used [92]. Other than trying to reconstruct 3D geometry, one can also use photos to inform the design process. Lau et al. use a single photo in a sketch-based design environment to fabricate objects from planar materials (e.g. through laser-cutting). Users need to know one dimension within the photo to establish the correct scale and can then trace the outlines of the object they wish to model. By virtue of fabrication constraints, the system constructs a fabricable object [93]. We can also use a photo to look up a corresponding 3D model from a database, and produce geometric variations of that model, in order to design a new object [94].

There are other physical features that we might wish to incorporate, for example appearance or behavior. The former, appearance acquisition, representation and reproduction is a heavily studied problem [95]. The latter, material and object behavior has been approached generically [96] as well as in a more fabrication specific manner. We can infer the deformation behavior of an object, and reproduce that behavior in our design, by deforming an object and capturing these deformation states [97]. Similar ideas exist for mechanical objects. Taking a motion-captured sequence of a human actor, we can automatically design mechanical linkage-based objects that describe the previously recorded motion [98].

Capturing human features is a common application of 3D scanning, and specialized systems exist. Focusing on human hair, Echevarria et al. [99] accurately reproduce hairstyles from 3D captured data using full-color 3D printers. At the expense of detail, we can capture full-body self-portraits resulting in water-tight meshes, ready for 3D printing. Li et al. account for users moving between the different camera view captures and signal-to-noise ratios of consumer-grade depth cameras, to enable such full body 3D portrait creation [100].

2.3.7 Computational Design

Physically fabricating a 3D shape, or creating a fabricable artifact from such, can be an afterthought. For example, one might want to physically produce a 3D model downloaded from one of the many online 3D model repositories available. Many of which do not cater to digital fabrication requirements and thus such a 3D model will likely not be fit for fabrication: details such as connectors will be missing, 3D meshes might not be watertight or have structures/geometry that are hard to produce using digital fabrication machines (e.g. overhangs when 3D printing). Even designs that are intended to be digitally fabricated, can have weakness that prevent them becoming physical artifacts: for example, structural weaknesses that will not survive the “green state”[16] of some 3D printing processes. Thus there is a need to determine if a 3D model can actually be physically fabricated. Should this not be the case, we will want to find a way to make such a 3D model fabricable in the first place, given a specific fabrication technology. Additionally, we might want to add specific physical features to objects: e.g., make a horse model balance on one leg.

Assessing and Computing Fabricability

Whether a 3D model can be fabricated depends on the fabrication technology, the material it is going to be fabricated in, and the geometry of the model itself. Given the first two, a major concern regarding fabricability is whether the object will mechanically sound, that is if it will withstand the forces imparted upon in in the real world. Such mechanical soundness also has to persist during the fabrication process itself. Many powder-based 3D printing processes, for example, produce very fragile and brittle objects at first, which are then reinforced. Objects then also have to sustain this brittle state of the fabrication process. Zhou et al. solve a constraint optimization problem to compute the worst-case structural analysis of

[16]Primarily sintering-based additive fabrication processes require a final curing step to finalize the fabricated object e.g., in a curing oven. In the intermediary *green state*, until the object is cured, objects are fragile and brittle.

a 3D shape, given the fabrication material properties [101]. Telea and Jalba [102] focus on a solely geometric treatment of the same problem.

Now that we know if we can fabricate an object, we will want to modify the 3D shape in case it currently would not survive fabrication or subsequent handling. *Stress relief* [103] extends the 3D printability assessment methods and can modify shapes so that they become fabricable. First, the system computes a stress analysis that takes gravity forces and imposed loads into account. Then, computes various printability optimizations, including locally thickening thing structures or adding struts where needed.

To 3D print a structurally sound object, one does not have to fill the object entirely with material. By computing good internal structures, we can optimize the material-to-weight ratio while maintaining structural and mechanical soundness. Prominent open source 3D printing applications (i.e., Makerware[17] based on Skeinforge[18], or Slic3r[19]) fill objects with hexagonal patterns. The density of these patterns (thus material to strength ratio) has to be manually set by the user. More advanced internal structures are based on surface attached skeletons, a so-called skin-frame structure inside the otherwise hollow object. This prevents having to fill the object, while still maintaining a level of rigidity [104]. *Build-to-last* reverses this concept by computing a honeycomb structure inside the model, optimizing the strength-to-weight ratio of the model [105].

Varying structures are not only required inside an object, but sometimes for external support also. FDM and Stereolithography, require external structures to support mid-air features of 3D models. Commonly such structures are created by filling the the volume underneath supported features with material e.g., as implemented by the aforementioned open source applications. Recently proposed methods create more efficient support structures using bridge scaffolding structures [106]. This reduces fabrication time and material use, while making support structures easier to remove.

Another constraint (next to need for internal and external structures) is the often limited build volume of fabrication systems. 3D printers can produce objects within a certain volume and level of detail, laser-cutters can handle material sheets of certain sizes. If an object is larger than these volumes, it has to be segmented into smaller parts which fit the available build volume. Hu et al. provide a generic treatment of the problem of decomposing 3D shapes into pyramidal shapes, which are well suited for 3D printing [107]. Such pyramidal shapes can save material and fabrication time, compared to the unsegmented counterpart. *Chopper* treats the segmentation problem in a more applied fashion by taking printabil-

[17]http://www.makerbot.com/desktop
[18]http://reprap.org/wiki/Skeinforge
[19]http://slic3r.org/

ity, assemblability, connector feasability and structural soundness into account [108]. The resulting segments are guaranteed to fit inside a predefined volume, while resulting in a desirable object upon assembly.

An edge case for making a 3D model fabricable is to produce a fabricable representation of it (as compared to producing supporting features such as structures). Given a 3D shape, we might want to produce an object that resembles this shape depending on fabrication technology, material or target application. A prominent material this concept is applied to is paper. *crdbrd* produces a set of fabricable, planar slices which once put together resemble the input geometry [109]. Slices produced by this algorithm are guaranteed to be assemblable through a sequence determined by the algorithm. A more domain specific example of such algorithms are *Popup* which produces foldable popup paper architecture artifacts [110], making burr puzzles [111], creating objects that can be folded into boxes [112] or producing surfaces that mimic 3D models within a constraint 3D volume [113]. Working with complex real-world objects, Lau et al. convert 3D furniture models into fabricable parts and connectors. Using formal grammars describing the structure of different types of furniture (e.g., a grammar for tables, one for cupboards) the algorithm analyzes the 3D model for elementary shapes (e.g., boxes and cylinders) and creates the final set of parts [114].

Adding Physical Features

Rather than enabling a 3D model to become physical reality in the first place, we might want it to have specific physical properties once it is part of the physical world. Often such desired properties relate to the material or material distribution of an object, and are thus a matter of correctly specifying such a material distribution prior to fabrication. Mechanical properties, such as balance or specific movements are also prominent.

Material distributions can be specified generically or in a goal-driven manner. *OpenFab* is a graphics-like programmable shader pipeline for specifying materials - "shading" an object with different materials [115]. This pipeline abstraction enables users to share material specifications, as they are fabrication technology independent and separated into small, independent aspects. Following a more goal-driven approach, Chen et al. propose a reducer-tuner model that lets users specify object properties directly (e.g., a certain shadow or caustics behavior) and then reduce the material configuration search space through a custom reducer-tree configuration [116]. This frees users from having to specify the exact material configuration in order to achieve a specific physical property. Bickel et al. follow a similar goal and enable users to design and fabricate materials with desired deformation behavior [117]. Using an optical system, they can measure the deformation behavior of existing materials and store them in a database. To fabricate a new deformable material, the authors use a this database to simulate new material behaviors and embed micro-structures in a multi-material print in order to create a desired behavior.

Fabricating deformable characters, e.g. action figures or collectors items, is a common domain specific use-case of multi-material fabrication and complex 3D printed mechanical arrangements. Skouras et al. present an algorithm which, given multiple deformed poses of the same 3D shape, can compute a material distribution so that the resulting object can assume the previously defined poses [118]. This method can be used to create flexible characters or even soft robotics. Following a more mechanical approach, Bacher et al. estimate articulated joints from skinned (textured) 3D meshes, to compute mechanical structures that enable the articulation of limbs as suggested by the input texture [119]. Taking this further, we can automatically add mechanical structures based on different gears and linkages, that ensure the limbs move along a prescribed path [120]. This way users can add complex mechanical features to objects without needing to have deep mechanical knowledge.

Static features, such as making an object balance on a specific point are difficult to implement manually, as they require the exploration of a large solution space. *Make it stand* solves this problem for 3D printed artifacts by creating cavities inside the otherwise solid filled object. This shifts the objects center of gravity. If a desired balance point can not be achieved that way, material is added on the outside while staying true to the models initial shape [121]. A very similar approach can be applied to optimize 3D printed objects moment of inertia, so that it spins well around a predefined axis [122]. By modifying the internal structure of a 3D printed object, we can also embed identifying information that can be recovered through imaging in the terahertz region [123]. All these examples enable users to add static, yet interactive physical properties to 3D shapes, thus physical objects.

2.3.8 Process and Social Aspects

There are various aspects that support the digital fabrication process, but do not immediately affect the design of fabricable artifacts. E.g., when using a laser-cutter, we need to place sheet material inside the machine out of which the parts will be cut; an action necessary to fabricate but unrelated to design. Before we then cut the parts, we need to arrange them on this material sheet so that we efficiently use the available material – possibly working around previously cut parts. *VisiCut* places a camera above the laser-cutter and enables users to place new parts on material sheets using the live camera feed [124]. *PacCAM* extends this idea, supporting part placement with a specialized multi-touch user interface [125].

Digital Fabrication has found use in non-goal driven, social environments. *ShadowGram* [126, 127] is a case study for *social fabrication* in public spaces. Museum visitors could capture their body silhouette, laser-cut the outline to produce a sticker, and add messages reflecting on the art exhibition this installation was a part of. A key point here was to use digital fabrication as a *curiosity object* [128], to engage the museum visitors in creative activities. Using digital fabrication to introduce a diverse audience to imaginative activities had been previously explored by Posh and colleagues [129], who situate a FabLab [130] in a museum. Nissen and Bowers [131] explore the use of digital fabrication to foster reflection and engagement in participatory design settings. They argue that by interactively designing fabricable artifacts which embody data (e.g. Twitter conversations), these artifacts become meaningful and encourage reflection upon the data they embody.

2.4 Summary

In this chapter, we have reviewed a broad range of related HCI, CAD and Computer Graphics work, all of which is concerned with digital fabrication and fabrication-aware design processes that precede production itself. We first demonstrated the relationship between HCI and the physical world. Following this we introduced fundamental digital fabrication concepts, and surveyed research that expands the capabilities of those production methods. Throughout the thesis we refer to the technologies introduced here (e.g., additive methods such as FDM or subtractive ones such as laser-cutting), and develop new fabrication technology (see Chapter 6).

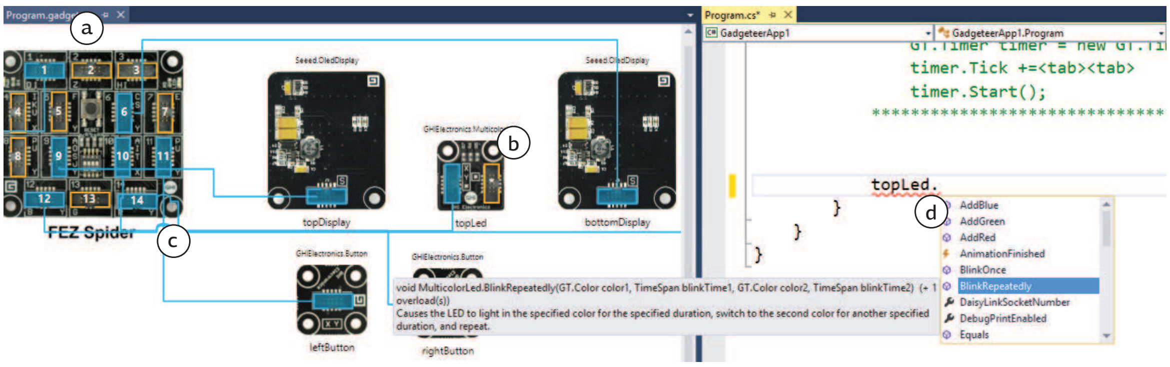
Following this introduction to digital fabrication, we presented work that is concerned with the design process that precedes production. Against the background of existing CAD techniques (which we integrate in Chapter 4), we survey other UI approaches supporting fabrication-aware design. *Tools that Reduce the Design Space* (see Section 2.3.3) are a class to which we contribute a new system in chapter 3. We also contribute a TUI in Chapter 4, thus we surveyed fabrication-related TUIs. Interactive Fabrication, or more generally fabrication integrated into the design process (see Section 2.3.5), is an important concept stemming from HCI research. We contribute a system to that category in Chapter 6. Throughout this thesis, we are concerned with integrating physical properties into digital design processes; thus we have to capture those features. Section 2.3.6 gave an overview of available ideas and technologies (which we make use of in Chapter 5). Lastly, computational design methods (see Section 2.3.7) are a way to encapsulate design and engineering knowledge to free users from having to learn it (P6) – we implement such an algorithm in Chapter 3.

3

Enclosed
Reference Objects in Domain-Specific Design Environments

Entirely virtual design environments, currently used for digital fabrication (see Section 2.3.1), would benefit from target-domain specific integration with the physical world. Such entirely virtual design systems integrate well with other development activities (e.g., device prototype development). Therefore, we explore the integration of the domain-specific physical objects involved in those development activities into virtual space. Concretely, this chapter investigates such integration in the domain of designing encasings for prototype devices using hardware toolkits (see Section 2.1.1). Currently, component toolkits offer a wide variety of hardware modules, and are accompanied by designated integrated development environments (IDEs) to write the required software. However, enclosure design is left to generic, virtual design environments (i.e., CAD).

Leaving the design of a prototype enclosures to general purpose CAD tools creates several issues. The parts that need to be enclosed are not explicitly integrated into the design process. Sometimes the 3D models of hardware components are available, otherwise their dimensions will have to be measured by hand (P3). Another problem is that users who build interactive device prototypes, while experts in their own domain, now have to become "digital fabrication experts", too. For example, HCI researchers, makers or occupational therapists [132], all of who design enclosures have to learn how to use powerful, yet complex CAD programs (P1). Additionally, they have to understand the particularities of the fabrication technology they are about to use e.g., when using laser-cutting, one has to add



```
            GT.Timer timer = new GT.Ti
            timer.Tick +=<tab><tab>
            timer.Start();
        *****************************

                topLed.
            }
        }
}
```

void MulticolorLed.BlinkRepeatedly(GT.Color color1, TimeSpan blinkTime1, GT.Color color2, TimeSpan blinkTime2) (+ 1 overload(s))
Causes the LED to light in the specified color for the specified duration, switch to the second color for another specified duration, and repeat.

Fig. 3.1 Screenshot of the .NET Gadgeteer development environment. (a) The hardware designer in which users choose physical components, name and connect them. (b) A hardware component in the designer. (c) Connections between the components are computed by the IDE and displayed in blue. (d) Source code referring to the hardware component in (b), showing the auto-completion to help users understand the hardware capabilities.

finger joints to connect the flat panels that make up an enclosure (P6). Lastly, in generic virtual design environments, getting a sense of the size of objects one is designing is difficult (P2). We believe that by integrating the prototype hardware components closely into an enclosure-specific design environment will alleviate these problems above.

To this end, we develop a virtual design system for the creation of laser-cut prototype enclosures, called *Enclosed*. For this system, we exploit the fact that users have the hardware parts that need to go in the encasing available to them: we focus user interaction on the 3D representations of those parts, making them *reference objects* (see Section 1.2). I.e., through the parts' immediate availability and prominent integration into the design environment, we hope to further spatial understanding in designers (P2). To this end, users can directly compare the size of electronic parts on the screen with objects in their hand. In *Enclosed*, designers can place the hardware components directly on the virtual enclosure model (which is comprised of planar panels), and modify that model through these components. For example, when a user places a large display module on a panel, the encasing is resized so that the display has enough space. Thus, interaction with our system always yields a fabricable and sound result (P1). Further, we actively integrate the fabrication specific aspects of physical parts (specifically, the mounting cutouts they require) by annotating their 3D representations (P3). A push-button part, for example, requires three cutouts: two screw holes to mount it, and one hole for the button itself. We annotate the 3D model of the button component with this information and utilize it when computing the outlines for final fabrication. The latter, algorithmically computing outlines for later production, frees designers from fabrication-specific knowledge (P6): *Enclosed* can automatically produce the

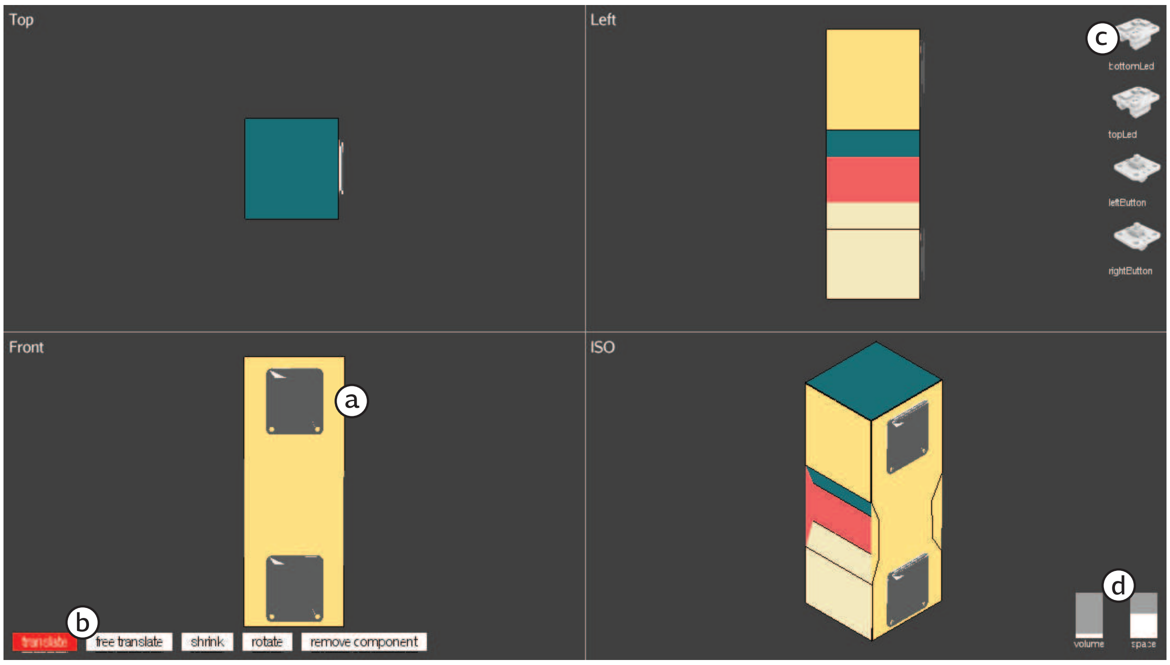


Fig. 3.2 The Enclosed user interface. (a) The enclosure preview with parts placed on the front panel. (b) The available actions with the currently active one highlighted in red. (c) The component palette from which users can drag hardware modules onto the enclosure panels. (d) Indicators for how much space has been used on a predefined sheet of material, and if the enclosures volume is sufficient to house the internal parts.

plans necessary for fabricating the encasing, including inter-panel connectors (finger joints and screw joints) and cutouts.

Our system extends the Microsoft .NET Gadgeteer platform [29]. This platform provides the physical hardware components, their virtual 3D models (which we annotate with the cutout annotations, as described above), and an IDE for software development (see Figure 3.1). We extend this platform through our enclosure design tool, that integrates directly into the Gadgeteer environment. Situated at the end of the prototype development process, we explore the *reference object* concept (see Section 1.2) using Gadgeteers hardware components. In this chapter, we contribute the design and implementation of *Enclosed*, and discuss lessons learned from using the system to develop three alarm clock prototype enclosures.

3.1 User Interface

The user interface of *Enclosed* (see Figure 3.2) is component-centric, meaning that all interaction with the system happens with respect to the device's hardware parts. This is in the same vain as the .NET Gadgeteer IDE which also, by focusing on the components, frees users from implementation specific knowledge. Users need not know about the specifics of the underlying hardware, as the development environment offers instructions which module needs to be connected where. When writing the device firmware, hardware components become instances of the object-oriented programming language used. The source-code editor offers auto-completion for the abilities of the hardware (see Figure 3.1, d), again freeing users from implementation specific knowledge. We develop our system in this component-centric vain, also freeing users from implementation specific (i.e., fabrication specific) knowledge.

We integrate *Enclosed* in the Gadgeteer development environment, so that once the hardware has been connected and the software is written, we can start designing the enclosure. Our enclosure editor receives the bill of material from the Gadgeteer IDE, and presents this list to the user upon startup. Here, users can choose whether a component is to be considered internal or external[1]. Internal components have no direct contact with prototype users and thus need no estate on the enclosure itself. Examples of internal components include batteries and WiFi or Bluetooth modules. External components serve their main through interacting with users or other objects. Displays, buttons, LEDs, wired connections (for Ethernet and power), and actuators are examples of such modules. Confirming the list of components leads to our enclosure design tool (Figure 3.2).

3.1.1 Placing components on the enclosure

Enclosed starts with an initial box-shape, on which users can place components, and which serves as starting point for new designs. To place a part on a face of the enclosure (e.g., the initial box, but also at any later stage), users drag the part out of the component palette (see Figure 3.2, c) and place it on the desired panel. If there is no space on that target panel, or the module is too big to fit on the face to begin with, the enclosure will resize automatically to provide the required space.

3.1.2 Translating components to resize the enclosure

Moving the component on the panel not only translates the component, but can also resize the panel. There are three cases depending on the currently selected action (which are

[1]Whether a component is internal or external is not unambiguous. For example, a buzzer is an internal component in that it is not operated by the user or would require user visibility. However, the buzzer is likely to be louder when the sound produced by it does not have to penetrate enclosing walls, hence if it is considered an external component. Thus, we let users choose whether a buzzer is internal or external.

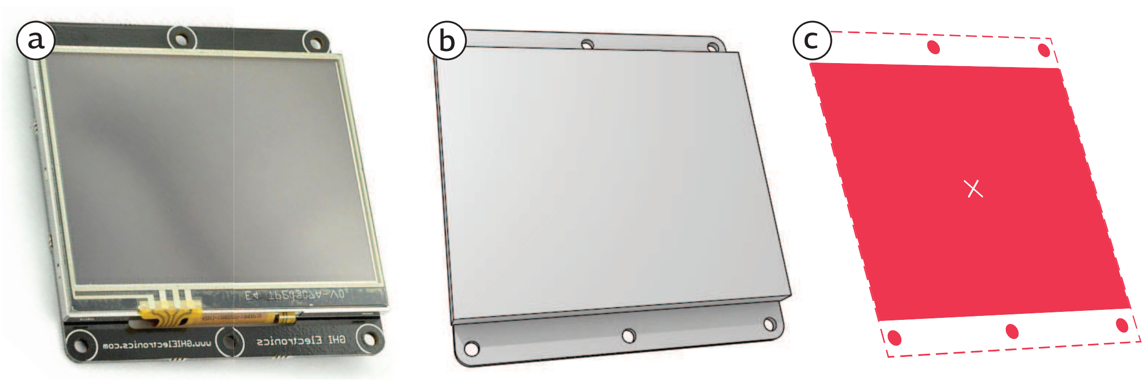


Fig. 3.3 (a) The Gadgeteer T35 display component manufactured by GHI electronics. (b) The 3D model as provided by the Gadgeteer platform. (c) The additional annotations (cut-outs are drawn solid, bounding rectangle as dashed line) to use it with our editor.

activated by holding down the according number key):

- *Translate* action: the component is translated on the panel only and never away from it. The enclosure does not shrink when the component is moved away from an edge but the component "snaps" back on the panel.
- *Free Translate* action: the component can be moved away from the panel, causing the enclosure to resize (grow and shrink) to again enclose the component.
- *Shrink* action: shrinks the enclosure when the component is moved towards the inside of a panel.

There is also a *Remove Component* action which allows the user to remove parts from a panel. Users can translate the modules in a continuous mode (default) or discrete mode (snap to nearest 5mm).

3.1.3 Rotating components to rotate/split/join enclosure panels

Panels can be rotated by selecting the *rotate* command (holding down the 3 number key), clicking on a component and, while the mouse button is still held down, moving the mouse to the left/right side to determine the angle of rotation.

The edge around which the panel is rotated is determined by which edge intersects the selected component's *manipulation range*: a fixed margin around the part model. The edge of rotation is chosen by checking which horizontal edge intersects with the manipulation range. If multiple horizontal edges intersect in the manipulation range, the lowest edge (the edge with the smallest y-coordinate) is chosen. We limit rotation to horizontal edges, as rotation around vertical edges would lead to enclosure designs that can not be produced on a regular three-axis laser-cutter. If no horizontal edge intersects with the manipulation range, a new edge is created at the lower horizontal end of the manipulation range, thus splitting the existing panel into two panels. When the angle between two panels is reduced to a value less than 2.5 degrees, the editor joins the two panels to become one and removes the edge that previously separated them. Users rotate the components in a continuous mode (default) or discrete mode (snap to nearest 10°).

3.1.4 Continuous vs discrete mode

All translation/rotation operations come in a continuous (default) and discrete version. The latter is activated by holding down the control-key and causes the operations to operate in fixed increments - 5mm steps for translation, 10° increments for rotation. In discrete mode, the translation distance/rotation angle is displayed next to the cursor. Actions are highlighted in red when they are continuous, green when used in discrete mode.

3.2 Implementation

Our implementation of Enclosed has three main elements: the enclosure graph which represents the encasing as a set of interconnected panels, the editor UI, and the laser-cutter outline generation algorithm.

3.2.1 Enclosure Graph

We represent an enclosure as a graph loosely resembling its bill of material. An enclosure consists of a set of panels, components, and the connections between them. In our enclosure graph, components and panels become vertices, connections between them become edges. A *panel* is represented as a set of vertices defining a polygon $P = (\mathbf{v}_i, \ldots, \mathbf{v}_{i+n}),\ \mathbf{v}_i \in \mathbb{R}^3$ in 3D space. All $\mathbf{v}_i$ of a panel must be coplanar, and there must be a polygon edge that is not parallel to the first one: $\exists \mathbf{v}_j, \mathbf{v}_{j+1}$ so that $(\mathbf{v}_{j+1} - \mathbf{v}_i) \cdot (\mathbf{v}_{i+1} - \mathbf{v}_i) \neq 1$. *Components* consist of a 3D mesh model representing the component visually and cutout annotations – circles and rectangles – located on faces of its axis-aligned bounding box (supra Figure 3.3). We enumerate the bounding box faces to uniquely identify to which face cutout annotation belongs to. Our model allows *panel-to-component* and *panel-to-panel* connections. A panel-to-component connection $(P, C, \mathbf{x})$ between the panel P and component C places the component on the panel, so that the components axis-aligned bounding box is located

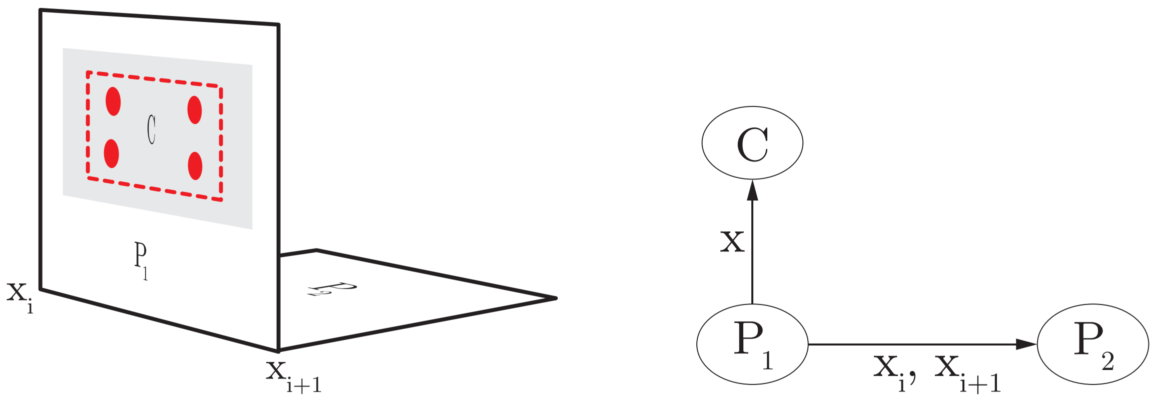


Fig. 3.4 A partial enclosure (left) and corresponding enclosure graph (right), consisting of the adjacent panels P_1, P_2 connected via the edge x_i, x_{i+1}, and a component C placed on P_1. The gray rectangle is the *manipulation range* of the component C.

at $\mathbf{x}_i + \mathbf{x}$ (where $\mathbf{x}_i$ is the first vertex of the panel P). Panel-to-panel connections can be established between adjacent panels P_n, P_m along their shared edge $(\mathbf{x}_i, \mathbf{x}_{i+1}) \in P_n, P_m$.

3.2.2 Editor User Interface

We implemented the *Enclosed* editor UI in Java using the *jMonkeyEngine*[2]. The 3D component models were converted to the surface-tessellation file format and annotated using custom XML-based descriptors. In the following, we detail the technical implementation of the actions available to users.

3.2.3 Translating components on panels

When a part is moved on a panel, we must update information regarding the component, panel, and associated connections. We have to consider three cases: (i) *growth:* component is moved outside of the panel bounds; (ii) *reduction:* component is moved towards the inside of the panel and shrinks the panel in the process; and (iii) *neutral:* component is moved onto the panel without any effect on the panel size. In the *neutral* case, we only have to update the position of the component on the panel. The *growth* and *reduction* cases require more computation.

First, we determine if a panel edge intersects with the component's manipulation range. If such an intersection is found and the part is moved towards the outside of the panel (*growth* case), we move the other panel adjacent to the intersecting edge, so that the edge

[2]http://jmonkeyengine.org/

is no longer intersected. When the module is moved towards the inside of the panel and shrinking is allowed (*reduction* case), we move the adjacent panel by the component's displacement vector, hence shrinking the enclosure. After all these operations, the part position is updated by storing the new position in the respective *panel to component* connection.

Formally, we find all edges $\mathbf{x}_i, \mathbf{x}_{i+1}$ that intersect the component's manipulation range when the component is moved to $\mathbf{x}'$, and their adjacent panels P_i. All panels P_i are translated along their normal vector so that the respective edge no longer intersects the manipulation range. To translate a panel, we move all its vertices x_j (and thus the edge vertices $\mathbf{x}_i, \mathbf{x}_{i+1}$), and the offsets of components connected to that panel by the translation delta. Lastly, we update the original components panel-to-component connection so that $(P, C, \mathbf{x}')$.

Rotating components on panels

Horizontally rotating a component on a panel can result in the panel being subdivided (note that we support no other form of component rotation). To split a panel, we create two new panels along the subdivision edge and rebuild the *panel to panel* and *panel to component* connections. A subdivision edge must

1. intersect with exactly two existing panel edges (can be violated by convexity);
2. be parallel to an existing edge to ensure a fabricable subdivision outcome;
3. not intersect with a component on the panel.

Such a subdivision edge is found parallel to the horizontal panel axis, intersecting the lowest point(s) of the component's manipulation range. To subdivide the panel, we create two panels to substitute the existing one. First we find the two panel edges intersected by the new subdivision edge – if there were more than two intersecting panel edges, criterion 1 would be violated. This also determines which vertices belong to which panel as the panel polygons follow a clock-wise winding order. After creating the new panels, the connections of the old panel are mapped to the new panels. Each old *panel to panel* connection is replaced with at least one, and at maximum two new *panel to panel* connections. All previous *panel to component* connections are mapped to either of the two new panels, depending on which of the new panels the global part position comes to rest.

3.2.4 Outline Generation

The laser-cutter outline generation algorithm is an essential part of the system as it frees users from fabrication specific design work, such as designing panel joints and accounting

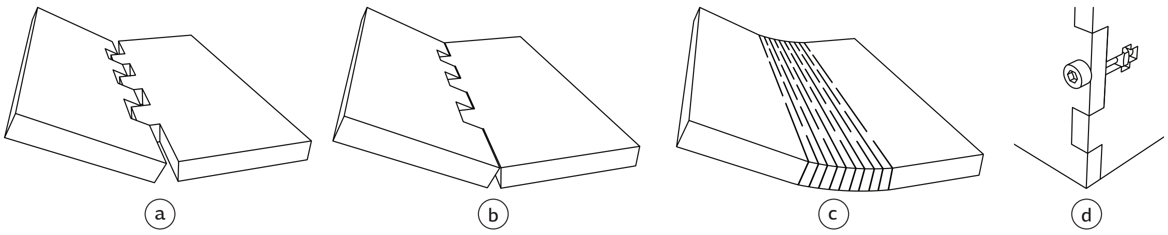


Fig. 3.5 Four different inter-panel joints. (a) was produced with the naive approach resulting in a too conservative size reduction. (b) was produced using our sin rule. (c) is a possible way of "joining" panels and exploits the flexibility of the production material. (d) is a screw joint that uses a bolt as connector.

for material thickness. We generate outlines to be cut using a laser cutter or CNC router and check for certain production specific issues in the model through the following algorithm:

1. **Project each panel to 2D** using the edges $\mathbf{x}_i, \mathbf{x}_{i+1}$ and $\mathbf{x}_j, \mathbf{x}_{j+1}$ to construct a local orthographic coordinate system.
2. **Assign each panel edge a joint gender** based on the directionality of the corresponding panel-to-panel connection in the enclosure graph. For example, consider the enclosure in Figure 3.4. Along the edge $\mathbf{x}_i, \mathbf{x}_{i+1}$, panel P_1 would have male connectors, P_2 female ones due to their roles in the enclosure graph (P_1 is source, P_2 is sink).
3. **Account for material thickness** by moving male edges inwards (with respect to the panel polygon) to shrink them appropriately. Edges are reduced depending on the angle of inclination with their adjacent panel. We found that reducing an edge by $m \sin(P_1 \angle P_2)$, where m is the material thickness and $P_1 \angle P_2$ is the angle of inclination between the two panels along the edge currently under consideration. This produces aesthetically pleasing results (Figure 3.5, a, b).
4. **Add joints to connect the panels.** This system is capable of adding three types of joints (see Figure 3.5), depending on the available space: pure finger joints, screw joints and a combination of the two. We fixed finger joints to be 10 mm wide, and add to each edge the largest odd number of finger joints that fits into the space. If an edge has at least three or more finger joints, the middle joint becomes a screw joint. If an edge has only one finger joint and there is enough space, the finger joint is replaced with a finger/screw joint combination (which is 15 mm wide to account for the screw diameter). Joint gender is based on the role determination in step one. Note, that joints have to be generated in a coordinate system that is invariant to the size reduction performed in step two. Otherwise the resulting joints do not fit together

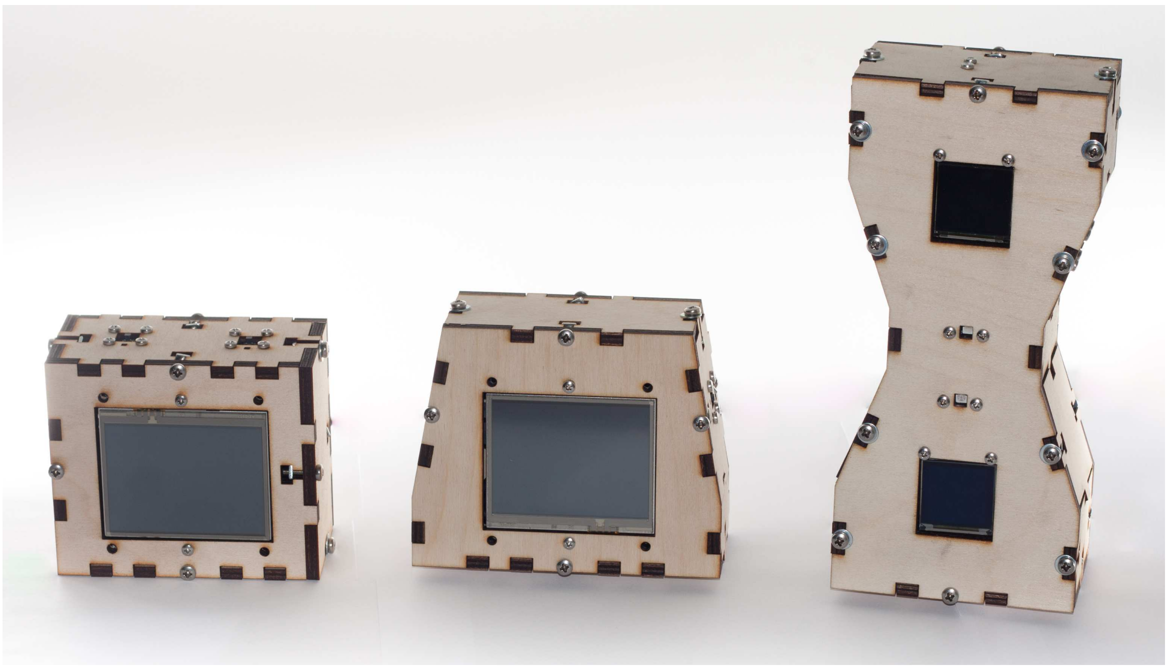


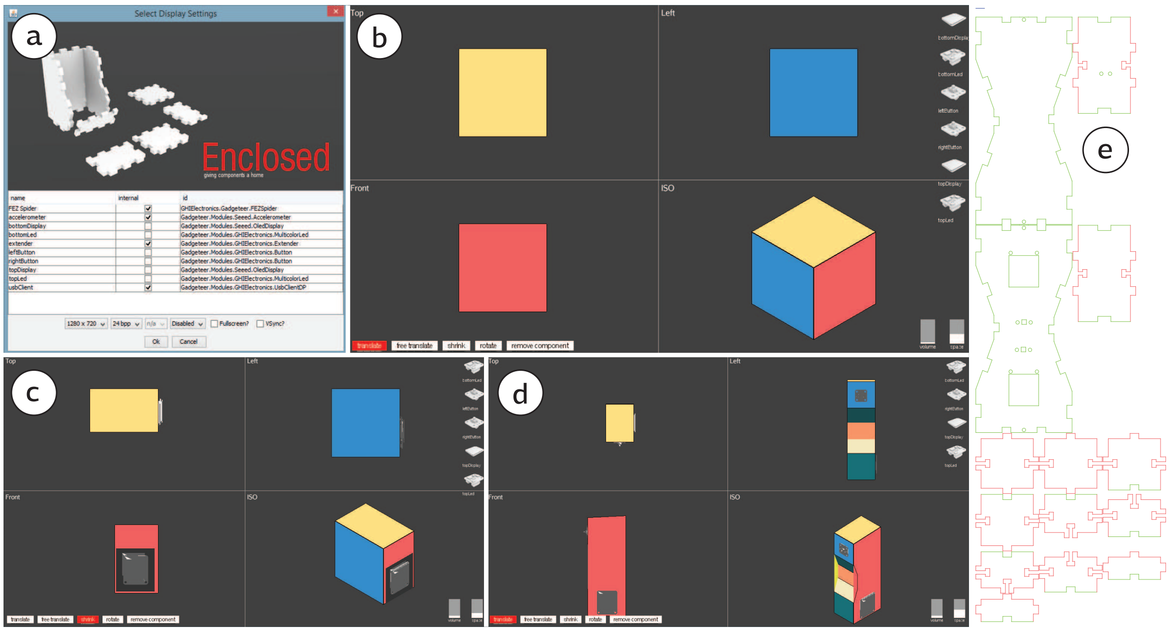
Fig. 3.6 Three alarm clocks, all of which share a similar bill of components but exhibit different shapes and behavior. On the left side is the *brick* prototype which represents the simplest alarm clock. In the center is the *pool* enclosure which has the buttons on the angled side panels. The third prototype on the right-hand side, bears deliberate resemblance with an *hourglas*.

as they are shifted by the length the panel was reduced.

5. **Place the computed outlines on the material sheet** so that material use is minimized. Placing the outlines, is an instance of the strip packing problem: place a set of rectangles of different size on a strip of fixed width but infinite height, so that the total height is minimized [133]. To compute the layout, we use the *First-Fit Decreasing Height* packing algorithm [134] as it has a low computational complexity of $\mathcal{O}(n \cdot \log n)$ and thus can be run in real-time. Further, the panels tend to be of similar size, lending themselves to be packed with this algorithm.

3.3 Application Example

To demonstrate the use of our system, we walk through the development of alarm clock prototypes – specifically the *hourglass* model (see Figure 3.6). We fabricated these enclosures out of $600 \times 300 \times 6$ mm plywood using the outlines generated from our system (see appendix A.1). The enclosures were assembled using M3 bolts for mounting the components



name	internal	id
FEZ Spider	✓	GHIElectronics.Gadgeteer.FEZSpider
accelerometer	✓	Gadgeteer.Modules.Seeed.Accelerometer
bottomDisplay	☐	Gadgeteer.Modules.Seeed.OledDisplay
bottomLed	☐	Gadgeteer.Modules.GHIElectronics.MulticolorLed
extender	✓	Gadgeteer.Modules.GHIElectronics.Extender
leftButton	☐	Gadgeteer.Modules.GHIElectronics.Button
rightButton	☐	Gadgeteer.Modules.GHIElectronics.Button
topDisplay	☐	Gadgeteer.Modules.Seeed.OledDisplay
topLed	☐	Gadgeteer.Modules.GHIElectronics.MulticolorLed
usbClient	✓	Gadgeteer.Modules.GHIElectronics.UsbClientDP

1280 x 720 · 24 bpp · n/a · Disabled · ☐ Fullscreen? · ☐ VSync?

Ok · Cancel

Fig. 3.7 Designing an alarm clock prototype. (a) users decide whether components are internal or external. (b) Enclosed right after being launched from within the Gadgeteer IDE. (c) sizing the enclosure using the display as *reference object* (d) the left side of the enclosure is shaped like an hourglass. (e) the final laser-cutting outlines, colored depending on the joint gender (for illustration purposes only, has no influence on production).

and M4 bolts for the finger-screw joints.

We start in the .NET Gadgeteer IDE which is based on Microsoft Visual Studio. There we add the required hardware components to our project: a mainboard, two buttons, two displays, two LEDs, an accelerometer, and an extender module that connects to the buzzer (see Figure 3.1, a). Once all parts have been added, connected, and named appropriately, we write the device firmware - again in the Gadgeteer development environment (see Figure 3.1, d). After uploading the firmware we are ready for the next step: design the enclosure. For this device, its suggestive shape also determines its function – if the hourglass is turned around, the alarm is snoozed. As we see in the following paragraphs, *Enclosed* lets us focus on that shape, rather than on creating the panel connectors or dealing with a complex design environment.

To start the enclosure design, we select *Enclosed* from the “Tools” menu within Visual Studio which starts our enclosure editor. We must now choose which components we consider internal, and which are external. Our system has these choices preselected, based on the type of component (see Figure 3.7, a). Confirming this selection yields the main editor window, initialized with a box-shaped enclosure and a component palette showing all parts that need to be placed on the encasing (see Figure 3.7, b).

First, we bring the initial box to size using the displays that will be mounted on the front panel. To place a display on the front panel, we drag it out from the component palette to its desired location. By moving the display downwards, beyond the enclosures current size, we increase the height of the prototype. By moving the display inwards from the sides, while having the shrink mode active (keeping number key 2 pressed down), we decrease the width of the box (see Figure 3.7, c). In this process, the *reference object* concept comes to play. Because we have those displays in front of us, we can get a sense of how big the enclosure will be in the end and size it accordingly.

With the proportions correct, we start shaping the device. To this end we place the left side button on the left panel, again using drag & drop. We then move this button 50 mm upwards on the panel using the discrete mode, activated by holding down the control key. This places the button just above the previously placed display. Here, we click on the button part while the discrete rotation mode is active, causing the display to rotate horizontally in 10 degree increments. Upon release of the click, the left panel is split in two and the new upper face is rotated by the 20 degrees that we previously rotated the component to. We continue these translation/rotation actions until the left side is shaped like an hourglass (see Figure 3.7, d). Right-clicking in the upper right quadrant shows the enclosure from the right side, rather than the left. We repeat the previous process to shape the right side of our alarm clock. Lastly, we place the other parts to finish the design.

Now that the device design is completed, we generate the outlines required for laser-cutting the enclosure panels. After pressing the *P* key (for print), *Enclosed* automatically produces the laser-cutting plans, including the required finger joints and cut-outs (see Figure 3.7, e). Note, that at no point we had to manually consider these or other fabrication-specific artifacts (e.g., account for fabrication material thickness), but were solely concerned with the design of the shape of the device.

In a last step, we laser-cut the panels and assemble the enclosure. This yields the device as shown in Figure 3.6, right.

3.4 Discussion

We used the .NET Gadgeteer platform as a case study and utilizing its existing hardware components and software development tools. However, the idea of designing enclosures using electronic parts as reference objects is applicable to a broader array of prototyping frameworks. Prominent toolkits such as the Arduino[3] and Phidgets [25] or generic micro-controllers could be used as well, provided 3D models exist for the components.

More generally, the concept of integrating objects which are at the designers immediate disposal as reference objects, and focusing interaction on them to instil a better spatial understanding of the object under design (P2) generalizes over other domains: for example, the introduction of a hand model could ease jewelry design as the items being designed can be immediately fitted to the virtual hand. At the same time the designers physical hand turned reference object could instil some spatial familiarity in the design environment.

3.4.1 Limitations

The outline generation algorithm as currently implemented has limitations. In extreme cases it can lead to an unfabricable outcome. During design time we do not explicitly check if the shrinking of panels (to account for material thickness) leaves us with not enough space to mount a component. To solve this we could continuously regenerate the outlines after each modification and thus detect such problems early on. Further, devices produced with this system might not always stand flat, as screw joints are not optimized for planarity. In its current form, the outline generation determines the joint gender (and thus bolt orientation) through the edge role in the enclosure graph (see Section 3.2.4). While this ensures a fabricable outcome, it results in the non-flat-standing screw-joint distribution. Optimizing the joint genders (e.g., through heuristics or simulation), or introducing new types of screw joints, would alleviate this problem.

All components, more precisely their 3D models, need to be annotated before they can be used with our system. Currently, this is a manual process. Users have to manually describe where cutouts need to be provided, so that the component can be mounted on a panel. This task has to be performed only once when a new component is introduced (e.g., a new Gadgeteer component becomes available on the market). Further, this task could be automated through computational methods e.g., by detecting holes or protrusions on the 3D model.

3.4.2 Automatic generation of fabrication-specific aspects

A key feature of our system is that it frees designers from fabrication-specific aspects through our outline generation algorithm: users do not have to create inter-panel connectors or account for material thickness themselves. This is beneficial to two user-groups:

[3]http://https://www.arduino.cc

first, “fabrication experts” do not have to spend the time and effort to design such features. Second, non-expert users do not need to know about of such connectors and how to design them, thus we enable novices to design enclosures in the first place (P6).

We could make use of the fabrication material properties, instead of using externally supplied fasteners. “Living hinges” (Figure 3.5, c) use the materials flexibility and could be used to connect multiple panels, if their adjacent angle allows that. Exploiting material properties also enables the direct fabrication of interactive elements e.g., integrated buttons. Designing such hinges or functional elements requires a good understanding of the material used, and the characteristics of the techniques themselves. Thus, they are difficult to design manually. Enclosed could encapsulate this design knowledge and automatically generate such connecting hinges or buttons.

To pack the generated outlines on a sheet of material, we utilize a simple but efficient strip-packing algorithm. We have found this form of optimization to work well for enclosures with moderate aspect ratios (height vs. width ratio). However, as illustrated by the walkthrough, the algorithm produces undesirable packing configurations for such extremely tall or wide devices. Our current implementation could be adapted by rotating each outline to its smallest dimension, thus consuming less space along the packing direction. Alternatively, one could integrate more advanced methods to place laser-cutter outlines [125] which have been proposed since the development of this system.

3.4.3 Enclosed Away from Mouse and Keyboard

As presented, Enclosed is a purely virtual design environment. However, the component-centric interface we developed is well suited for other scenarios. Using the physical components directly, as tangible proxies, for example would allow us to transport enclosure design directly into the physical world. As such, we could extend the component-centric interface to interactive tabletops, or to augmented reality systems.

Schneegass and colleagues present a system that explores this idea of using the electronic parts of prototypes to design their shapes [135]. While focusing primarily on sizing simple shapes, their method could be extended using this system. We would show the enclosure in isometric view, and whenever a part is placed on the tabletop, the enclosure rotates so that the panel the item is placed on becomes co-planar with the tabletop surface. Then, the component-centric modifications outlined above would become available through the physical part directly.

Similarly, the reference objects (hardware compoments) could serve as markers in an augmented-reality setup. Initially, the box-shaped enclosure would be displayed in free space before the user. By taking up a physical part, and aligning it with the displayed model, users would place components. Once a component has been placed, it could be used to manipulate (move and rotate) or modify (resize, slant or subdivide) the encasing. In chapter 5, we present a similar concept which makes use of augmented reality and existing objects to enable novices to design objects for digital fabrication.

3.5 Summary

In this chapter we developed a component-centric interface guided by the *reference object* strategy. Virtual 3D models of physical objects which are at the designers immediate disposal become central interaction elements. This way, we hope to further spatial judgement, as designers can relate the model shown on the screen directly to the physical object in front of them (P2). By displaying the object-under-design from all sides, we minimize navigational requirements. And lastly, through offering only actions that yield a fabricable result, we ease the interaction with the system. (P1). We further provide support for fabrication-specific aspects, freeing users from concerning themselves with implementation issues (P6), shifting the focus more towards the design of the object itself.

In this chapter we contribute concrete implementations of the aforementioned concepts through an enclosure design system. This system is closely integrated into the .NET Gadgeteer landscape, utilizing the hardware components of this platform as reference objects. We developed a UI that focuses interaction on the these components, and always yields laser-cuttable results. Our system automatically generates the required outlines to fabricate the encasings on a laser-cutter. To this end, we contribute a novel outline-generation algorithm. Using the implementation, we presented a walkthrough designing alarm clock prototypes, demonstrating the systems use and benefits.

The next chapter integrates physical measurements in virtual design environments. Dimensions and angles, which were integrated as virtual 3D models in this chapter, will be physically represented through active tangible devices.

4 SPATA
Active Spatio-Tangible Measurement Tools

Virtual design environments, such as parametric CAD tools, would benefit from the bi-directional integration of physical measurements. Currently, virtual design environments and the measurement of physical dimensions are disconnected. This impedes on fabrication-aware design processes, as existing sizes and angles are cumbersome to manually integrate into new designs (P3). In this chapter we introduce two new active spatio-tangible measurement tools (calipers and protractor, see Figure 4.1, b - c) which connect measurements in physical space to the virtual realm, and vice versa – thus are bi-directional in that sense.

When we design physical objects, their dimensions and angles are important design decisions to make. In the previous chapter, we support such decisions through annotated reference objects: predefined bundles of cutouts and measurements. In this chapter, we unbundle such measurements to support decision making based on arbitrary existing objects, function, personal preference or aesthetic considerations. For example, we might want to choose the dimensions of an object so that it comfortably rests in a users hand, hence will need to measure width and length of said hand. In purely virtual design environments such decisions necessitate the use of disconnected physical measurement tools. We have to pick up a ruler or pair of calipers, perform the measurement and manually transfer the value back into the digital realm. If we want to scrutinize previously chosen dimensions, for example to get sense of how big an object is (P2), we have to read their value off a computer screen, switch to on analog tools and manually move the caliper jaw to that length. There is a clear disconnection between the physical measurement tools and the virtual design environments used during the construction of physical objects.

To remedy this disconnection, we introduce spatio-tangible tools for fabrication-aware design called the *SPATA tools* (implementing the *Active Spatio-Tangible Measurement Tools* strategy, see Section 1.2), which connect the physical and virtual world and seamlessly work in both. Such tools can measure length, angle or other properties in either world (physical or virtual) and transfer the measured property into the other space, respectively. For example, to size a model in a virtual environment, one can measure the width of a physical object and have that dimension automatically applied as new virtual model width (P3). Conversely, one can measure the length of the virtual model and have that output in physical space, e.g. for comparison (P4). In non-fabrication contexts a similar concept of digitally connected measurement tools exist e.g., HandSCAPE [136] is a digital measurement tape that can transfer its measurements to an interior design application. The commercially available Mitutoyo USB calipers[1] simulate keyboard input when their value changes. However, these devices are uni-directional: measurements can be transferred from physical to virtual, but not the other way around.

In this chapter, we create two concrete instances of *Active Spatio-Tangible Measurement Tools*, both of which are digital adaptations of two commonly used measurement devices: calipers for measuring length, and bevel protractors for measuring angle. The SPATA tools can measure their respective value (length or angle), but are also actuated so that they can actively present it in the physical world: the calipers have a self-actuated lower jaw that can physically represent length; the protractor can move its blade to output an angle (Figure 4.1, red parts). Our tools integrate closely into virtual environments used to design fabricable artifacts: parametric modeling, mesh-based modeling, and 2D design. In those environments, we create shapes following a series of prescribed modeling tasks (e.g., by creating boxes, cylinders or rectangles). Those tasks, in a fabrication-aware context, often require physical measurements taken from existing objects. To reduce the need to switch between the virtual and physical world, we partially offload control to the measurement tools. For example, to model a box-shaped object (e.g., an enclosure) using SPATA, users can measure all three dimensions (width, height, and depth) in sequence without having to put down the SPATA calipers or manually type in measurements.

To further support the design tasks, our tools can sense their orientation, display additional information (such as estimated fabrication time or the amount of required material – P6) and have a button built-in. We combine these capabilities to provide a more integrated

[1]http://www.mitutoyo.co.jp/eng/index.html

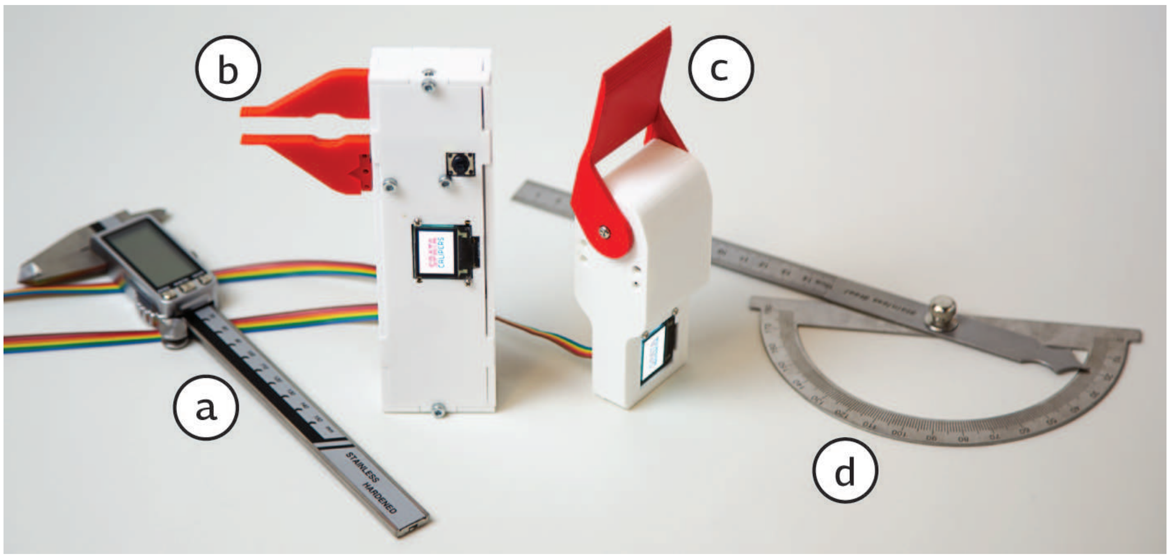


Fig. 4.1 The two SPATA tools next to their original counterparts. (a, b) calipers for size in-/output, (c, d) the protractor for angle in-/output.

and convenient design experience when designing for the physical world. In total, we make the following three contributions:

1. We present digital adaptations of calipers and bevel protractors, that can bidirectionally transfer information between the physical and virtual world, providing an active tangible interface supporting fabrication-aware design.
2. The integration of both tools into three design environments commonly used for fabrication-aware design: parametric modeling, mesh-based modeling and 2D design (e.g., laser-cutting or circuit board design).
3. Lastly, we demonstrate both tools and their integration in three application examples that highlight the benefits of the bi-directional information transfer and design-environment specific task support.

4.1 SPATA tools

SPATA tools are intended to become part of existing virtual design environments e.g., parametric CAD. When creating new objects, users often need to transfer a measurement into the virtual environment or visualize another measurement to help make a design decision. We go beyond the trivial step of digital acquisition of measurements and transfer to the virtual environment. Our tools provide a higher level of context-awareness to the steps of the

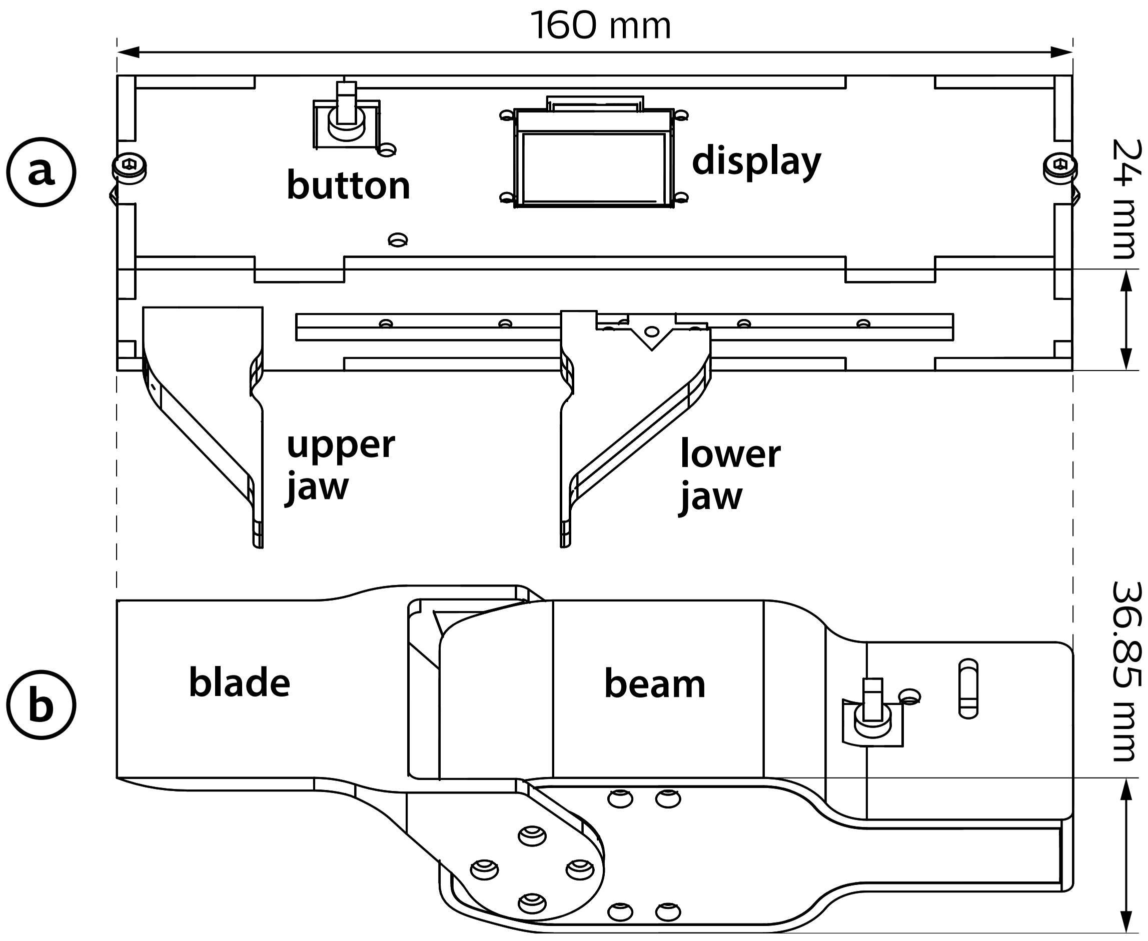


Fig. 4.2 (a) The SPATA calipers and (b) the protractor design. Both have a fixed and actuated part, a five-way button and a display (for (b) the display is on the other side, see Figure 4.1).

modeling tasks and become integral to their progression. The specific features of the tools are:

- **Measuring and presenting physical values**: SPATA can measure length and angle, as well as present those values in physical space. As both tools are actuated and computer-controlled, they can tangibly output physical dimensions.
- **Bidirectional value transfer:** Both tools automatically transfer a measurement from the virtual world to the design environment, or the opposite way. Users do not have to manually enter measured values or manually move the jaw/blade (see Figure 4.2) of the SPATA tools.
- **Design task integration:** We support design tasks by using transfered values in context (e.g., as correct dimension when modeling a box), and by partially offloading task control to the tools. Through the built-in display, five-way button and orientation

sensing, users can navigate task steps, select modes and navigate in the 3D world.

In design practice, many different tools are used for measurements. Which tool is used depends on the type of measurement (e.g., size vs. angle) and the order of magnitude at which the measurement is taken (e.g., millimeters vs. meters). Large-scale measurements, for example using a measurement tape, have been uni-directionally integrated into their respective tasks [136]. Small-scale measurements, as required for personal fabrication, have yet to be integrated.

We identified tools which are important in a personal fabrication setting, through a short questionnaire among 26 personal fabrication practitioners (experience in months: $M = 29.27$, $SD = 33.98$). The results of the survey indicate that for size input, calipers are the most prominently used tool (22 out of 26 practitioners). This potentially relates to the scale of objects targeted by fabrication processes, as well as the precision afforded by the calipers. The only angular measurement tool mentioned was the protractor (4 out of 26 practitioners).

4.1.1 SPATA Calipers

Calipers measure length, diameter and depth. We designed the SPATA calipers to resemble their analog counterpart (Figure 4.1, a-b). Their size of $160 \times 43 \times 24$ mm approximates the bounding box of traditional calipers. The jaws are shaped to support the measurement of length and inner diameter (using the thin front of the jaws, see Figure 4.2, b). As with the original, the upper jaw is fixed and the lower jaw can be moved from 0 mm to 100 mm; either manually or computer actuated. Our calipers are designed so that they can be held in one hand, with the display and button being easily accessible (Figure 4.2, b). The display is centered and recessed into the top, giving it a prominent and easily visible position. The button can be operated with the right thumb while holding the tool, particularly when holding it with a single hand (see Figure 4.4).

We implemented the calipers using audio sliders as actuators. Positional feedback is provided through a voltage divider relative to the sliders position. We drive the audio slider using a dual H-bridge, controlled through a custom PID controller implemented on the microcontroller. The positioning error is less than 1.5 mm – the measurement error is less than 0.5 mm. The enclosure is made from laser-cut acrylic.

4.1.2 SPATA Protractor

Protractors measure the angle between two lines or surfaces. They are often used to explore or measure the slant of a surface. Bevel protractors, a common kind of protractors in mechanical design applications, consist of a beam and a blade. The beam is fixed, the blade can rotate around the center point where both intersect. We stay true to the beam and blade mechanism, with the latter attached to a servo motor (Figure 4.2, c). Both surfaces are co-planar in their default position (0 deg), and can assume angles from -90 deg to 170 deg which fits digital fabrication requirements.

The SPATA protractor is based on a Dynamixel AX-12A servo motor, which provides a serial interface and reports its orientation with 10 bit resolution. We can sense and actuate the angle with an error of less then one degree. A 3D printed enclosure provides the blade of the bevel protractor design, and encapsulates the electronics, including the servo motor.

4.1.3 Common Hardware

Both tools, calipers and protractor, are based on the same hardware platform (for schematics, please see Appendix A.2.2). An ATmega328p microcontroller controls the actuator and display. It also captures the button input and accelerometer values. The latter come from an ADXL335 accelerometer that is centered on a custom circuit board. A 4D Systems μOLED-96-G2 module is used as display. Our tools connect to a communication board that supplies 5 V and 9 V, and provides a USB/serial interface through an FTDI based USB-serial bridge. Both tools send their measurements and orientation at a rate of 10 Hz.

4.1.4 Client-side Integration

We integrate the tools and the design environments using a custom middleware layer. This middleware contains the communication with the tools, the access to the design environments, and the workflow logic. Both tools are free of integration specific logic, and do not require reprogramming to be used in different environments. The same is true for the design environments, none of them contains any workflow logic; all logic is contained in the middleware layer. For intuitions sake, we have included a source-code excerpt from our middleware in appendix A.2.1.

We implemented the integration for three different software packages which are used by digital fabrication practitioners. To integrate into Autodesk Inventor, representing parametric modeling, we used the API that comes with the software. Blender, which represents mesh-based modeling, can be scripted with Python. Hooking into the redrawing routine of Blender, we can read commands from a file/Unix pipe and execute them in the modeling environment, thus creating our own remote control API. Because Adobe Illustrator does not provide any extension mechanisms, we simulate key-strokes and analyze screenshots to

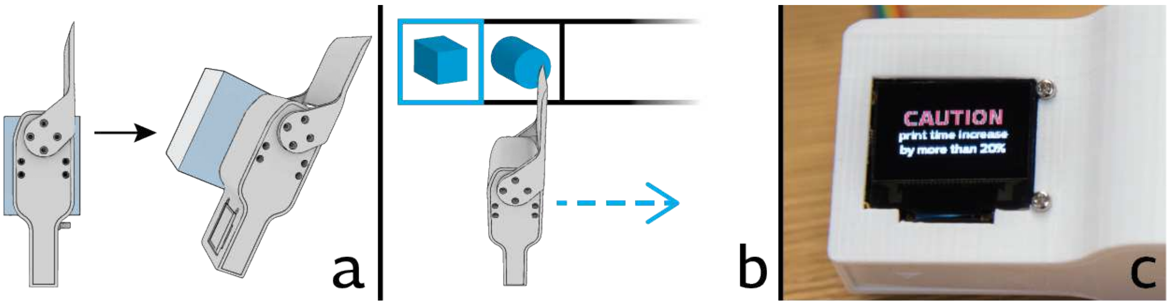


Fig. 4.3 (a) Rotating the SPATA tool to the perspective of the model on the screen. (b) Flicking the tool to the side changes the mode of operation. (c) SPATA tools showing fabrication related information.

control this environment.

4.2 Design Environment Integration

Automating the value in- and output from the physical world to the virtual design environment removes the need for manual value transfer. Existing projects and products, such as the *Mitutoyo USB calipers*, provide automated value input, but are not integrated with the design environments (other than simulating keyboard input). SPATA integrates into design environments, aiming to reduce the required context switches and to make the design process more convenient.

We integrate the SPATA tools into three types of commonly used design environments. Starting from a general 3D environment integration, we specialize to specific environments: parametric modeling and mesh based 3D modeling (see Section 2.3). The former is often used for product design, the latter for more artistic, organic modeling of shapes. Lastly, we describe how our tools integrate with 2D design environments, for example laser-cutting or circuit board design.

4.2.1 3D Design Environments

We implemented SPATA tools support for two 3D design environments: parametric modeling and mesh-based modeling. Both environments have common tasks which do not directly affect the 3D shape. For example, manipulating the 3D model of the object-under-design to see it from a different perspective, or selecting a new operation/tool/workflow within the design environment.

Viewing a model from different angles, zooming in and out, as well as panning the model are essential tasks during design. SPATA supports such operations by serving as tangible proxy. By making SPATA tools tangible proxies for the object being designed, users can change the model orientation by physically changing the orientation of the SPATA tool. This allows users to view and inspect a model from different angles (Figure 4.3, a). Both tools can represent a continuous variable i.e., the current zoom level. When in zoom mode, users can zoom in and out by moving the lower jaw of the calipers or changing the angle of the protractor blade respectively.

While designing, we need to navigate the design environments user interface to start some operation, change a value or select a tool. Often times, we need to alternate between two tools e.g., measuring or manipulating a length. Such selection tasks can be performed using quick mid-air gestures: flicking the tools to either side. Users could flick through a color swatch by quickly moving to the left or right (Figure 4.3, b). Manipulation/measurement modes could be changed by quickly moving forward or backwards.

Design decisions often depend on their influence on fabrication: their impact on printing time, material cost or if support structures become necessary. This information can be displayed on the second screen of the SPATA tools. For example, when slanting a surface of a 3D model that is going to be 3D printed, beyond some slope support material is needed. Presenting this information enables users to make an informed decision if they want to cross that threshold or not (i.e., make an angle 40 degrees instead of 45; see Figure 4.3, c).

4.2.2 Parametric modeling

Parametric modeling (see Section 2.3.1) is used by engineers and designers for product design and prototyping. It revolves around shapes typically found in man-made objects, such as cylinders, blocks and curves. Modern parametric modeling systems e.g., Autodesk Inventor or SolidWorks, are based on 2D sketches which are extruded or revolved into solid 3D objects. When drawing sketches or creating these objects, users need to constrain different dimensions, often using physical values, such as length and angle.

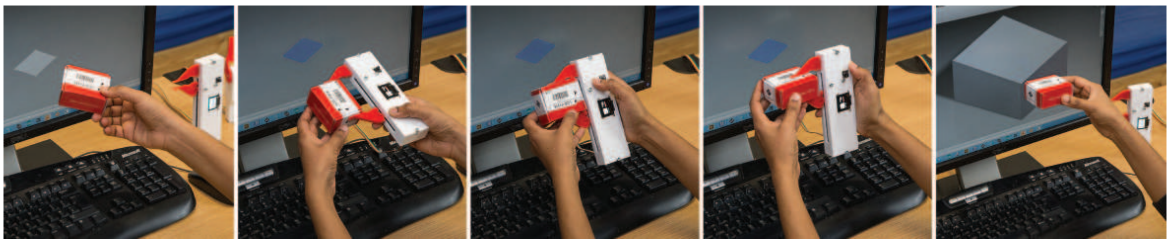


Fig. 4.4 Creating a box (from left to right): select the ground plane, measure width, height and depth.

SPATA supports the creation of boxes from a prescribed series of real-world measurements (e.g., width, height, depth; see Figure 4.4). The measurements can be preformed in rapid succession using the button on the SPATA tool, thus users can create a new cube with no context switch. A similar sequence exists for cylinders: first measuring the diameter, then height. After a primitive has been created, SPATA stays in this mode enabling a series of primitives to be built on top of each other, demonstrated by the *broken sprocket* use-case (see Section 4.3.1).

More complex shapes can be created by extruding or revolving 2D geometry. The height of the extrusion, or angle of revolution are often determined by existing physical artifacts, or by the liking of the designer; both of which are best determined in the physical world. Further, parametric modeling systems often support semantic actions, such as creating holes. We support those tasks by providing a continuous value input: if the value of the tool changes, it is directly used as the respective design parameter (e.g., extrusion height).

This mode is particularly useful if one wants to create a hole with respect to a physical artifact. For example, if a user in a previous step measured a box, and now wants to create a hole in it, they could use the physical box in conjunction with the SPATA calipers to determine the depth of the hole.

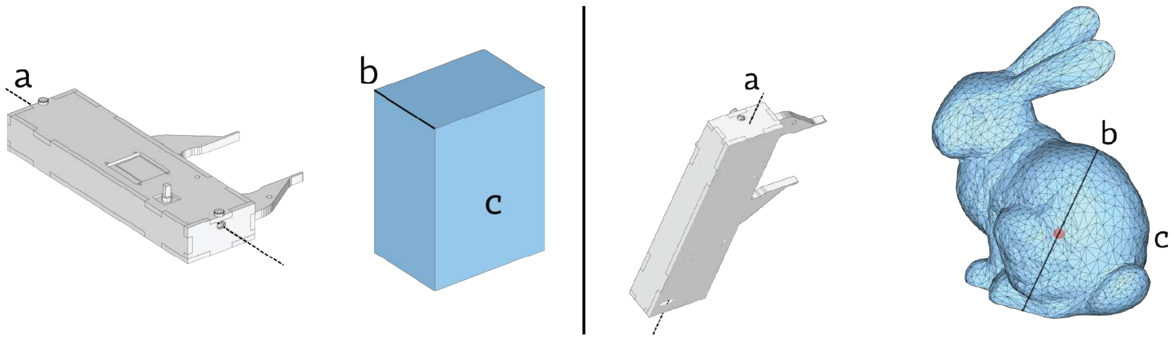


Fig. 4.5 (left) Feature based selection. The axis of the SPATA tool (a) is aligned with a feature (b) of the virtual object (c), causing this feature (i.e., edge) to be selected. (right) Ray-based selection. The axis of the SPATA tool (a) is fixed at a pivot point inside the model (c) and rotated around it (b). The intersection points of this selection ray select vertices.

Each SPATA tool can sense its orientation in space. When a SPATA tool is aligned with a feature e.g., an edge, hole or plane in case of the protractor, that feature is selected (see Figure 4.5). Once the selection mode has been enabled, the selection is updated continuously until it is confirmed using the button on the SPATA tools. Often times selection is part of another task, in which case the selection is transfered to overarching task e.g., scaling the just selected edge.

The calipers can transfer length into the physical world. To this end, users measure the length of a feature e.g., the length of an edge or diameter of a hole. The user first selects that feature, either using the SPATA based selection mechanism or using the design environments native one. Once selected, the calipers move to that length in physical space. A similar process is used for measuring an angle: after the user selects two planes either using SPATA or the environments native methods, the protractor transfers that value into the real world.

4.2.3 Mesh-based Modeling

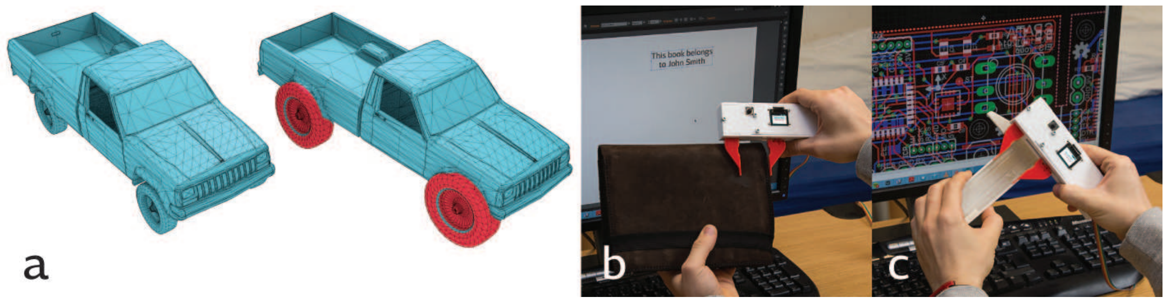


Fig. 4.6 *Local scaling:* (a) Deforming a mesh by scaling only the selected red parts. *2D design:* (b) Placing text on a physical artifact. (c) Ensuring a PCB fit's inside an existing enclosure.

Mesh-based 3D modeling is a general-purpose modeling paradigm, that is used for creating organic and artistic models in tools such as Autodesk Mudbox, Blender or ZBrush; see Section 2.3 for an introduction. To create new models, designers often start with geometric primitives which are then combined, subdivided and scaled. Vertices, the smallest unit of manipulation, are often directly manipulated to form the desired shape. Additionally to directly manipulating vertices, designers use tools like brushes and stamps to refine the shape. All these operations act on generic vertices and do not carry semantic information (as compared to parametric modeling).

We use the accelerometer of the SPATA tools for selection. By rotating a line around a fixed anchor point users can select vertices or vertex groups. This selection line by default extends to one side only, but can also extend in both opposite directions to select orthogonal pairs. The anchor point of the selection line can be moved by the user (see Figure 4.5). Depending on subsequent actions, the selection can be continuous so that every orientation change, modifies the selection. In this mode, we can use this method similarly to existing selection mechanisms e.g., like a brush to create vertex groups. It can also be a one-off selection, where the selection is confirmed using the built-in button.

We support global and local scaling. Global scaling is applied to the whole model, local scaling is applied to a specific selection of vertices or vertex groups. Scaling the whole model can be used to bring the model to a certain size based on a single dimension. Scaling a selection of vertices is often used to modify the shape locally and to articulate features of the model (see Figure 4.6, left). For local scaling we support a *select and scale* task, where users first select what they want to scale (using the previously described selection technique) and then perform the modification.

To measure length in the design environment, we select two vertices, or planes using the selection mechanism described earlier. We can also use built-in measurement tools (e.g., the ruler/protractor feature in Blender). To measure angle, we select two planes: two model faces, or a global plane and a model face; again using the previously described mechanism or environment specific selection techniques. In both cases the respective tool will output the value in the physical world, as well as on its display.

4.2.4 2D Design

2D design environments are used in many domains, such as for laser-cutting, desktop publishing (DTP) and electronic computer-aided design (eCAD). The often domain-specific design environments revolve around semantic objects such as circles, holes, text blocks and electronic components; objects that need to be arranged and scaled on a 2D canvas.

Most 2D design environments have a global coordinate system that spans the working area. Objects placed in the working area have an anchor-point in the coordinate system which serves as point of reference for transformations. Some design environments support snapping mechanisms with regards to that anchor point e.g., snapping to multiples of 5 mm when translating, or 45 degrees when rotating. SPATA supports such snapping mechanisms in form of tactile feedback e.g., the calipers physically snap to the underlying grid.

Translation is used to place objects in the working area; e.g., text on an existing object (see Figure 4.6, b). SPATA calipers can be used to place objects. Users first select the axis along which they want to place the object; this makes the calipers output, the objects current location along that axis, into the physical world. Second, any change of the calipers measurement is continuously applied as translation value along that axis, synchronously moving the object (with respect to the anchor point). Scaling is performed in the same way. After selecting the axis, the calipers output the current length along that axis and update the size with every change of measurement. During object placement, it maybe necessary to rotate an object. Entering rotation mode, causes the protractor to output the current orientation into the physical world. Changing that angle will immediately update the objects rotation around the anchor point.

We implement a semantic feature-based measurement strategy, as most two-dimensional artifacts have such semantic annotations/features (e.g., parts placed on a PCB or a rectangle drawn on a poster). Users first select the features they want to measure using the design environments built-in selection mechanism, which is typically mouse-based. When a single feature is selected (e.g., the outline of a printed circuit board, see Figure 4.6, c) the length of that feature is transfered into the physical world. When two features are selected at the same time, and they have an angle of inclination to one-another, we output that angle using the protractor.

4.3 Application Examples

We illustrate the integration of the SPATA tools into the three design environments. By walking through examples for each environment, we demonstrate the tools capabilities and how they make the design process more convenient.

4.3.1 Replacing a Broken Sprocket

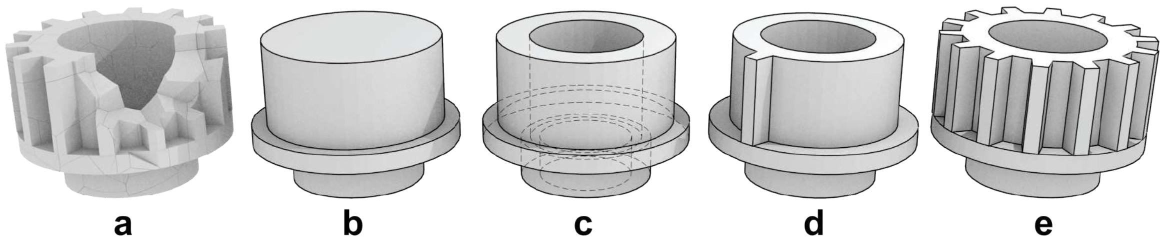


Fig. 4.7 Replacing a broken part. (a) the broken sprocket we want to replace, (b - d) the intermediary steps for modeling the replacement part, (e) the finished part.

In this example scenario we will model a working replica of a broken sprocket (see Figure 4.7, a) in a parametric modeling environment. Repairing broken parts with digital fabrication requires the creation of a printable model. Because the part is physically broken, we need to complete it while modeling it. In this case, 3D scanning the part is not feasible, as we require an exact geometrical representation of the object in order to repair it.

As with many man-made objects, the sprocket consists of geometric primitives, primarily cylinders. We start modeling by introducing a series of cylinders (Figure 4.7, b). For each cylinder we first measure its diameter, then the height, confirming each value with the built-in button. In this *cylinder mode*, SPATA builds one cylinder on top of another, resulting in a configuration depicted in Figure 4.7, b. Note that as long as we are creating cylinders, we need not switch back to the design environment, thus saving two context-switches per cylinder.

Next, we add the two inner holes. In the design environment, we select the *cylinder mode*, only this time we use it to cut out the cylinders, instead of adding them. Measuring first the diameter, then height of each of the two holes, yields our second intermediary model shown in Figure 4.7, c.

In order to add the gear teeth, we first create a single tooth which is then replicated around the gear. Each tooth is a regular cube. Using the SPATA calipers we specify its width, height and depth, without needing to switch back to the design system, fully describes the shape (see Figure 4.7, d). We count the number of teeth, assume a uniform distribution of teeth around the gear, and use the design environments *circular pattern* function to add the according number of teeth, creating the final replacement part (Figure 4.7, e).

Modeling the replacement sprocket required a total of 12 sizes to be measured of the physical object. Using analog calipers, we need to not only manually type in all measurements, but also change repeatedly change context to do so (23 times if one measures the height of the sprocket teeth in CAD). The SPATA tools automatically transfer the measurements, and support creating the primitives that make up the sprocket. This way, we have to refocus our attention fewer times (6 times) and can perform the task more efficiently.

4.3.2 Sculpting

In this scenario we want to create a flower vase which will be 3D printed. To model that vase, we use a mesh-based design environment that supports vertex-based modeling, sculpting and constructive solid geometry. For artistic modeling often pen input is used instead of a mouse; we follow this practice.

We start the design process by creating a new cylinder. The vase needs to be correctly sized so that flowers fit in it and that it can be placed on a desk. Using the SPATA calipers and their ability to globally scale (by spreading the caliper jaws), we scale the cylinder until it is eight centimeters high. Using local scaling, we scale the diameter of the vase to 4 centimeters (see Figure 4.8, a).

Next, we add the decorative features by drawing on the cylinder using the pen. We use the SPATA calipers, which we now hold in our non-dominant hand, to rotate the model so that we can draw on all sides (see Figure 4.8, b). This way we do not have to change the mode from drawing to rotating, but use the pen to draw, the SPATA tool to rotate.

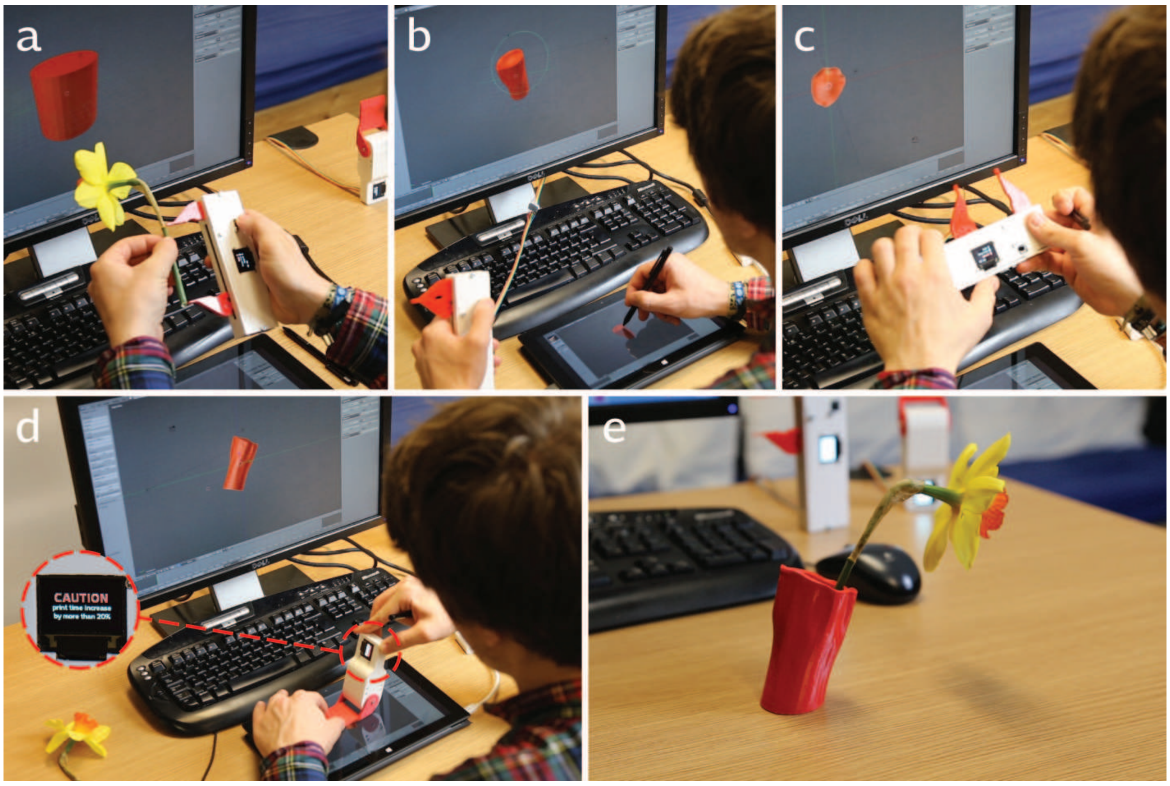


Fig. 4.8 Creating a vase. (a) We start with a cylinder, scaled to eight centimeters using the SPATA calipers. (b) We sculpt decorative features, using the SPATA tools for orientation. (c) Checking the size of the model. (d) Exploring different flower hole angles, resulting in a print time warning. (e) The printed result object.

Sculpting the shape has changed its size as well. Using Blender's built in measurement tool, we measure the vase model. This causes the SPATA calipers to output that size in the physical world (see Figure 4.8, c). This way we can compare the size against the flower, or get a feeling for the dimensions of the vase we are creating.

To make the vase more interesting, we want it to stand slightly angled. To explore different angles, we use the SPATA protractor. During this exploration our focus is on the SPATA tool, which gives us additional, fabrication specific feedback. When we use a too steep angle, we will be warned when the current angle will make the fabrication take longer[2] and be more expensive (see Figure 4.8, d).

Lastly, we cut off the bottom to create a flat surface for the vase to stand on and add the flower hole similarly to the holes in the sprocket example. We then send it to a 3D printer. The resulting vase fits the flower as designed and does not need support structures

[2]When using the protractor to angle a part, SPATA continuously recomputes the fabrication time by simulating the fabrication toolpath.

to print (see Figure 4.8, e), as we made design decisions to avoid that (SPATA warned us accordingly, see previous paragraph).

4.3.3 Desktop Organizer

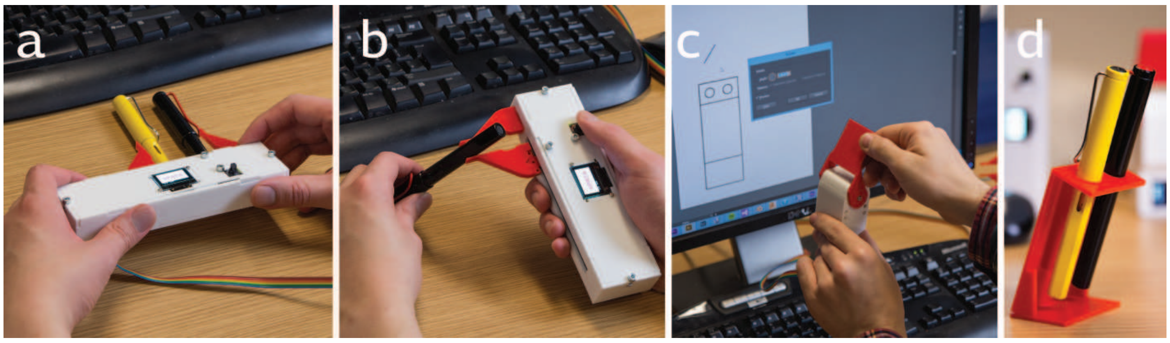


Fig. 4.9 (a) Measuring the width of the pen holder using the pens it will hold. (b) Measuring the pen holders height. (c) The drawing used for laser-cutting the parts of the holder. (d) The final object.

This scenario demonstrates SPATA's integration into a 2D design environment, which are often used for laser-cutting. Here, we create a simple, laser-cut pen holder starting with the top face of the pen holder. After starting to draw a rectangle, we use the SPATA calipers to measure the pen holders width and height (see Figure 4.9, a). The button on SPATA tools can be used to confirm and navigate between measurement axis (width and height). Next, we create the back piece. Its width is determined by the width of the top face, but the height is measured using SPATA. We then select height as dimension we wish to scale and measure the height of the pen. This sets the height of the pen holder to the height of the pen. Then we create circular cutouts much like we created the first rectangle: measuring the diameter of the pen yields holes of correct size (see Figure 4.9, b).

We use the protractor to explore which angle we want the pen holder to be at. After enabling rotation, the protractor assumes the current orientation: zero degrees in this case. Manipulating the protractor blade rotates the line on the screen accordingly (see Figure 4.9, c). We confirm the rotation using the build-in button. Finally, we laser-cut the drawing and assemble the pieces using acrylic cement, yielding our final object (Figure 4.9, d).

4.4 Discussion

The SPATA concept, bi-directional measurement transfer and integration into design environments, generalizes to tools other than calipers and protractors. On different scales, different tools are used. HandSCAPE [136], the digital measurement tape for example, could also output its value using an additional motor. Alternatively, a folding ruler could be augmented to support input and output of not only length, but also angle along its joints.

Physical features besides length and angle could also be considered. For example, an integrated measurement tool for material stiffness could be used to design multi-material 3D printed objects. Techniques such as jamSheets [137] could serve as output technology. Similarly, tools for transferring elasticity or weight and volume could be built (e.g., using technology presented by Niiyama et al. [138]).

4.4.1 Limitations

Our current implementation is design to demonstrate the SPATA concept. As such, the concrete devices lack features which would make them suitable for daily use. Our current implementation is not precise enough to be used in practice. The calipers have a measurement error of ± 0.5 mm, the protractor has a measurement error of less than one degree. While these tolerances are low enough to demonstrate the SPATA concept, they prevent real-world use. Adapting a more traditional caliper/protractor design would likely alleviate these issues, but would make the actuation more difficult to implement. Further, our tools do not have a vernier scale, but display their state digitally. This requires power to be supplied to the devices for them to be useful.

SPATA is tightly integrated into digital design environments, thus their use is primarily beneficial in a digital design process. When measuring things in an analog setting, traditional tools are preferable over our prototype implementation. SPATA tools are tethered to a computer, restricting the environments they can be used in. Additionally, in their current iteration, the tools are not as precise as their analog counterparts. This prevents them from being used to very small parts. Further, traditional calipers have a thin depth probe to measure the depth of cavities and holes. The current implementation of SPATA calipers does not have such a depth probe, rendering depth measurements difficult.

4.4.2 Implementing the Integration

While building the tools, we implemented their integration into Autodesk Inventor, Blender and Adobe Illustrator. We found it beneficial to contain all logic in an integration middleware layer. Many software packages support an API to extend their functionality (e.g., SketchUp has a Ruby based plugin mechanism). We further kept both SPATA tools free of design environment specific artifacts, to avoid firmware changes. Through this clear separation of concerns into loosely coupled components (tools, environments, logic-containing middleware), we lower the effort to integrate new tools or design environments. By providing an API for our tools, we would enable others to integrate them into their applications. When developing other tools based on the SPATA concept, we recommend a similar approach to enable quick prototyping and exploration.

4.4.3 Customization

We have implemented tasks commonly found in their respective environments (e.g., creating a box in parametric modeling environments). However, specialists often customize their environments to better support their work. A macro editor (or other forms of end-user programming) would enable users to create their own workflows or tasks that integrate the SPATA tools. Custom jaws and blades for specific applications could also be built. For example using specialized task support and adapted tools that align well with human physique, doctors could quickly model a splint for their patient.

4.4.4 Generalizing Spatial Understanding

The physical rendition of virtual models, provided by SPATA, is limited to dimensions and angles. As we have demonstrated in the walkthroughs (e.g., section 4.3) this is of some value. It enables users to review individual design decisions, even in their target context (e.g., by comparing a previously chosen length to a users hand). However, how many of such singular physical representations are required for users to gain a full spatial understanding of the object-under-design is an open question. Similarly, do we not know much about the strategies that designers would employ towards that end. Would users adopt a top-down approach where they output the larger dimensions first, followed by important details? Our prototype implementation would help answer such questions through controlled lab experiments.

In the next two chapters, we describe concepts that provide physical renditions of the object-under-design, one through augmented reality (chapter 5), one through digital fabrication (chapter 6). Both concepts and systems thus offer alternatives to this physical representation as single-dimension. Later, we discuss how those systems could be combined (see Section 7.2.3).

4.4.5 Non-Fabrication Scenarios

The need for integrating spatial features extends beyond design for fabrication. In computer supported collaborative work (CSCW), or whenever there is a spatial/temporal division between users, SPATA could be used to transfer spatial features. For example, two spatially disconnected users could exchange the screen-size of the new tablet they've bought. In a temporally disconnected scenario, users could get an impression of the size of an object offered in an online store, or measure parts of their body to order a custom-made artifact.

4.5 Summary

In this chapter we presented *Active Spatio-Tangible Measurement Tools* that integrate physical measurements into virtual environments, and vice versa. With the SPATA tools, users can take measurements in physical space and immediately apply them to digital objects and dimensions (P3). We thus remove the need to manually transport the measured value (i.e., read it off an analog tool and manually type it in). This bridge between both spaces is bidirectional: lengths and angles in the virtual environment can be automatically transported into physical space. Thus, we help designers understand the size of the object-under-design (P2), as said size is rendered physically. It can also be used to reflect on design decisions in context (P4). By using these tools as physical props for the object-under-design, we further enable easy navigation inside the virtual environment (P1). Additionally, as SPATA tools are used to make design decisions, they communicate fabrication-specific aspects to the user (e.g., when tilting an object and a threshold is crossed so that support is needed, that is displayed on the protractor), integrating fabrication-specific knowledge (P6).

We contribute two such SPATA tools: a digital adaptation of calipers and of a bevel protractor. In this chapter we have described their design and implementation, as well as their integration into three different virtual design environments: parametric CAD, mesh-based modeling, and 2D design tools. Through our implementation, which we used in three walkthroughs, we have demonstrated that *Active Spatio-Tangible Measurement Tools* can make fabrication-aware design processes more efficient and convenient.

In the next chapter we move another step closer to the physical world. We situate the design environment in a mixed-reality space where virtual and physical artifacts co-exist.

MixFab
Mixed-Reality Design for Digital Fabrication

Digital fabrication design environments would become accessible for novices if they were situated closer to physical space. In previous chapters we have explored two concepts that connect the physical world and digital design environments, while these evironments remained entirely virtual. Thus we still use mouse and keyboard for the majority of interaction with those virtual design tools. For example, interaction with general-purpose CAD software remains difficult as users still have to learn how to create and manipulate objects (P1). While, in previous chapters, we have shown ways to alleviate problems revolving around physical objects in virtual environments, some issues persist. Existing objects still find little representation as they remain in a separate space (P3), and we lack direct engagement with the object-under-design (P5).

In this chapter, we explore and develop the *mixed-reality design for digital fabrication* concept which breaks free of flat computer screens and creates a mixed-reality space where the physical and digital coexist. In such a space, users can directly – through gestures – interact with virtual objects, including the one they are designing. This eases manipulation and interaction with the design environment (P1) as users can reach into the virtual space and move things as they would in physical space. Due to the co-location of the physical and virtual realm, existing physical objects can interact with virtual ones. We can use an existing physical artifact and compare its size with the item we are designing e.g., to see if the latter is big enough. Further, we can capture the shape of an existing thing and reuse it in our

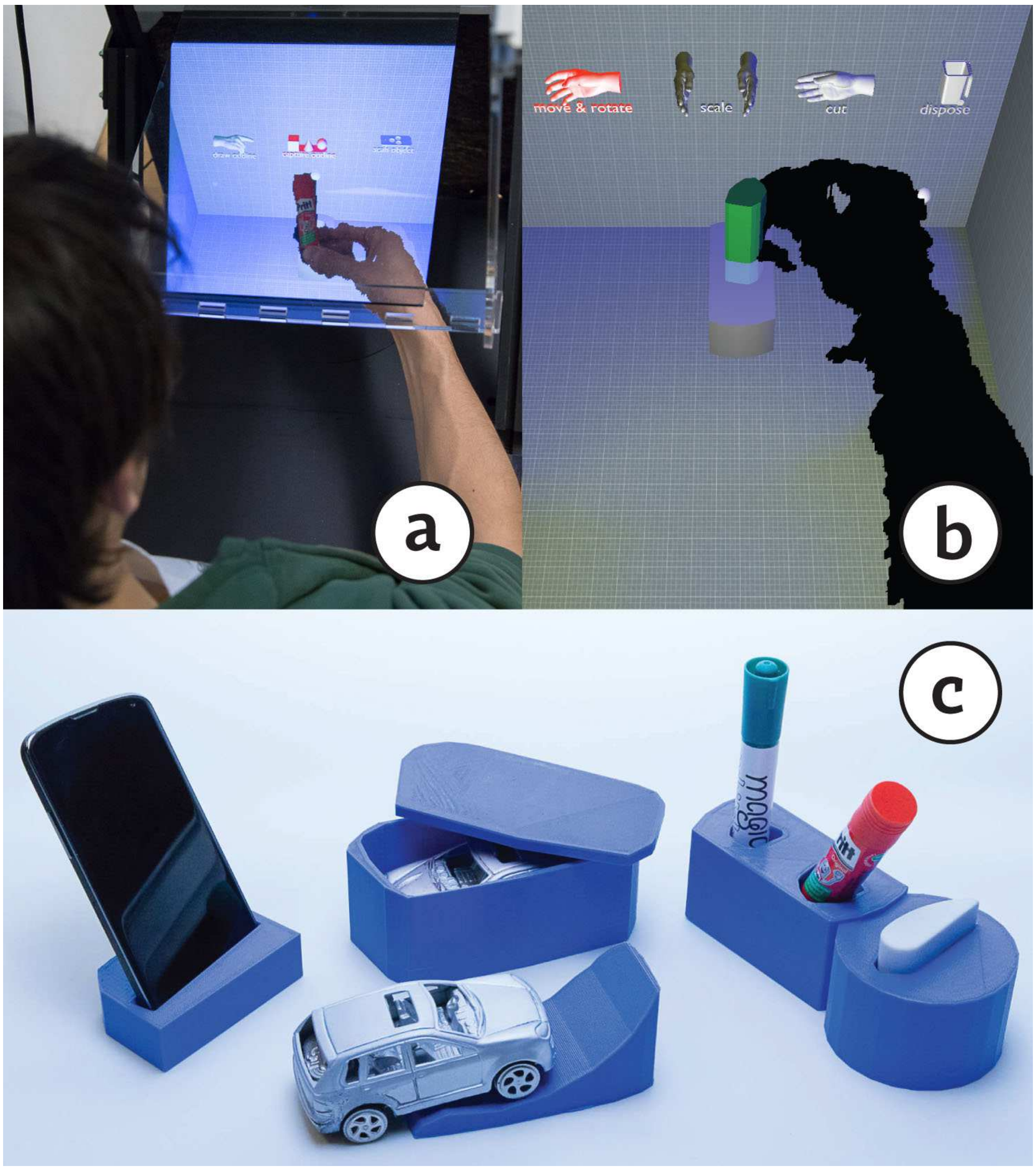


Fig. 5.1 MixFab: mixed-reality design for digital fabrication. (a) a user positioning a physical object in the MixFab prototype system, (b) screenshot of a user manipulating a virtual object, (c) 3D printed objects created with the system.

design. Combined with a constructive modeling paradigm (Constructive Solid Geometry (CSG), see Section 2.3.1), that works by adding or removing material, mixed-reality design enables novices to create meaningful objects using digital fabrication. In summary, *mixed-reality design for digital fabrication* integrates three core concepts: (1) use of immersive augmented reality to provide a 3D visualization of the object-under-design projected in the real world; (2) support for users to shape artifacts directly with their hands, replacing the need for advanced modeling skills with intuitive gestures; (3) enabling use of real artifacts in the design process such that new artifacts can be shaped to fit existing ones.

In the following we present a system that implements such a mixed-reality design environment: *MixFab*. The mixed-reality space is by virtue of a Holodesk-like structure [139] where the user sees virtual content merged with the real world. Users can introduce physical artifacts as size-reference or to capture their shape – Figure 5.1, a shows a user placing a glue-stick inside a virtual object to create the glue-stick's virtual replica in place. In Figure 5.1, b the just created glue-stick replica (green object) is manipulated as if it was still a physical entity; by virtue of gesture recognition, users can directly manipulate virtual and physical objects alike. Hands and other physical artifacts properly occlude other objects and face-tracking provides a parallax-corrected image, creating important depth-cues.

We make four contributions in this chapter. First we propose and implement an immersive mixed-reality environment by combining an augmented reality setup, gesture recognition and 3D scanning capabilities. Our second contribution is a set of user-defined gestures for 3D modeling obtained through a study in which we observe how users would perform basic tasks unconstrained by any system or augmentation. We then present MixFab's design environment, which is based on these gestures. It is centered around direct and natural interaction with virtual artifacts, effortless integration of physical objects into the design process and a self-explanatory interface. Fourth, we evaluate MixFab's design decisions in a user study and provide evidence that, in particular, the effortless integration of existing physical objects is of value.

In the following, we first give an overview of the system, followed by a description of its UI. We then describe the implementation of the MixFab prototype implementation, and the user-defined gesture set the system builds upon. The prototype and *mixed-reality design for digital fabrication* concept are subsequently evaluated in a user-study. We end with a discussion and summary of the presented work.

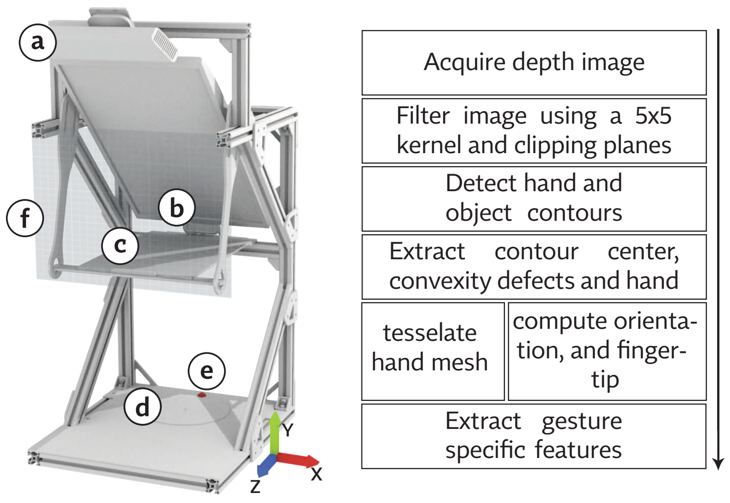

a
f
b
c
e
d
Y
X
Z
Acquire depth image
Filter image using a 5x5 kernel and clipping planes
Detect hand and object contours
Extract contour center, convexity defects and hand
tesselate hand mesh
compute orientation, and fingertip
Extract gesture specific features

Fig. 5.2 [left] The MixFab hardware (a) Microsoft Kinect depth sensor, (b) webcam used for face tracking, (c) 50T/50R half-mirror, (d) motorized turntable/system floor plane, (e) common world origin (virtual), (f) face-tracker calibration plane (virtual). [right] MixFabs processing pipeline

5.1 System Overview

At MixFab's core is an immersive mixed-reality system creating a high permeability between the virtual and physical world. It enables new and exciting interactions that were not possible with each component taken by itself.

We implemented MixFabs physical configuration (see Figure 5.2, left) by building upon the *Holodesk* frame [139], although other hardware implementations may also be used e.g., *MirageTable* [140]. The setup superimposes virtual content with the real world using a beam-splitter and a display mounted at a 45 degree angle. It provides an interaction volume roughly the size of modern 3D printers. A depth camera placed at the top of the frame provides data for interaction within the system. We integrate a motorized turntable for 3D scanning at the frames bottom.

Our processing pipeline is designed to specifically support seamless interaction between virtual and physical objects, blurring the border between the two. The main components of this pipeline are:

Gesture Recognition which is solely based on the depth-data provided by the Kinect serves as input modality for the interaction with virtual objects. It does not require any user-augmentation or prior calibration.

3D Shape Acquisition is supported at a trade-off between time and precision. One can

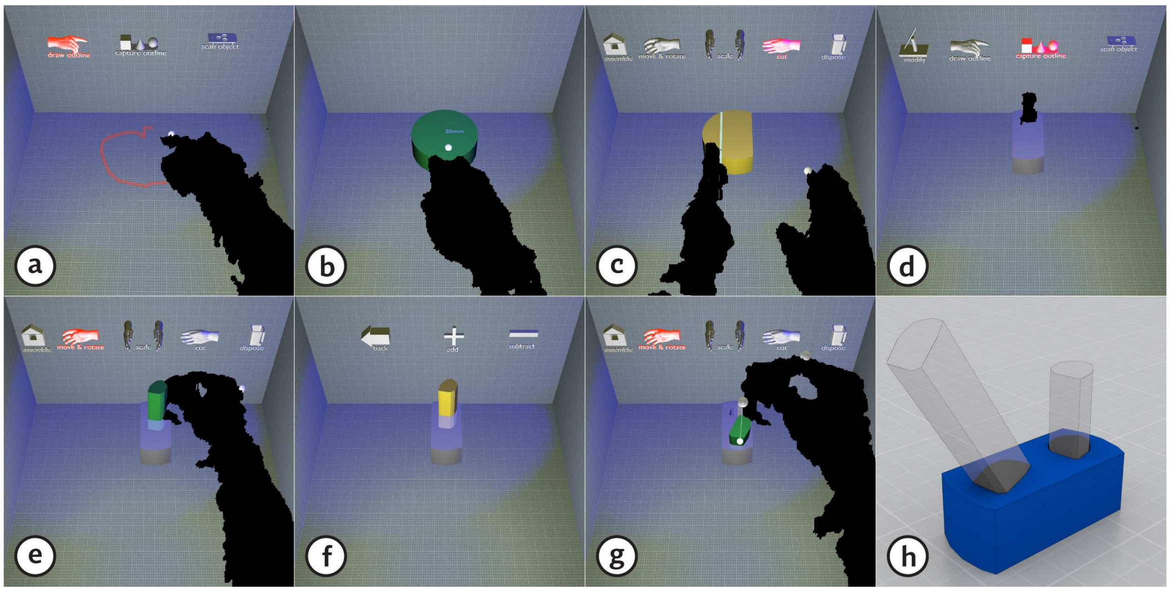


Fig. 5.3 MixFabs user interface: (a) a user drawing an object's outline, (b) setting the height of the cylinder, (c) plane cuts of the cylinder, (d) capturing the shape of a physical object (glue-stick) positioned in the virtual one, (e) moving an object (glue-stick) upwards by grabbing it with one hands, (f) object assembly (difference of glue-stick with existing virtual shape), (g) rotating an object (pen), (h) the desktop organizer (blue)

capture the rough shape of an object in real-time or acquire a more precise scan in about a minute. Physical objects can be captured anywhere in the frame, allowing their placement relative to virtual objects.

Sketch Recognition enables users to describe objects they want to create without having to be very precise.

Mesh data manipulation serves as back-end for object creation, acquisition and manipulation. We support complex operations (e.g. constrained Delaunay triangulation or plane segmentation) required for object acquisition, and *constructive-solid geometry* operations for shape manipulation which produce 3D printable models.

5.2 User Interface

MixFabs user interface is centered around the gestural creation, modification and assembly of objects. Using gestures to describe and modify shape has also been explored in other work. Some let users deform models using both hands [141, 142]; others use the motion of the hand [71] or its curvature [143] to define the shape. We build on a symbolic gesture set which we develop through a user-study in section 5.4.

The user mainly interacts with "gestural icons" and the virtual objects being created and assembled (Figure 5.3). Gestural icons depict a certain hand pose, showing the user what gesture to perform to trigger a certain action or change a certain property. All hand icons translate to gestural input (e.g. draw outline), and all non-hand icons translate to automation (e.g. scan object). To perform an action, users first select the appropriate icon and then either perform the gesture (e.g. scales the object) or wait for the system to complete its task. The icons are context-sensitive, hence inform users about currently available operations.

Virtual objects displayed in MixFab can have three different states. Inert objects that cannot be modified without selecting a gestural icon, are colored in a slightly transparent gray (inert state). Once an object becomes modifiable using a gesture, it turns yellow (inactive state). Objects that are currently being modified, are colored green (active state). This color-coding provides feedback about the current system state, especially the grasping of objects. It allows users to determine whether the system recognizes them as engaged in a gesture and what influence their movement will have on the scene.

5.2.1 Creating objects

There are four ways users can create objects: drawing an outline, having the system capture the outline of a physical artifact, 3D scan an existing object or load an existing 3D model.

Users can draw the outline of primitive shapes on the system floor (also referred to as system ground, see Figure 5.2, d), using only their index finger. The system then recognizes the sketch as either a circle or rectangle and extrudes it to 3D space – thus in this manner users can initially create boxes and cylinders. The height of the object is set by the height of the hand above the systems ground (in discrete 5 mm increments, to make this operation more accurate; for a discussion see Section 5.6). Once the height is as desired, the other hand taps the floor to fix the height.

Another way to create the initial 3D model, is to capture the 2D outline of an existing physical object, and to extrude it to 3D space. This offers a simple but fast method to capture an object's shape. Users can place existing objects anywhere in the frame and after a fixed dwell time, the system captures the outline as seen from above (in XY plane, see Figure 5.2 for the coordinate system) and automatically extrudes it to the physical object's height. Users can manipulate the object's height, by indicating the height with one hand and confirming it with the other.

The MixFab frame has a turntable built in which serves to rotate objects so that objects of more complex shapes can be scanned. To scan an object, the user selects the "scan object" icon, and places the physical object together with the scanning rig (see Section 5.3.4) on the turntable (Figure 5.4). The system then waits until all hands are out of the frame before it starts rotating the object to capture it from all sides. Once scanning is complete, the virtual object appears where the existing one was placed.

CAD drawn models can incorporate functional aspects or higher-resolution details than what the built-in 3D scanner is able to capture. When the user selects the “load model” icon, a grid of scaled-down models is provided, showing models loaded from a pre-defined folder, from which the user can select the desired model (by selecting icons). The new object is then placed in the center of the frame.

5.2.2 Manipulating objects

Once an object has been created, it can be manipulated in three ways: translation & rotation, scaling and removing parts of the object.

Translation and rotation is performed using a one-handed grabbing gesture, much like one would grab a cup. If the object is grabbed so that the hand intersects the object, it attaches to the user’s hand so that the object can be moved freely within the interaction volume. Grabbing any point away from the object lets users change the object’s orientation. A lever is formed between the base of the object and the hand-tip, which is then used to rotate the object. Translation and rotation both snap to common values (e.g. the floor for translation and 0/90 degrees for rotation).

Objects can be uniformly scaled using a two-handed compression gesture, as defined by our user-defined gesture set (see Section 5.4.3). Users place their hands on either side and the scaling factor is a function of their distance. We implemented relative, yet direct scaling using a fixed control-display gain. When users first assume the compression posture, their hand distance is identified as 100% scale. Changing the distance between both hands then scales the object.

Cutting objects removes material, rather than splitting objects. To perform a cut, the user indicates the desired position of the cut using their flat hand (“Shuto”/Karate gesture) along the X-axis. Tapping on the ground with the other hand confirms the cut. If the user indicates the cutting position with the right hand, the right side of the cutting plane is discarded; indicating with the left hand removes the left side.

5.2.3 Assembling objects

Object assembly combines two objects, either by adding them together or by subtracting one from the other. Fusing two objects can be used to add material or refine the shape of an object. Subtracting one object from the other is commonly used to create holes or cavities to hold other objects. There is no specific gesture for assembly. Object assembly is simply a matter of selecting the way the two objects are to be combined, using MixFab's gestural icons. Union or difference of meshes are symbolized with a plus or a minus sign respectively (Figure 5.3, f).

5.2.4 Walkthrough: constructing a desk organizer

We illustrate the systems use by constructing a desk organizer (Figure 5.3, h) that will hold a pen and a glue-stick.

We start with creating the base shape by drawing a circular outline (Figure 5.3, a). The system recognizes the drawing as a circle and offers the outline for extrusion. We set the height using one hand; the height snaps to 5 mm increments and is displayed just above the object. To confirm the height, we tap with the other hand (Figure 5.3, b).

To create the semi-rectangular shape of the organizer, we cut off both sides. First we cut off the right side of the object by indicating the cut position with the right hand and confirming with the left by touching the system floor. To cut the left side, we repeat the procedure, this time holding the left hand where we want to cut and confirming with the right one (Figure 5.3, c).

Next, we create the first hole which will hold the glue-stick. We position the physical glue-stick where we want the hole to be within the virtual object. Once in position, we select "capture outline" and move our hands out of the frame (Figure 5.3, d). The system then captures the outline of the glue-stick and extrudes its height. Confirming that initial height with the left hand, turns it into a virtual glue-stick replica.

As the glue-stick was standing on the ground of the frame, the virtual glue-stick replica is on the ground as well. If we were to assemble the object as is, we would create a hole through the whole object-under-design. To ensure there is material at the bottom of the hole, we grab the virtual glue-stick with one hand, move it a few millimeters up and release it to fix its position (Figure 5.3, e).

Eventually we assemble the virtual glue-stick and the previously created base to create the hole for the stick. After selecting assembly, we are asked to choose the method of assembly: add or subtract (Figure 5.3, f). Choosing subtract removes material where the glue-stick was, leaving a hole of correct size and position.

Lastly we repeat the steps above for the pen, placing it in its desired position, capturing its outline, extruding it and moving it up a few millimeters. To make the pen easier to access, we tilt it forward by grabbing at a point in space, forming a lever with which the object is re-oriented (Figure 5.3, g). Once in correct position and orientation, we assemble the virtual pen replica resulting in the final desktop organizer (Figure 5.3, h).

5.3 Implementation

We built a prototype to implement the MixFab system by using *Holodesk's* hardware frame [139]. Holodesk provides an immersive environment with an interaction volume roughly the size of modern 3D printers. Our hardware differs in that we use a Kinect for Windows rather than a Kinect360 and mirror setup, and have a turntable built into the frame for 3D scanning. Most importantly, on the software side, we employ a different processing pipeline and provide a gesture-based interface rather than a physics-based one.

5.3.1 System Hardware

The hardware consists of a display mounted at a 45 degree angle, being reflected through a 50/50 half-mirror into the interaction space. A Microsoft Kinect depth sensor mounted at the top of the frame is used for capturing the interaction with the system, while a second camera placed between the display and half-mirror is used to implement perspective correction through face tracking. A motorized turntable for 3D scanning is built into the floor of the frame (Figure 5.2).

Calibration

Two cameras need to be calibrated once (not per user): the facetracker and the Kinect. The facetracking camera is calibrated to a plane perpendicular to the half-mirror (Figure 5.2, f). As the dimensions of the frame are known, the exact position and orientation of the facetracking camera (and thus faces tracked) can be mapped to real-world coordinates. The Kinect is calibrated using its RGB camera by placing a laser-cut 8×6 checkerboard calibration pattern at a fixed location on the bottom of the frame. Using an approximation of the depth-to-RGB transformation, we can map depth data to the real world coordinate system with respect to a common origin (Figure 5.2, e).

5.3.2 General Processing and Gesture Recognition

Gesture recognition often relies on user augmentation, particular when only a single depth camera is available[1]. Surface Drawing utilizes gloves [71], others use reflective markers [144]. Oka et al. [145] require only a camera in the environment and no user augmentation. For an overview of hand pose recognition techniques, see [146]. We implemented an appearance based approach using a single depth-camera, requiring no user-worn equipment or prior calibration.

MixFab follows an appearance based approach to hand posture and gesture recognition that requires no prior calibration or user augmentation. Depth data from the Kinect is processed to extract a set of features which is later used by specialized gesture recognizers. We rely solely on the depth image, as the half-mirror occludes the hands in the color image.

The general processing pipeline is as follows: first we acquire a depth frame from the Kinect, filter it using a 5x5 convolution kernel, remove points using previously defined clipping planes and tessellate the remaining points to generate an occlusion mesh; all of which is implemented in OpenCL[2]. Then, in the depth image, we find all connected components touching the image border, with an area A greater than A_{hand}; these are hand contours. For each such contour, we compute its center, orientation via the Hu moments and finger-tip which is the convexity defect farthest along the principal axis.

Touching the floor

To implement touch input on the floor of the frame, we threshold the Y component of the finger-tip (vertical finger height) [147]. As the floor surface is flat and we map the Kinect depth-data to real-world coordinates with an origin in the floor plane, we set the parameters introduced by Wilson [147] to $d_{surface} = d_{max} = 0$ and $d_{min} = 20$mm.

Sketch recognition

Sketches drawn on the floor are represented as 2D polygons $p_1, \dots, p_N$ which we simplify using the Douglas-Peucker algorithm [148] and an edge-angle based filter (all adjacent edges p_i, p_{i+1} with an enclosing angle $\alpha = p_i \angle\, p_{i+1}$ less than α_{join} are joined). We consider the sketch to be a rectangle if its average segment length $\varnothing_L = \frac{1}{N}\sum_{i=1}^{N-1} |p_i - p_{i+1}|$ is less than $\varnothing_L^{rect}$. A circle has its center at the center of the polygon and its radius is the

[1]Unless the depth camera is placed in front of the user. In that case segmenting the users hands is commonly done with a heuristic: find the first point in front of the camera, and consider all points up to 10 cm back. In MixFab, the depth camera sees the users hands from above.

[2]https://www.khronos.org/opencl/

average distance of each point to that center. Rectangles are the bounding rectangle of the polygon.

Open hands and Grabbing

Open hands (Figure 5.8, c) pointing along the Z-axis result in a local minimum of the hand contour's arc length. Thus, we can recognize an "open hand" posture if the arc-length L of the hand contour (within a fixed distance from the hand tip) falls below L_{max}.

A not-grabbing hand forms several convexity defects [149]. If a defect with an angle greater than α_{min} and depth greater than d_{open} is found, the hand is considered to be in an open state. If no such defect is found, the hand is considered to be in a grabbing state. While maintaining a grabbing pose, the finger-tip detection heuristic does not work reliably. The grabbing pose involves bending the hand compared to the arm, causing the finger-tip to move away too far from the previously computed orientation axis. The real tip of the hand and origin of the hand contour form another convexity defect, which is stable when the correct posture is maintained. Kalmann filtering yields a usable hand-tip estimation.

Wiping

When performing the wiping gesture, users move their flat hand from one side of the frame to other in a speedy fashion. To detect that gesture, we continuously sample the contour centers X component with a fixed window size w (corresponding to the time in which the gesture has to be performed). If all points in that window are equidistant, their distances d monotonously inc-/decreasing and the start/end points are at least d_{width} apart, a wipe gesture was performed.

5.3.3 Contour capturing

MixFab's processing pipeline distinguishes between hand and object contours, if they are not connected (see Section 5.3.2). All contours are subject to perspective distortion, which we correct using the previously acquired calibration. To capture the outline of an object, we build the convex hull of all object contours in the frame. Thus, objects can be grouped and produce a smooth shape from the noisy Kinect data, but we also slightly reduce precision. We find the highest point within the hull, making that the initial extrusion height. This method is fast (it requires only one depth image), but offers only an approximation of the physical object's shape. 3D scanning (see next section) provides a more detailed capturing process.

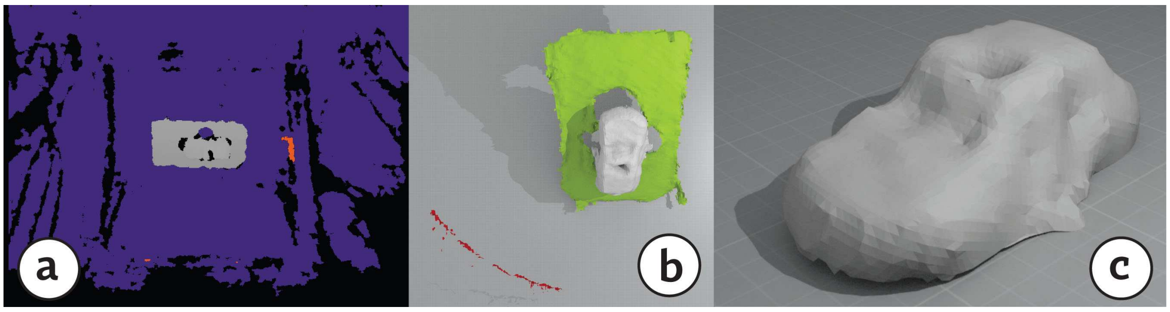


Fig. 5.4 The object scanning process. *(a)* The Kinect depth data (white) is filtered using clipping planes (purple), leaving some noise (red), *(b)* the scanned object (white) contains that noise (red). Removing noise and scanning table (green) and closing remaining holes results in the scanned object *(c)* .

5.3.4 Object scanning

3D scanning in MixFab uses Kinect Fusion [91] and a custom built turntable/scanning rig. Kinect Fusion estimates the camera to world coordinates using the *iterative closest point* (ICP) algorithm [150]. ICP implicitly requires geometric prominent features to converge, resulting in a poor scanning performance on “uninteresting” scenes. Normally Kinect Fusion is used on a room scale, with enough clutter so that the ICP based camera tracking works well. In MixFab however, we scan single objects only, resulting in severe alignment errors without our scanning rig. The latter is designed with clear geometrical features (prominent edges and corners, extreme width to height ratio) which aids Kinect Fusion in producing its camera alignment. After filtering the depth image using clipping planes, there is still a some degree of noise left (Figure 5.4, a). Due to the sparse nature of the images produced by the clipping, noise has a drastic impact on the camera tracking and thus scanning performance.

The scanning process begins with integrating plane-clipped point clouds into the internal 3D scene representation maintained by Kinect Fusion. Once the object has been captured from all sides, this representation is transformed to a mesh 3D model. All unconnected components with a face count f less than f_{noise} are removed. The scanning table surface is found by computing the largest connected component using a threshold of the discrete RMS curvature [151] as connectivity condition. We then fit a plane to the scanning table vertices and rotate the mesh, so that the scanning table is in the XZ plane. All vertices and faces with a distance to the scanning table d_{tab} which is less than d_{tab}^{cut} are removed. We remove unconnected components with a face count f of less that $\frac{1}{2}f_{noise}$ faces, as well as non-manifold vertices/faces and fill gaps less than L_{hole} units in arc length.

The resulting mesh is likely to contain larger, still unfilled, holes – at least one from cutting away the scanning table. We smooth the mesh (thus hole boundaries) using Laplacian smoothing [152]. For each remaining hole, we fit a plane to the boundary vertices, project those vertices to that plane and compute a Constrained Delaunay triangulation (CDT), hence closing the hole. In a last step, we remove non-manifold vertices/edges created by the CDT and fill the resulting gaps (Figure 5.4, c).

The mesh processing pipeline is implemented using the *Visualization and Computer Graphics Library*[3], *sgCore*[4] and *qHull* [153]. It takes 30 seconds to complete one revolution of the turntable and less than 10 seconds to perform the mesh processing.

5.4 User study: User-defined 3D modeling gestures

In previous work, hand gestures are typically defined by the respective authors, rather than users [141, 143, 144, 154, 155]. We are interested in what gestures users would intuitively perform to create and manipulate objects, also to inform our subsequent system design. To this end we conduct a user-defined gesture study with a methodology similar to Wobbrock et al. [156]. For a list of all gestures elicited from users (which were performed by at least two users), see Table 5.1.

5.4.1 Tasks and Procedure

Each participant was subsequently given a set of tasks (order determined using a balanced latin square): create box, create cylinder, move box, rotate box, one degree-of-freedom scale, uniform scale, plane cut, add material, scan object and remove object. For each task, they were shown one or two printouts depicting the desired outcome and asked to perform a gesture to create that desired outcome. A more detailed description, and the images shown to participants, are given in the next section. Participants were instructed to imagine the objects depicted on the images as being displayed in front of them.

After each gesture, users were asked to rate the gestures suitability and how easy it was to perform, both on a rating scale from 1 (very unsuitable / very hard) to 5 (very suitable / very easy) . Once all ten tasks were completed, all users completed a survey querying their age and gender. We further asked for prior CAD experience and how much that experience influenced the proposed gestures (Rating scale, 1 no experience / no influence to 5 a lot of experience / strong influence). Users were seated at a table with a camera placed a meter above the surface, resulting in an interaction area of about 60×50 cm.

[3]http://vcg.isti.cnr.it/vcglib/
[4]http://www.geometros.com/

Upon completion of the study, the recorded video material was transcribed and coded to extract the suggested gestures. Quantitative data collected through the questionnaires, as well as user agreement [156] is used to judge the quality and confidence of users in the proposed gestures.

5.4.2 Participants

We invited twelve participants from various departments at our university. Half of the participants were female with age ranging from 19 to 42 (*M*=30.83 years, *SD*=7.47). We aimed to minimize the impact of existing CAD modeling experience, as we target novices with this system. However, we did not exclude participants with CAD experience. Eight users reported no experience with CAD (overall *M*=2.08, *SD*=1.31) and thus little influence of prior experience (*M*=2, *SD*=1.53). This experience distribution is a result of the convenience sampling of participants on our campus.

5.4.3 Results and Observations

All participants were able to propose a gesture for every task, sometimes more than one in which case they were asked to report the one they preferred. In the following we report gestures that were proposed by at least two or more participants (for the complete list of gestures, see Table 5.1).

Creating primitives

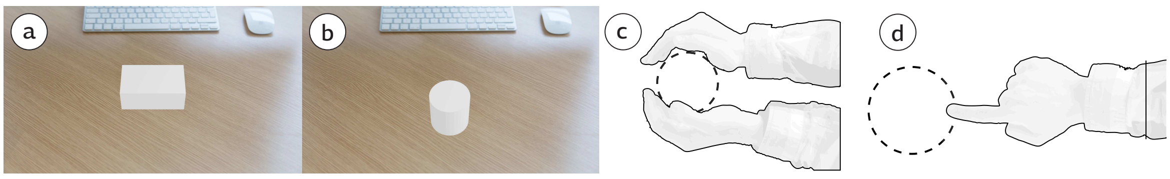


Fig. 5.5 (a, b) the pictures shown to study participants (create box/cylinder), (c) *hand boundaries* gesture, (d) *drawing outline* gesture

While creating primitives has no equivalent in the physical world, describing 3D shapes is a common task. When creating boxes, three users choose a method similar to *Data Miming* [143] and define the shape by describing it using their hands (hand boundaries gesture). In case of the cylinder, a majority (6 of 12) preferred to describe the shape using the curvature of their hands. Drawing the outline of the shape with the index finger and extruding it into 3D space was proposed by a majority (7 of 12) when creating boxes, and by a third of the participants suggested it for cylinders.

Rotating and translating

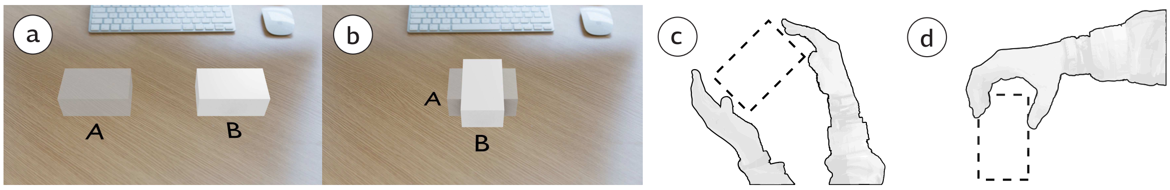


Fig. 5.6 (a, b) the pictures shown to study participants (Rotate/move box from A to B), (c) *two-handed grab and rotate* gesture, (d) *one-handed grab* gesture

Rotating and moving objects are everyday tasks performed using two variants of the same gestures: one-handed vs. two-handed. The pictures shown to the participants (see Figure 5.6) contained a keyboard and a mouse next to the virtual object, as a size reference. It seems that the size of virtual object was interpreted differently by each participant, leading to the use of one hand if the object ws perceived to be small or both hands if the object is perceived to be large. When asked to move the depicted object, 4 of 12 participants used one hand; another 4 of 12 used both hands (see table 5.1). For the rotation task, 7 of 12 used one hand, 4 of 12 used both.

Scaling

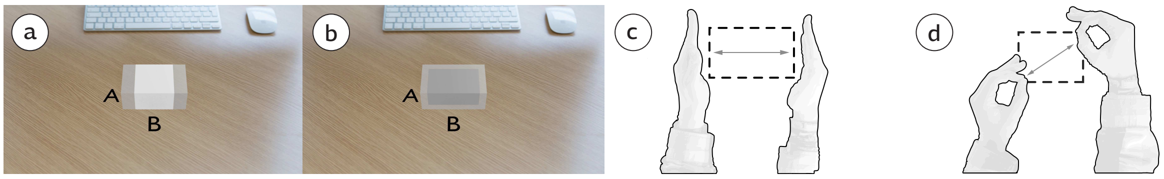


Fig. 5.7 (a, b) the pictures shown to study participants (scale 1DOF/3DOF from A to B), (c) *compression* gesture, (d) *corner pinch & resize* gesture

We presented two different varients of the scaling task: a uniform scaling task where all three dimensions are scaled at the same time, and a one degree-of-freedom scaling task where only one dimension is scaled (see Figure 5.7). For both tasks, 10 out of 12 users suggested the same gesture: compression. Both hands are placed around the object and changing their distance, changes the objects size.

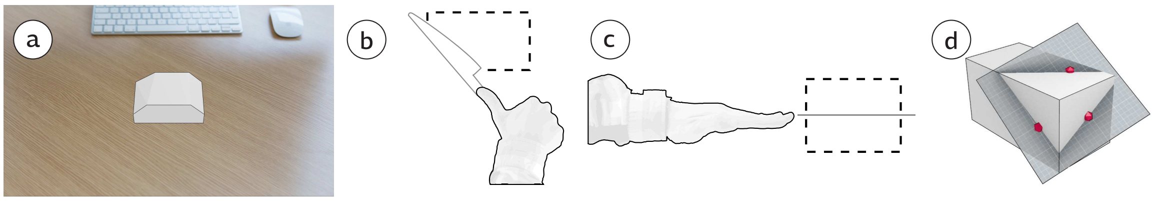


Fig. 5.8 (a) the picture shown to study participants (plane cut of shapes from box), (b) *knife* gesture, (c) *Karate* gesture, (d) three *points* defining a plane

Plane cut

Cutting a slice of an object is a daily task (e.g. cutting a slice of bread or cheese). Users without CAD experience proposed gestures resembling such cutting motions (see Figure 5.8). Four users moved their flat hand or thumb where they wanted to cut, miming a knife. Five users indicated how they wanted to cut by performing a "Shuto" (Knife Hand) motion from Karate. Those experienced in CAD, suggested that one might select three points on the object to define a plane used for cutting.

Adding material

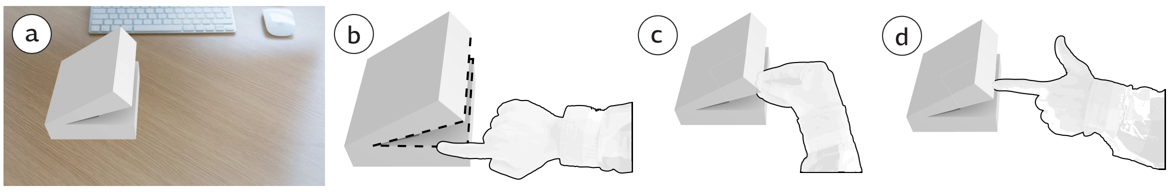


Fig. 5.9 (a) the picture shown to study participants (add material inside gap), (b) *gap trace* gesture, (c) *stuff(ing)* gesture, (d) *filling tool* gesture

Many participants (4 of 12) suggested a "stuffing motion" as if they were to take a handful of material and put it onto the object (see Figure 5.9). Others (also 4 of 12) traced the gap they wanted closed with the index finger and confirmed again using a stuffing gesture or by pressing a button. We believe that the "gap-trace" gesture is an artifact of the picture that demonstrated the task: the gap to close was of regular and linear nature. Suggestions would most likely be different when manipulating more organic shapes.

Scanning objects

Transforming physical objects to virtual ones proved to be the most challenging task; to some extent because it is difficult to convey the need of the operation without a system being present. Participants proposed a variety of actions (agreement score: 0.17), but only one was

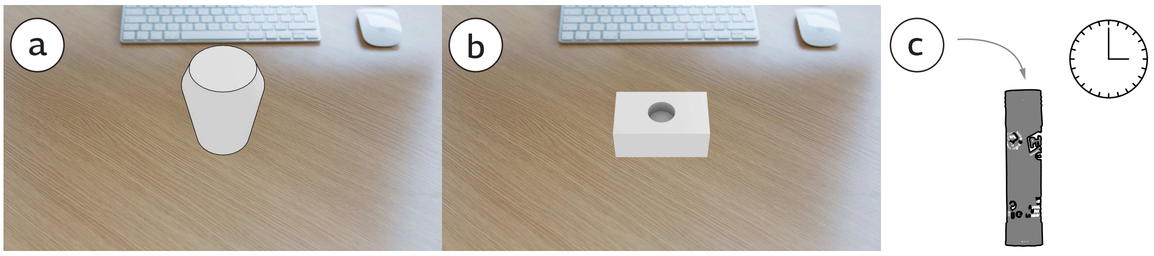


Fig. 5.10 (a, b) the picture shown to study participants (positive scan of existing cup, negative imprint of cup in box), (c) the *dwell time* action

mentioned multiple times: dwell time. Users place the object in the desired position, move their hands away and wait for a certain amount of time.

Removing objects

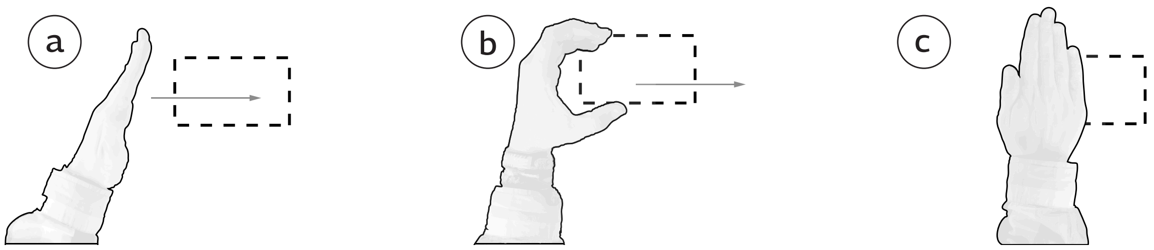


Fig. 5.11 (a) *wipe* gesture, (b) *move out* gesture, (c) *smash* gesture

Removing objects is a daily task. We often throw things away, place them elsewhere or deform them prior to disposal. The gestures suggested for removing objects tend to resemble such actions. *Wipe* and *move out* – the two most prominent gestures – have the same intent: move the object out of the workspace. Wiping objects (moving a hand fast from one side to the other, hitting the object) was suggested more than *move out* (four times compared to three times) and rated easier/more suitable.

5.4.4 Discussion

We observe a similar pattern as Wobbrock *et al.* [156] in that user agreement is inversely proportional to the task complexity (see table 5.1). More complex tasks (such as object scan) have low user agreement scores, whereas more simple ones (such as rotation or translation) yield higher agreement amongst users. Despite low agreement rates, suitability and easiness remain at high levels, suggesting confidence in the proposed gestures.

To choose an appropriate gestures for each task group, we use the count of how often a gesture was suggested as main metric. In cases where the suggestion count is not distinctive,

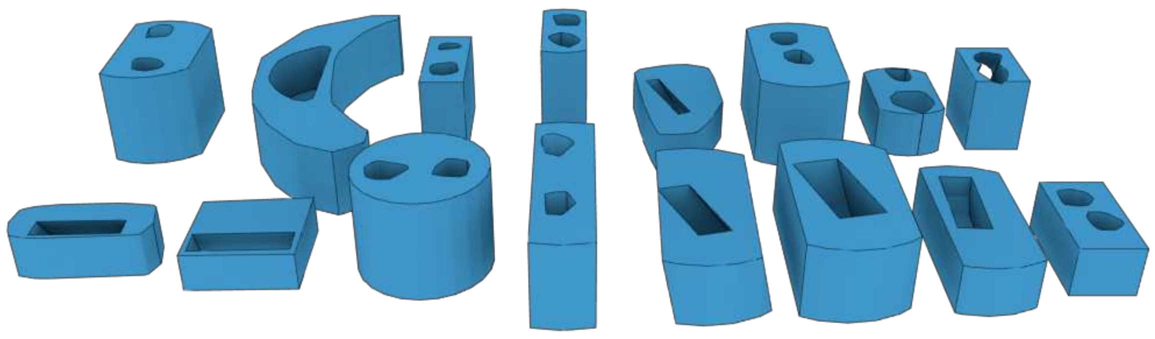


Fig. 5.12 Rendering of multiple objects designed by study participants.

we decide based on suitability and easiness rating. The gestures recommended for each task are marked in italic in table 5.1.

For most task groups suggestion count, easiness and suitability are sufficient criteria, except for the creation of primitives. When looking at box and cylinder creation separately, we would be required to choose different gestures for each of them which is undesirable as it would be likely to cause confusion with users. Adding the suggestion counts within the task group however, yields a slight preference for the *draw outline* gesture (11 suggestions vs. 9 for *hand boundaries*).

5.5 User study: System Evaluation

In this study, we evaluated the interaction cycle, design decisions and prototype implementation of MixFab. We were interested in how well non-engineering users (novices) could use the system to design meaningful objects – such as the ones depicted in figure 5.12. During the study, we collected primarily qualitative feedback to gain insight to the experience of using the system.

5.5.1 Tasks and Procedure

Participants were first asked to sign a consent form and given an introduction to the system. We started by introducing the idea of designing objects for 3D printing by showing example objects created with MixFab (Figure 5.1, c). They were then shown the "desktop organizer" walk-through (Figure 5.3) and given five minutes to familiarize themselves with the system.

To guide users during their exploration of the system, we asked them to replicate the desktop organizer example. The glue-stick and pen were provided and we guided users when necessary. Once the example was completed, we asked users to design a phone dock and provided our phone dock example (Figure 5.1, c). Participants could use their own phone or an LG Nexus4 we provided. Users were encouraged to design the object on their own, but were assisted when necessary.

Upon completion of all design tasks, we presented users with a set of statements and asked them to rate how much they agreed with each of them, on a 5-point Likert scale. We then went into a semi-structured interview asking about their experience, trying to gain insight into the usability of the system.

5.5.2 Participants

We invited 10 participants (5 female) from various departments on our campus. Participants were between 19 to 31 years old (*M*=24.3 years, *SD*=4.52), and all were right-handed. All except one participant had no experience with CAD systems or an engineering background (Rating scale, 1 a lot of experience to 5 no experience, *M*=4.56, *SD*=1.01). The latter is important because we want to evaluate how well novices can design meaningful objects using this system. To ensure that the participants would not have preconceived notions about the system and gestures we implemented, none of the participants had participated in our user-defined gesture study (see Section 5.4). Through this participant selection criterion, we aimed to further validate our gesture set.

5.5.3 Results and Discussion

All users were able to complete the tasks at hand. Figure 5.12 shows the objects designed by the study participants. These objects were designed as replication of the desktop organizer example or, in case of the phone dock, through participants own strategy.

Creation, Preparation, Assembly

MixFab's construction mechanism was quickly understood by all users. Knowing when to create a new object, modifying it and assembling two objects seemed to pose no problem for the participants. User 2 reported that *"when I was told to create the phone-dock I had a strategy in my head, thus knowing when to use a physical object."* The method of assembling objects to create new ones, was particularly well received – some users said that they *"[...] very much liked this way of putting things together, to compose objects"* (U6).

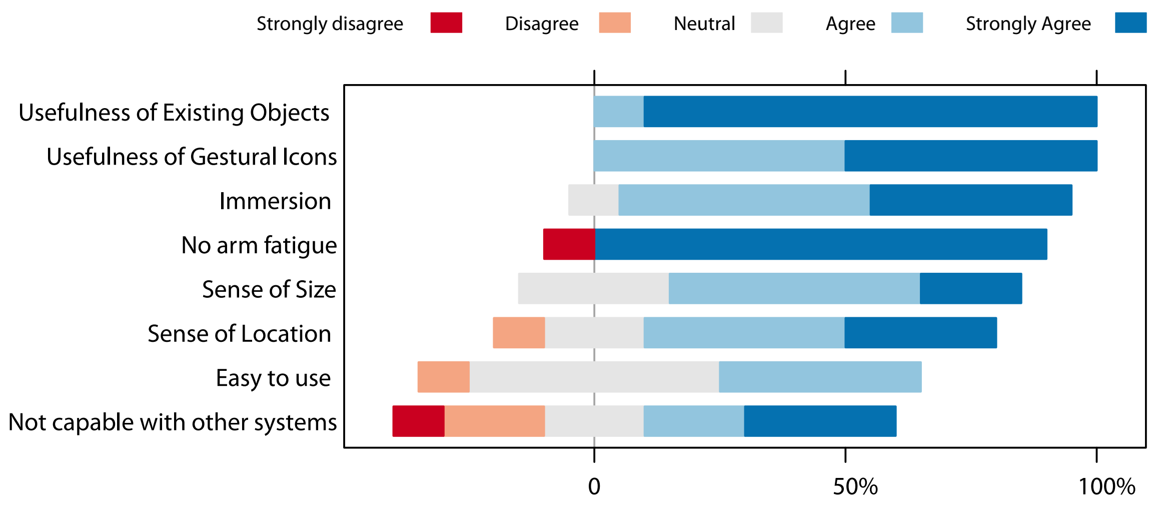


Fig. 5.13 Agreement distribution of the post-task questionnaire. The further bars extend to the right (relative to zero), the more users agree.

Using existing objects

Using existing objects during the design process was deemed useful by all users (100% agreement, Figure 5.13). Not having to measure objects and being able to place them in their desired position was highlighted by users *"I very much liked [...] the thing that you can bring real physical objects in there."* (U6). Being able to use an existing object as starting point or base for designing new ones was mentioned as one of the benefits of the system: *"I like the idea of being able to put my phone in there and design something around it."* (U8). The effortless integration of existing objects was even considered fun: *"[...] it's fun because you know there is no sort of effort required to replicate existing objects."* (U1)

Natural object manipulation

For interaction to be natural users to feel immersed and have a sense of object size and location. 90% of the participants agreed that they were immersed into the system. A majority of users (70%) agreed that they had a sense of size and location of objects as well as their hands (U10: *"I liked that the objects were as big as they are in reality."*). Users had no issues with selecting the gestural icons, further indicating that they had a sense of where things were in the frame. Manipulating objects was reported to be easy (U3: *"it felt easy to create and manipulate 3D objects compared to other systems which I image would take quite a bit of competency"*) and interacting with the system felt natural (U8: *"I liked how natural [...] the way I interacted with it [felt]"*).

Usability

Several usability aspects are subsumed under "ease of use": navigating within the system, ergonomic aspects and implementation specific artifacts. Finding their way around the system posed no greater challenge to users (U2: *"it [the system] is easy to comprehend; it's self-explanatory.")*; partly because of the gestural icons. While interacting with the system, users at times asked what to do next but shortly afterwards selected the appropriate icon and continued on their own; all users reported that the gestural icons were useful (Figure 5.13).

Mid-air gestural interaction runs the risk of inducing arm-fatigue when used for an extended period of time. During our study, 9 of 10 users reported no arm fatigue. When using MixFab, many of the gestures are performed on the floor of the system and are often interleaved with short pauses of rest. As users sit close to the system, they do not have to extend their arms very far, further reducing the risk of arm fatigue.

The accuracy of the gesture recognition had the biggest impact on usability. Some users found it hard to execute precise movements (U5: *"sometimes I wished that it was more accurate"*, U6: *"the system is very sensitive, it was hard to really make accurate movements."*) or had trouble with disengaging a gesture. Most of the issues revolved around moving objects (U8: *"when you were moving and let go it was jumping a bit"*). Others however, found the precision to be sufficient. When asked if precision was a problem, user 7 answered: *"no, that was easy"*. Overall, users agreed that the system was easy to use (Figure 5.13).

Using other systems

Our study participants had no experience with CAD and modeling tools. When asked if they would be able to design the items they designed during the study with other systems, 40% answered that they would be capable of doing so, despite no prior experience. Some users expected our system to be the way items are commonly designed: *"I have never used any of the CAD tools, but I think it's kind of like this one"* (U7).

5.6 Discussion

The results of our study provide evidence that MixFab can be used by non-expert users to design meaningful objects for fabrication – see Figure 5.12 for objects designed by users. Integrating existing objects was found particularly useful by all participants. Interaction with the system is natural, by virtue of the gestures proposed during the user-defined gesture study and the mixed reality setup. Users have a sense of size and location of the object they are designing.

Our study specifically focused on novice users, as we wanted to evaluate if MixFab can be used by this demographic. We further have demonstrated the usefulness of MixFab, the ability to produce meaningful objects, through said study and the resulting objects. However, studying how experienced designers and CAD users would interact and design with our system, would likely yield insight in the scope of the objects that can be design with MixFab could be extended. Given the expertise of those users, we would likely learn about missing operations usability issues.

5.6.1 Technical Limitations

Our current implementation has three main, technical limitations: the shape-capturing accuracy due to the depth camera performance, the gesture recognition performance due to the depth camera and appearance based approach, and a physical/virtual world misalignment due to the Holodesk-based setup.

We implemented MixFab using a Microsoft Kinect v1 consumer RGBd camera. While this camera is readily available (and at the time of writing this, also surpassed by better performing models) it suffers from high noise. This noise along the Z-axis (depth component) impacts our system implementation in two ways. For one, our shape-capturing/3D scanning performance is low (error of ± 1 mm), meaning that we can not capture detail well, and produce noisy shapes. We filter the depth image to counteract this issue. In the next chapter, we use a more advanced sensor to alleviate this problem altogether. The gesture-recognition performance also suffers from the low sensor performance. For example, users reported that clutching (letting go of) objects as difficult at times (see section 5.5.3). This is likely due to the sensor noise, as gestures do not always register correctly.

The physical setup of our system is based on Holodesk [139]. This setup enables a versatile space in which physical and digital objects co-exist. However, there is a misalignment between both types of objects. As we mirror a monosopic display into the interaction volume, digital objects which are rendered away from this mirrored display-plane cause eye convergence problems: our eyes focus on point different from where they converge. This requires users to focus on either the physical world, or the digital space. In such a situation, both spaces can not be focused on by the user at the same time. Using a stereoscopic AR setup would alleviate this problem.

5.6.2 Existing Object Integration

In MixFab existing objects first have to be digitized before they can be used which is beneficial in that it allows us to e.g. scale and alter the object. Using the physical object as tangible proxy however, would likely increase immersion. We could introduce recursion by designing an object, fabricating it and introducing its physical manifestation back into the design process, making it semi-interactive fabrication.

Capturing existing objects comes at a detail versus cost trade-off. Our prototype can capture a crude form of objects in real-time, a more detailed one can be had at a small time cost. This trade-off is likely to shift towards an increased level of detail at decreasing costs. Other material properties, such as color and texture will likely be capturable in the near future. With recent advances in appearance fabrication and 3D printing such features could also be physically reproduced.

5.6.3 Spatial Judgement

The mixed-reality environment of MixFab helps users to get a sense of size of the objects they are designing by bringing both, physical and digital world, closer together (P2). In the MixFabs prototype implementation objects look artificial however. Immersion could likely be increased by providing a more realistic object representation taking environmental lighting, proper material appearance and texture into account. Stereoscopy, in combination with the head-tracking, would further improve realism.

5.6.4 Precision

Not having to wear special equipment increases immersion thus naturalness of the interaction; not having to go through a calibration procedure prior to using the system increases the users readiness to engage with the system. Being free of user-augmentation and calibration comes at a cost, however: precision and accuracy. To some extent this is caused by the coarse spatial resolution of consumer depth cameras – something that is likely to change in the near future. A model-based hand tracking approach or specialized hand-tracking sensors [157] are bound improve precision.

Gestural modeling is less precise than traditional CAD environments. First, the RGBd sensor limits attainable precision, compared to i.e., a mouse; something that will get better as such sensors improve. Second, gestures themselves can limit precision. It is hard, for example, to accurately place an object in mid-air without haptic feedback. MixFab currently implements snapping to the ground when moving objects, or snapping to 45 degree increments when rotating. Extending this approach to tool-specific constraints (as in Interactive construction [81]) will improve gestural modeling precision, and enable users to design symmetrical, reflected and parallel features - which is not yet possible in MixFab.

5.6.5 Mixed-Reality and CAD Environments

The line between MixFab and existing CAD systems has yet to be explored. Parametric CAD environments offer precise design capabilities and are very expressive. One could imagine a *mixed-reality design for fabrication* based environment that resembles Solidworks, but is viewed through the Holodesk structure. Users would interact with such an environment through moue and gestures, and it would offer similar physical object integration capabilities as described in this chapter. Compared to MixFab, such a design environment would no longer target novices, but experienced designers. Yet, it would offer a combination of the precision and interoperability of existing CAD systems, with the benefits of MixFab: the spatial judgment support (P2) through the AR display; physical object integration (P3) through contour capturing and 3D scanning; and easier interaction (P1) through gestures.

5.7 Summary

In this chapter we have moved away from purely virtual design environments towards a **mixed-reality design for digital fabrication** approach. By virtue of an augmented-reality space, in which the physical and digital co-exist, users can interact with the virtual object-under-design directly, offering a more direct engagement (P5). Using gestures, users can manipulate this and other virtual artifacts as if they were real (P1). Due to the co-location of the design world, and the physical space, users can integrate existing physical objects easily: they can compare existing artifacts with virtual ones, or integrate existing shapes into their design (P3).

We have presented a user-defined gesture set that forms the basis of the interaction with our *MixFab* system. The latter which we have prototypically implemented, thus contribute the system as a whole. This includes technical contributions, such as MixFab's hardware design, processing pipeline and UI. We used this system to demonstrate mixed-reality design for digital fabrication as a means to lower the barrier for the casual design of fabricable 3D content. We found the effortless integration of existing objects and mixed-reality environment creates an engaging and immersive environment to create content for digital fabrication.

In the next chapter, we explore a concept similar to MixFab: where in this chapter we gave physical properties to digital artifacts, in the next chapter we will give digital properties to physical objects.

	Task / Gesture	**Count**	**Suitability**	**Easiness** / **Agreement**
Create Primitives	**create box**			0.42
	draw outline	7	3.86	4.57
	hand boundaries	3	4.00	5.00
	create cylinder			0.37
	hand boundaries	6	4.33	4.83
	draw outline	4	4.00	4.00
Rotating and Translating	**move box**			0.33
	1h grab & move	4	5.00	5.00
	2h grab & move	4	4.50	5.00
	1h push	4	5.00	5.00
	rotate box			0.40
	1h grab & rotate	7	4.57	4.86
	2h grab & rotate	2	5.00	3.50
	2h rotation	2	4.50	5.00
Scale	**scale 1 axis**			0.71
	compression	10	4.80	5.00
	scale 3 axis			0.26
	compression	5	3.80	4.80
	corner pinch & resize	3	3.67	4.00
Plane Cut	**plane cut**			0.35
	karate	5	4.00	4.20
	knife	4	2.25	4.25
	points	3	4.33	4.33
Scan	**object scan**			0.17
	dwell time	4	3.50	4.50
Add Material	**add material**			0.31
	stuff	4	4.00	4.50
	gap trace	4	3.50	4.50
	filling tool	2	2.50	3.00
Remove	**remove object**			0.22
	wipe	4	5.00	5.00
	move out	3	4.33	5.00
	smash	2	4.50	4.50

Table 5.1 List of tasks and corresponding gestures (described by more than one user) during the study. *c* is the count of how many users suggested the gesture, *s* is the reported suitability and *e* is the reported easiness. *A* the agreement among the users as defined by Wobbrock. Gestures recommended for each task group are written in italic.

6 ReForm
Bidirectional Fabrication

Situating fabrication-aware design in both the physical and virtual space simultaneously would mitigate many problems caused by the disconnection of design and target space. In the previous chapter, we demonstrated some of this mitigating power, enabled by moving design closer towards the physical world. MixFab situated the digital design environment in a mixed-reality space that enabled intuitive gestural interaction (P1), seamless integration of existing objects (P3) and offered a more direct engagement with the object-under-design (P5). However, the interaction with MixFab is not tangible (but gestural instead), thus does not offer the full quality of physical materials. Also, it is bound to a specific location which impedes in-context exploration (P4).

In this chapter we create a new design environment that maintains a physical and digital representation of the object-under-design. We do this by changing the fabrication-aware design process. Currently, the actual fabrication of the object-under-design is the last step of the process: the object would be fully designed before it is made physical reality. To overcome the rigidity of the conventional fabrication process, we introduce *bidirectional fabrication* which continuously synchronizes a digital model and physical object through rapid additive and subtractive fabrication. This gives users the ability to move flexibly between working on the digital model and the physical object (see Figure 6.1). Users are no longer limited to working on a digital model alone, but can also shape and annotate the physical object directly (P1, P5). By giving digital properties to physical objects we can

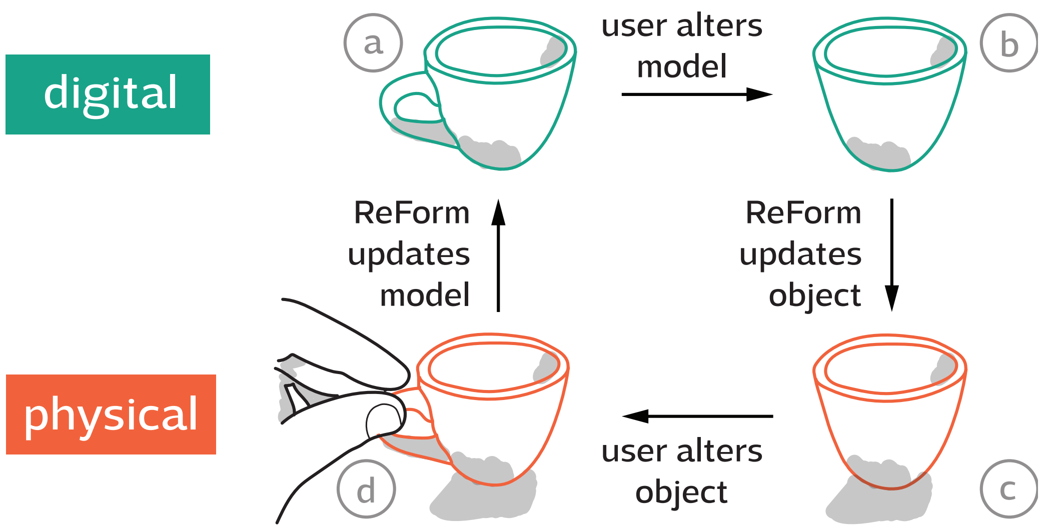


Fig. 6.1 Bidirectional fabrication closes the loop between digital modeling and physical shaping. For example: (a) user has a digital model of a cup, (b) removes the handle (c) ReForm updates the physical object (d) the user adds a new handle to the physical object (e) ReForm updates the digital model.

produce physical forms from digital models *and* use physical shape as input (P3) to produce or update corresponding digital models. Lastly, we can take the physical rendition of the object-under-design into its target context and explore it there (P4).

To demonstrate and explore bidirectional fabrication, we built *ReForm* (Figure 6.2), a system that supports design and fabrication with polymer clay. The ReForm system integrates a custom-built clay 3D printer for additive fabrication, a CNC milling machine for subtractive fabrication, a structured light 3D scanner, and a projected augmented reality display aligned to the physical object. The system can produce shape output by adding or removing clay from an object, and supports recycling of the removed material. It can take shape input from digital 3D models or by scanning physical objects.

The fabrication process starts from either a digital model or a physical object—this can be an existing object or a clay object manually produced by the user. Users provide input by editing the digital model for the next fabrication step, by directly manipulating the shape of the physical object, or by annotating the clay object with markup instructions for the system. The system supports the iterative design process with global operations on the model, such as flattening of the surface, and local operations such as extrusion based on annotation. ReForm keeps track of each model version, allowing users to navigate the design history, undo steps, and have the machine re-shape the object to an earlier version. Fabrication previews overlaying the object and interactive input are provided by a projection-based augmented

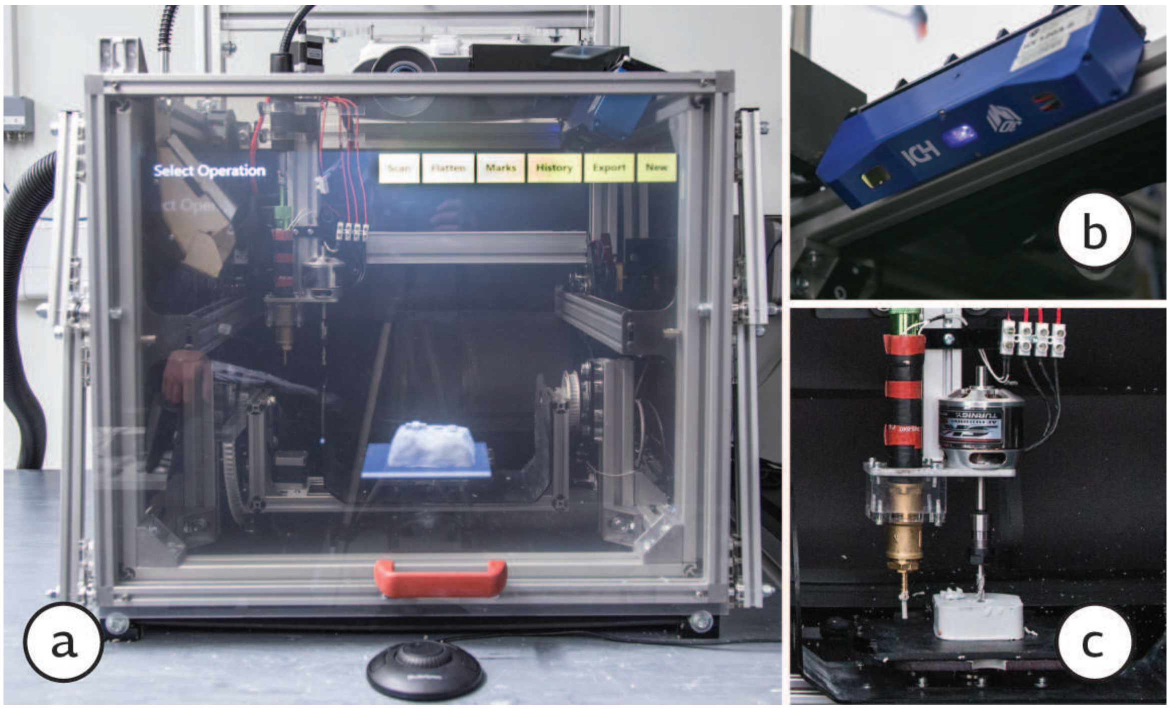


Fig. 6.2 (a) The ReForm prototype system while designing a game controller with the physical object inside the machine and its digital counterpart projected over it. (b) ReForms structured light scanner. (c) Additive and subtractive tool head.

reality interface. To summarize, this chapter contributes:

1. The concept of *bidirectional fabrication* that enables users to move flexibly between the digital model and physical object in a relaxed turn-taking fashion (which e.g., enables “undo” functionality in physical space).
2. ReForm: a bidirectional fabrication system that blends digital modeling and physical shaping practice.
3. A prototype implementation of ReForm and usage examples executed with our implementation.
4. Specific technical innovations, including the use of a two-state material (machinable and malleable) for interactive design and a novel toolpath generation algorithm for additive and subtractive fabrication.

6.1 Bidirectional Fabrication

Bidirectional fabrication fundamentally changes the digital fabrication design process and produces a range of advantages. First, it allows users to choose the best-suited tools for each portion of the process: creative, expressive, and ad-hoc 3D design is easy to perform through direct physical manipulation of an object, while tasks involving precise input or repetition are better done using digital tools. Second, it enables 'turn-taking' between the user and machine. This allows each to leverage their respective strengths and permits incremental fabrication of objects with gradual addition of parts or detail. Users can then perform 'on-the-fly' validation and refinement of the style, size, and confirm each element is fit-for-purpose. Third, the bidirectional mapping between the digital model and physical object facilitates the extension of version tracking from the digital to the physical. Combined with support for both additive and subtractive processes, bidirectional fabrication can extend undo, redo, and add 'previous version' functionality to physical objects.

Bidirectional fabrication fundamentally builds on four key components: shape input, shape output, visual input and visual output. Physical shapes can serve as input. Such shapes can be pre-existing, or may have been previously produced, in a malleable material, during the design process. Shape output is the ability to produce and update physical shapes, which enables the digital-to-physical synchronization. Through visual input, users can annotate the physical object to interact with its digital counterpart e.g., mark the position for a hole on the physical object, but create the hole on the digital model. Visual output provides a preview of such operations. With it, users can configure changes and modify the digital model.

- **Shape Input**: Physical shapes can serve as input. Objects previously produced by through bidirectional fabrication can be input after modification to update their digital counterpart. Other pre-existing objects can serve as starting point for the design process.

- **Shape Output**: Bidirectional fabrication produces physical objects which can be inspected, modified, combined, taken into context, compared, used, destroyed and its material reused. These objects need not be completely re-fabricated when the digital model changes, but can gradually be updated by combining additive and subtractive fabrication.

- **Visual Input via Annotation**: Users can annotate the physical objects using colored pens. These annotations are interpreted by the bidirectional fabrication system and serve as a selection and command mechanism. As these annotations are made directly on the physical model, they are highly contextual and intuitive to apply.

- **Visual Output via Projected Overlay**: Objects within a bidirectional fabrication system can be augmented through projected augmented reality. We overlay information and can enhance the physical object e.g. give new colors, textures or additional information such as the model dimensions or volume. Users can configure operations before they execute, in addition to enabling intuitive visualizations and previews.

An addition, a bidirectional fabrication system could also supports digital input (via existing modeling environments or online databases) and digital output (via model export). Based on these key capabilities, we provide a variety of operations that can be used to design fabricable artifacts.

6.2 ReForm

ReForm utilizes bidirectional digital fabrication to enable a relaxed turn-taking style of iterative design. By synchronizing the physical object and digital model, the tangible artifact can be altered by users and the system alike. This allows us to maintain important digital operations, such as undoing changes, regardless of their source (user or machine). We also support operations that would be difficult or tedious to perform manually e.g. smoothing clay-modeling artifacts, patterning parts of the model or creating accurate holes. ReForm builds on the components of bidirectional fabrication as described in the following.

6.2.1 Digital Model Management

ReForm maintains and synchronizes a digital model and its physical counterpart. The digital model is a triangle mesh storing geometry, normals and luminosity. On update, ReForm creates a new version of the model, and maintains a copy of the previous version. This version history enables features such as undoing changes and allows new users to understand the steps taken to design an existing object.

The latest model is available to external mesh-modeling systems such as Blender, Maya or Rhino. These systems may alter the digital model; ReForm will then update the physical object. A tight integration into these external software packages (similar to the SPATA tools, chapter 4) would offer a rich set of digital modeling operations, especially for expert users.

6.2.2 Shape Input

ReForm fabricates objects using a clay-like human-deformable material, so that users can manually alter the physical object e.g. with their hands or by using tools (see 'Physical Shaping'). Once altered, ReForm scans the modified object and synchronizes the digital model. This shape-input mechanism enables users to directly modify the object (hence the model) in physical space. The object can be taken into context, manipulated there and placed back into ReForm for synchronization. Users can add fine artistic details and features that are beyond the shape-output capabilities. Current 3D printers typically produce rigid plastic objects which can not be altered in such a way, thus are less well suited for this iterative design style as alterations have to be performed in the virtual design environment.

6.2.3 Shape Output

ReForm supports physical shape-output, both by fabricating an object from scratch and through incremental updates. This output is performed subtractively and additively, so that the physical object need not be recreated in every update step, saving time. Updating, rather than recreating the object offers a range of benefits. First, we do not have to discard the entire physical object for every update. This wastes less material than complete refabrication. Further, users can reuse previously removed material for additive updates, further reducing material waste.

Second, ReForm can choose the fabrication method most suited to a given update or fabrication task. Concave shapes are difficult to produce subtractively, but become feasible additively. Depending on the shape—and in the case of subtractive fabrication, the input material—one method will usually have a lower fabrication time and each will offer different surface qualities and finishes. We execute shape-output with the most well-suited fabrication method, or when appropriate, combination of methods.

6.2.4 Annotation

Users can directly annotate the physical object using colored marker pens. ReForm detects such annotations during object/model synchronization. Annotations serve two purposes: selection and commands. To select an area for later processing (see 'Selective Operations') users draw a closed loop around the area and fill it with a hatch pattern (Figure 6.4, a).

Visual languages can be used to command machine operations (e.g. Song et al. [82]). In ReForm, a simple annotation language uses drawn shapes to preselect specific operations (which are then configured and executed using the visual output capabilities):

- a cross preselects the hole drilling operation, with the diameter of the hole preset to the diameter of the cross' inscribing circle;
- rectangular shapes preselect the local surface smoothing operation (see Section 6.3.2);
- irregular shapes preselect the extrusion operation (see Section 6.3.2).

6.2.5 Visual Output

To complement shape-output and facilitate interaction, ReForm provides a rich visual output channel comprised of two components: an augmented reality interface and graphical user interface. The back-projected AR interface is aligned with the physical object. We maintain this alignment by correcting for motion parallax, which also provides important depth cues. Through this AR interface we can preview design decisions and new model states before updating the physical model. We use the AR preview to guide users when configuring digital operations such as drilling precise holes or flattening the top surface.

The graphical user interface is overlaid on top of the AR display. Users interact via a jog wheel input device as the UI only necessitates flat menus, sliders, buttons, and sequential selection mechanisms. This form of interaction and UI is in line with existing fabrication machines. As users do not have reach out to the ReForm system (but only hold the jog wheel) we neither introduce fatigue nor occlude the AR interface.

6.3 User Interaction

ReForm combines two previously separate design practices: digital 3D modeling and physical shaping. Users first create a model/object pair, either starting from a digital model or a physical object. Throughout the design process, users manipulate the artifact being designed in a relaxed turn-taking fashion, either through digital modeling operations or by physically shaping it. All operations, no matter if brought about digitally or physically can also be undone.

6.3.1 Model and Object Creation

To begin users must create a digital/physical object pair. Users have the choice to (a) start from an existing digital 3D model or primitive, (b) start from an existing physical object, or (c) to restore from a previous design session/clay representation. If the user does not have a clay representation, ReForm fabricates one. The system recommends a subtractive or additive approach depending on the estimated fabrication time (see Section 6.4.3 for implementation details).

Digital models can come from a variety of sources. Online-databases such as Thingiverse[1] and GrabCAD[2] offer users access to a large variety of existing starting points. Physical objects from various sources can also serve as a starting point. Existing objects, for example items bought in a store, can be used as input and transformed into a model; ReForm can replicate otherwise unmodifiable objects in clay. Users can also start with hand-made clay objects or objects from previous design sessions.

6.3.2 Digital Modeling

ReForm enbles users to perform operations in the space that suits them best. Thus, precise or regular operations can be performed through digital modeling. We support two classes of modeling operations: the ones the modify the object-under-design globally and the ones that modify selectively. The latter become selective through annotation-input: users draw on the physical object to mark the area of influence. Users can manipulate and preview the effect of any operation through the augmented-reality interface, enabling them to make informed decisions.

Besides the operations offered by ReForm itself, existing mesh-based design environments can be used. Users can open the digital model in e.g. Blender. Annotations drawn on the physical object, become selections inside the mesh-modeling tools (in case of Blender, they become vertex groups). Once all modifications made inside the virtual design environment are complete, ReForm updates the physical rendition. This way, operations previously confined to the digital space, get transported into the physical world immediately. Note however, that to use such existing mesh-based design tools, expert knowledge is required (P1) and they do not use the AR interface of ReForm, thus may diminish spatial judgment (P2). This is why ReForm offers its own operations, which are detailed below.

Global Operations

Global operations (e.g. flatten, scale, and virtual assembly) affect the entire model. They are previewed using the visual-output feature before they are applied. This preview enables rapid exploration of the design choice at hand (e.g. setting the cut-height when flattening an

[1] http://thingiverse.com
[2] http://grabcad.com

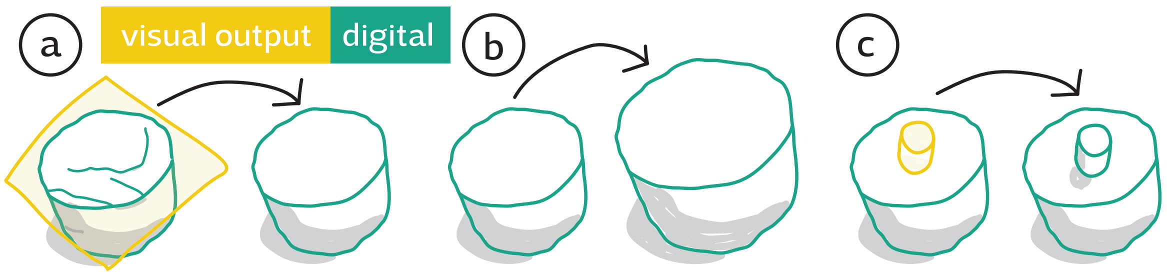


Fig. 6.3 Global Operations. (a) global flatten to produce a level top (b) scaling the whole model (c) virtually assembling two models

object, see Figure 6.9, a) and reduces reduces the risk of error.

Flatten (Figure 6.3, a) removes a side of a model to reveal a flat surface. Due to the material properties of the physical object, manual modifications tend to result in undesired artifacts such as waves, ridges and valleys. Using global flatten, users choose a side and cutting height at which a flat and smooth surface is created, thereby doing away with the undesired artifacts.

Global scaling resizes the model (Figure 6.3, b) by a variety of measures. Users can scale the model based on a single dimension and scale the others in an aspect-ratio preserving manner, or set the desired value for each dimension individually. A target volume could also be specified, then the object is uniformly scaled to the desired capacity.

Virtual assembly (Figure 6.3, c) enables users to add an existing digital model to the object-under-design. Users first select the digital asset they want to use in the same way they would create a new model/object (see Section 6.3.1). Then, users place the new model with respect to the current one, using the AR-based visual preview. By choosing whether the object is to be added or removed, users confirm their placement and the two models (the new one and the current object-under-design) are assembled.

Selective Operations

Targeted operations require the selection of an area of influence. We use annotation-input to enable users to mark the area they want to manipulate on the physical object – hence, users draw on the physical object directly and then select what they want to do. Through annotations users can issue commands, selecting the operation to perform. For example, shading in an area with a hatch pattern (Figure 6.4, a) offers it for flattening or extrusion, drawing a cross results in a hole being drilled (Figure 6.4, b) and two circles produce a patterning (Figure 6.4, c).

Local flatten serves the same purpose as its global counterpart: remove undesirable

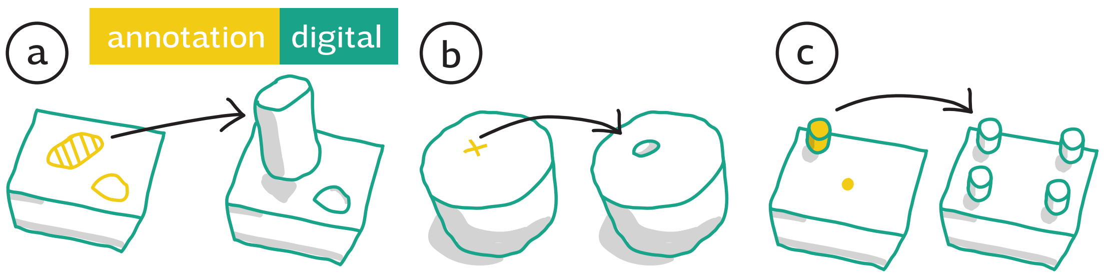


Fig. 6.4 Selective operations using annotations. (a) Extruding an annotated patch or contour (b) Producing accurate holes (c) Patterning model features

physical manipulation artifacts. Besides annotating the desired flattening area, no user-interaction is required. The cutting height is automatically determined based on the mean height of the annotated area. Within the annotated area, a smooth surface (with a normal parallel to the average normal of the flattening patch) is created at the determined height.

Extrusion (Figure 6.4, a) adds depth to the annotated outline resulting in material being removed or added. The latter depends on the extrusion height parallel to the average normal of the annotated patch: moving upwards from the surface (positive extrusion height) results in material being added, moving downwards from the surface (negative extrusion height) results in material being removed. This feature enables users to produce cavities and protrusions which would be difficult or tedious to create manually. Similar to extrusions are holes, which are difficult to produce manually due to varying diameter requirements. A cross annotation marks the center of the hole (Figure 6.4, b) and users configure the hole diameter and depth through the graphical user-interface.

Replicating patterns is tedious to do manually, but effortless in the digital domain. To create a pattern (Figure 6.4, c), users select the area they want to replicate by drawing an outline around them. The pattern origin is annotated using a filled circle. Users can then choose the desired pattern type (circular, rectangular, and linear patterns), number of repetitions and distribution, on the augmented-reality interface. After making those choices, the selected area is replicated accordingly.

6.3.3 Physical Shaping

Due to ReForm's malleable material, users can modify the physical objects directly and in context. As the material sticks to itself, users can manually add more material. The shape can also be bent, smeared and otherwise plied using bare hands (Figure 6.5, a), much like one would with any other clay object. The rich set of existing physical clay sculpting tools can also be used to manipulate the physical object (Figure 6.5, b). Simple tools like knifes

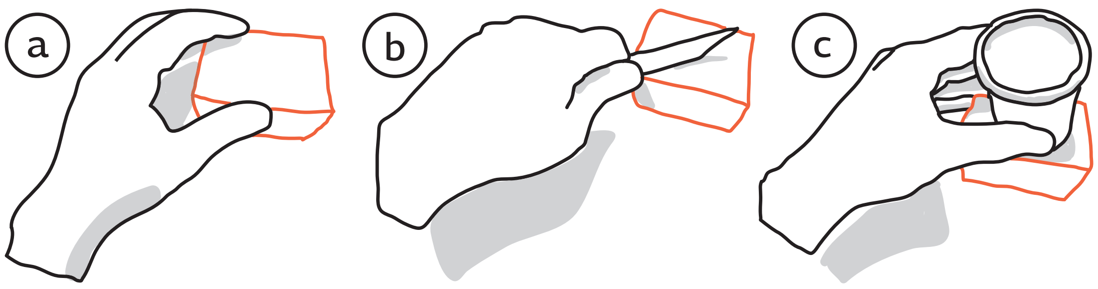


Fig. 6.5 Manual modifications of the material. (a) Users can mold objects with their hands (b) using tools, (c) use existing objects.

and cutters to more specialized sculpting devices provide a broad spectrum of devices, and expand the input possibilities for the digital fabrication design process. All modifications of the physical object are reflected back to the digital model through ReForm's continuous synchronization cycle.

Physical objects are not bound to any location, thus can be taken into context and manipulated using any object found in the environment (Figure 6.5, c). One could create an outline to serve as guide for other operations by impressing an existing artifact into the clay; for example to create a hole for a pen, users could press the pen into the material. This form of material interaction, combined with the operations detailed below, makes for a more intuitive design process.

6.3.4 History and Versioning

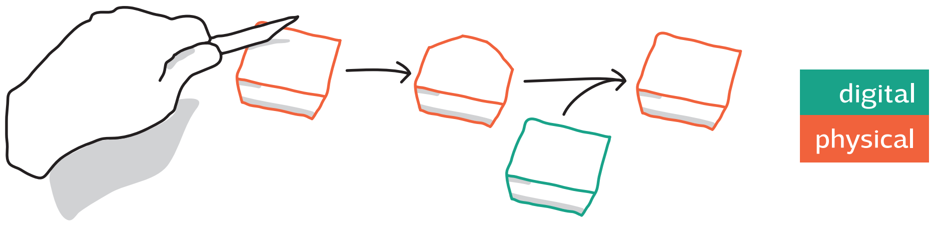


Fig. 6.6 Undoing changes to the model or object. Here, the user modifies the physical object which we can undo using the previously stored digital model.

Changes made during the design process can be undone, no matter if the modifications were made on the physical or digital artifact. We maintain a history of 3D models, each of which we can restore as a physical object. This way, we enable free exploration with the physical artifact, as there are no irreversible “mistakes”. Actions that did not result in the desired outcome can be undone. For example, if a user cuts away parts of the object to explore the aesthetics of these changes (Figure 6.6), but does not consider the outcome desirable, we can restore the previous state by undoing the manual interaction. While choosing which version to revert to, ReForm provides a preview using its augmented reality interface.

6.4 Implementation

We built a prototype implementation of the ReForm system in order to evaluate the bidirectional fabrication concept. ReForm integrates several components in a novel way: a material which is machinable, yet malleable; a five-axis CNC machine with a custom clay extruder and milling spindle; a physically aligned and motion-parallax compensated, augmented reality interface; a structured light 3D scanner; annotation detection and custom toolpath generation to use our machines capabilities.

6.4.1 Material

Common polymer clays and puttys are too soft to be machined. Their malleability makes them easy to work with manually and easy to extrude in an additive fabrication setup. However, their softness also renders these materials unsuitable for subtractive methods as soft material clogs the milling bits. To use additive, subtractive, and manual fabrication methods with one material, we use TecClay[3] as it is machinable at room temperature but becomes malleable when heated to approximately 50 °C. ReForm can produce both a cool and hot airflow (see Section 6.4.2) in the machine in order to regulate the model temperature for removing, adding, and forming the material. The extrusion cartridge (see Figure 6.7, d) and nozzle are kept heated to 55 °C to reduce the required extrusion force. At this temperature the clay becomes slightly adhesive and bonds well with itself and the perspex build-platform.

6.4.2 Hardware

We separated the hardware into two main components: the control system (ReForm Core) and the main frame (ReForm). ReForm Core contains supporting components for the main machine. It houses 12V and 24V power supplies, six *CW5405*[4] stepper controllers that are connected to a LinuxCNC powered MiniITX computer through a *HW08* IO board. An emergency switch at the front of the ReForm Core cuts power to the motors if necessary.

[3]http://www.kolb-technology.com/en/products/classic/clay.html
[4]http://cnc4you.co.uk/

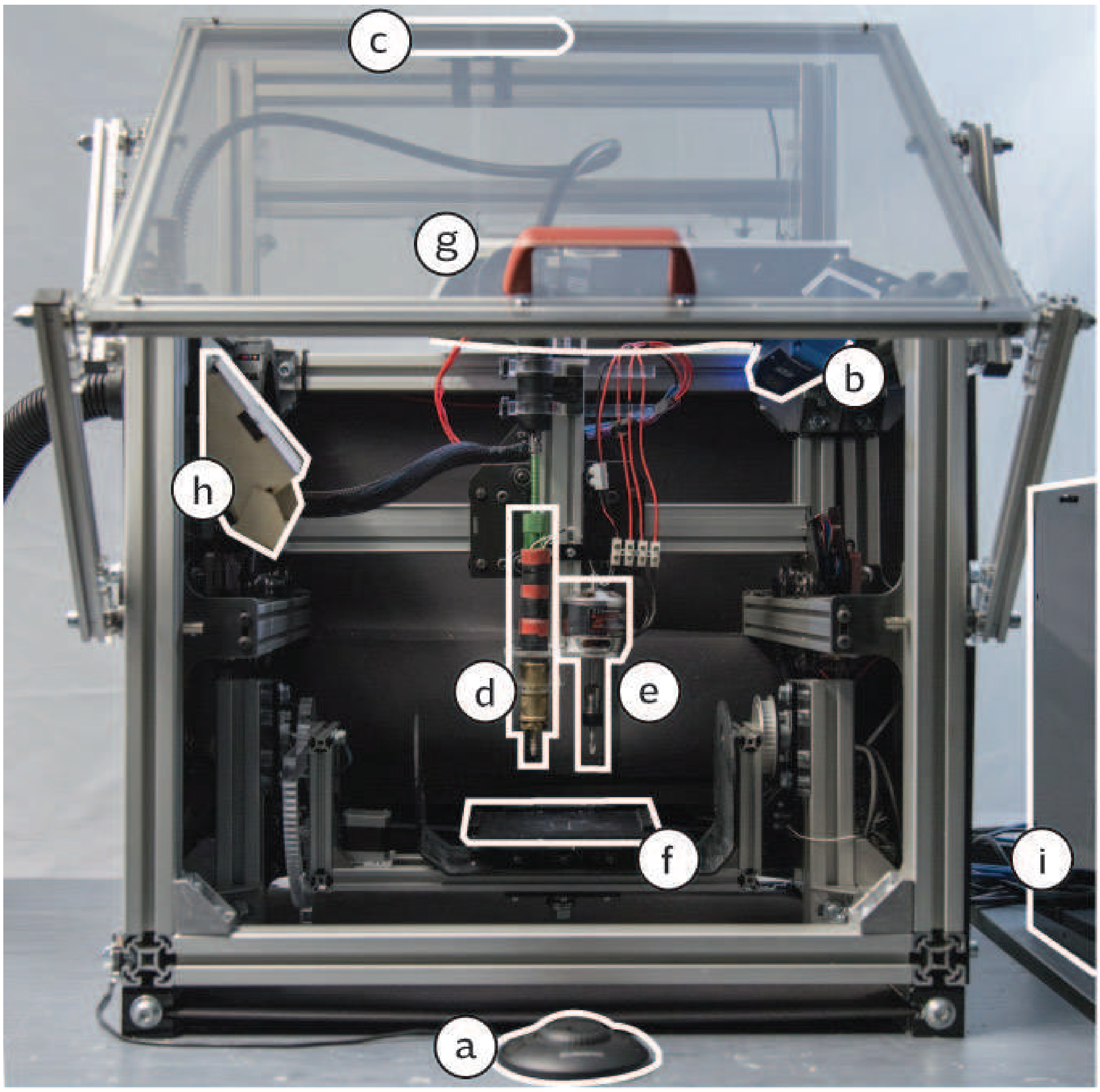


Fig. 6.7 The ReForm prototype: (a) a jog wheel for user-input (b) LMI HDI120 3D scanner (c) Asus Xtion depth camera (d) heated clay extruder (e) milling spindle (f) build plate (g) projector and screen (h) air-guide (i) ReFormCore.

ReForm is constructed within an aluminium frame. A spindle drill and clay extruder pair (Figure 6.7, d, e) are mounted on an XYZ motion platform. The clay object is attached to a build plate held onto the two rotary axes A/B using ball detents. A structured light scanner (Figure 6.7, b) is mounted on the right side of the frame for an unobstructed view of the object. A custom air-guidance system directs an airstream to the workpiece (Figure 6.7, h). The airstream is generated using a *Kärcher MV3 P* vacuum cleaner and passed through a heating element. We use an Arduino-controlled relay to automatically turn the airstream on and off. Situated at the top of the frame is a *Xtion* depth camera (Figure 6.7, c) and a short-throw projector for the augmented-reality interface (Figure 6.7, g). This interface is projected on the articulated front-door which holds a semi-transparent projection screen. In front of the machine (outside of the door) users find the jog wheel (Figure 6.7, a) for interacting with ReForm.

The spindle is based on a 260 rpm/V brushless DC motor whose 8 mm shaft we replaced with an ER11 collet (Figure 6.7, e). A 6 mm flat-tip two-flute cutter is fitted in the collet. Compared with steeper tip angles this flat-tip configuration produces non-clogging clay flakes. The motor speed is controlled from an *Arduino* through an electronic speed controller (ESC).

We extrude warm TecClay through pressure by actuating a threaded rod plunger in a metal cylinder. Due to the surface friction of the clay (which is reduced by heating the cartridge), a 3.1 Nm motor is required. To reduce the moving mass of the XYZ platform we mount the 1.4 kg heavy motor off-axis and transport its rotational movement with a flexible drive shaft to the extruder. This assembly extrudes the clay through a 3 mm heated brass nozzle mounted 2 mm above the cutter (Figure 6.7, d).

To 3D scan the object we use an LMI HDI120 structured light scanner (Figure 6.7, b) with an accuracy of 60 - 118 μm. While scanning we take six snapshots; rotating the model by 60° each time around the build-plate center. By using white TecClay we minimize exposure time for each snapshot, so that a 360° scan takes about 1.5 minutes. We use LMIs *FlexScan* software to align the snapshots and merge them into one 3D mesh model. This scanning process also recovers a monochrome texture which we use for annotation-input. A scanned model has approximately 100k vertices.

A BenQ W710ST short-throw projector (Figure 6.7, g) projects onto the transparent projection screen held in the door. We manually calibrated the virtual camera to match the physical scene, and using an *Xtion* depth camera (Figure 6.7, c), we track the users body to provide a motion-depth cue. This allows us to render aligned virtual 3D previews over the physical clay model. We use the WPF-based Helix toolkit to render 3D, and custom WPF

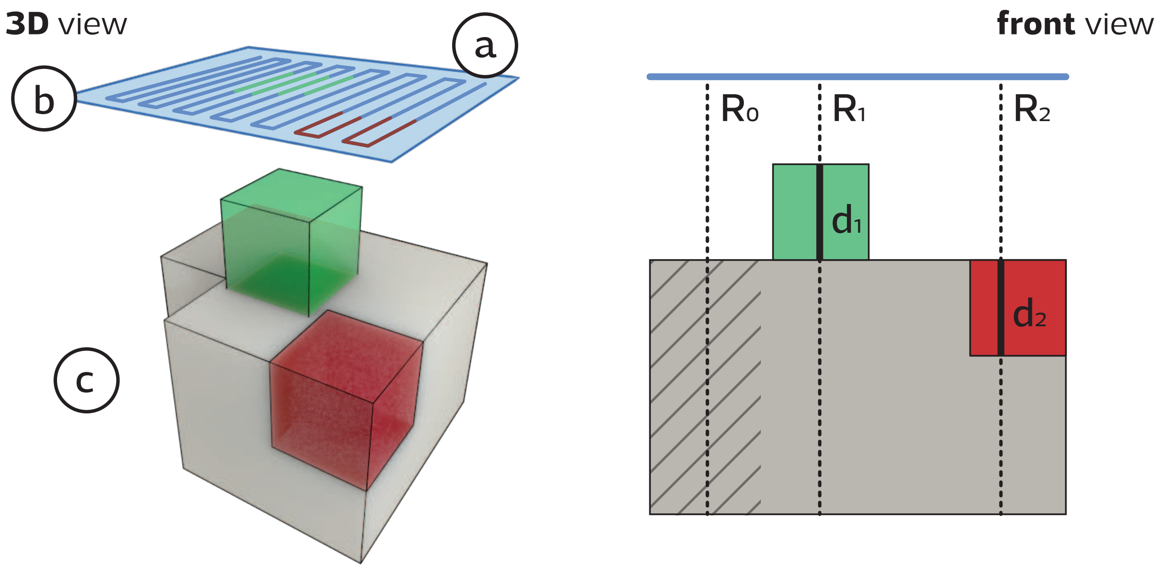


Fig. 6.8 Toolpath generation. (a, b) a space-filling curve is computed on a generator surface within the extremes of the scene. (R_0, R_1, R_2) Along the curve, rays are cast normal to the generator surface. (d_1, d_2) The distance between the existing and target surface determines whether material has to be added or removed.

controls for the 2D menu.

Users interact with the system using a *Contour Design ShuttleXpress* jog wheel (Figure 6.7, a), which is well suited for the discrete menu scheme and other AR operations. This way the user's hands do not occlude the display, contaminate it with fingerprints, or suffer from fatigue.

System Performance

The toolhead can travel at a maximum speed of 45 mm/sec along the XY axes, 2 mm/sec along Z, 600 deg/sec around the rotatry A axis and 30 deg/sec around the table-tilting B axis. Our clay extrusion system can extrude material at a maximum rate of 1 cm^3/sec with its cartridge holding 104 cm^3 of material. When milling, the maximum spindle plunge depth is 4 mm. The 3D scanner to machine calibration error is less than 0.15 mm.

6.4.3 Toolpath Generation

Bidirectional fabrication requires us to compute machine instructions that transform between two arbitrary digital meshes. The resulting *toolpaths* describe the motions a machine has to execute in order to add or remove material. Toolpath generation for subtractive and additive methods, in isolation, are well studied problems [39]. A prominent method is based on isoparametric planar surface curves [158] where a zig-zag curve is sampled and projected onto the model surface to determine the cutting depth. Additive layered manufacturing toolpaths are generated by slicing the model parallel to the XY plane and following the generated profile [159]. In this work, we combine subtractive and additive toolpath generation in one algorithm, based on isoparametric planar curves.

We however need to generate additive and subtractive toolpaths, determine where to add and where to remove material, and compute the paths themselves. We implemented a novel toolpath generation algorithm, which combines both tasks. The algorithm takes as input the currently existing surface and the target surface we want to produce. It consists of four steps: isoparametric curve sampling [158] (see Figure 6.8), patch extraction, optimization, and path development.

Note that this algorithm does not take the machine geometry into account, and as such might produce toolpaths that result in tool/workpiece collisions (spacing and dimensions of ReForm's tools mitigate this problem). This limitation of the algorithm could be addressed by optimizing the generated path so that the workpiece is oriented in a collision free state using the two rotary axis. Near the build-plate and for extreme model convexities no such collision free state exists, thus existing model geometry would have to be removed and rebuild.

Curve Sampling

We start by constructing an isoparametric zig-zag curve on a generator surface (a plane for XYZ milling, a cylinder for rotary milling; see Figure 6.8), so that the curve fills the extents of the models. We determine the feed-forward step (sampling distance: d) heuristically from the machining tolerance (also called scallop height: t) and cutter radius r as $d = \sqrt{r^2 - (r-t)^2}$. Even though more advanced estimation methods are available [39], this simple heuristic works well in practice. At each sampled point on the curve we cast a ray normal to the generator surface to determine the machine action required at this point (see Figure 6.8, right). Three cases are possible:

1. **No intersection** (R_0): the cast ray intersects neither the existing nor the target surface. No action is required.

2. **Target Surface before Existing Surface** (R_1): the ray intersects the target surface before the existing one, hence material needs to be added.

3. **Existing Surface before Target Surface** (R_2): the ray intersects the existing surface before the target one, hence material needs to be removed.

Patch Extraction, Optimization and Development

The previous step produces a path consisting of subtractive, additive and passive samples (Figure 6.8, a). We group consecutive samples of the same kind forming *machining patches*. Passive patches (see case 1 above) become travel moves along the path. As ReForm executes additive and subtractive passes separately, at this point we decide whether we want the subtractive or additive path and replace the other patches with travel moves also. We now have the surface machining path with many unnecessary travel moves. We optimize travel patches by finding the shortest path between the start and end point of the patch along the generator surface. The travel height is determined by sampling the existing model along the new travel path via ray casting. The optimized surface machining path does not account for material being successively taken away or added. In this step we interpolate the path to remove material at a given layer height (and not plunge all the way into the model), or add material at that height respectively. In this stage we also incorporate fabrication specific aspects, such as a slower first layer when adding material, to ensure it bonds well with the printing surface.

Execution Time Estimation

We use the toolpaths generated by our algorithm, not only to update the physical object, but also to estimate the fabrication time to initially create a physical rendition. Through this estimate, users can made an informed decision whether to use additive or subtractive fabrication for the initial object creation.

To estimate a toolpaths execution time, we integrate (sum up) the toolpath segment execution time. We compute the latter as product of the segment length and its feedforward rate (moving speed). This simple method ignores acceleration, thus produces too optimistic estimates. However, we do not display this fabrication estimate directly, but use it to compare the fabriction times of additive with subtractive manufacture. As both toolpaths are produced by the same algorithm, the segment length similar, thus the acceleration-induced error is roughly the same.

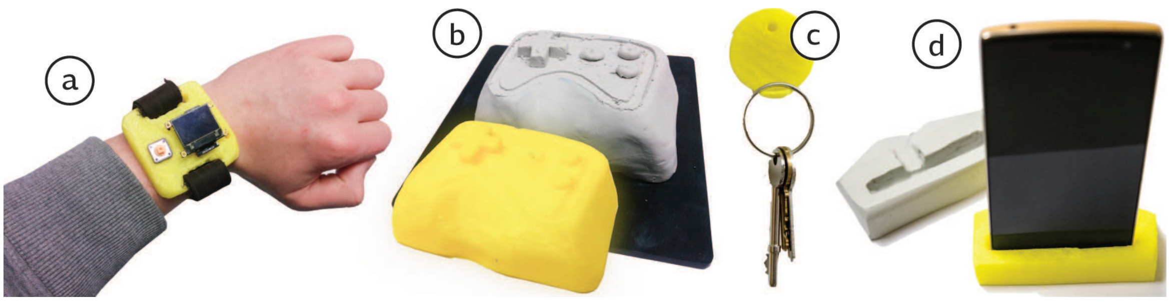


Fig. 6.9 Objects designed with ReForm. (a) User wearing the exported smartwatch prototype. (b) 3D printed and clay versions of the game-controller. (c) Key hook. (d) Phone dock and corresponding clay object.

Implementation Details

Our algorithm relies heavily on ray-casting meshes. We accelerate this process using a KD-tree to reduce the required triangle-ray intersection tests. Sampling a 100×100 mm planar zig-zag curve with a $t = 0.5$ mm machining tolerance and $r = 1.5$mm cutter radius (resulting in a sampling distance of $d = 1.118$ mm) requires 7921 ray-casts.

Using spherecasts (mesh-sphere intersection along a ray), rather than raycasts would yield toolpaths closer resembling the model surface. However, spherecasts come at a computational and simplicity-of-implementation expense.

6.4.4 Annotation recognition

We use the *DBSCAN* clustering algorithm [160] to detect user-drawn annotations on the model. First, we compute a set of candidate vertices by applying a Luma threshold filter on the monochrome vertex colors recovered by the 3D scan. Then we cluster vertices based on their color and spatial proximity using DBSCAN. To ensure the cluster is on the surface, we check if all vertices in the cluster are topologically connected in the mesh.

To detect shapes (e.g. unfilled rectangles or crosses), we fit a plane into the cluster vertices, project the vertices onto that plane and compute their convex hull. We then find all connected vertices whose projection falls in the convex hull using an arbitrary vertex in the cluster as seed point. As a result we get all vertices within the cluster—no matter if they were colored or not—and a 2D projected image that we can use to detect commands.

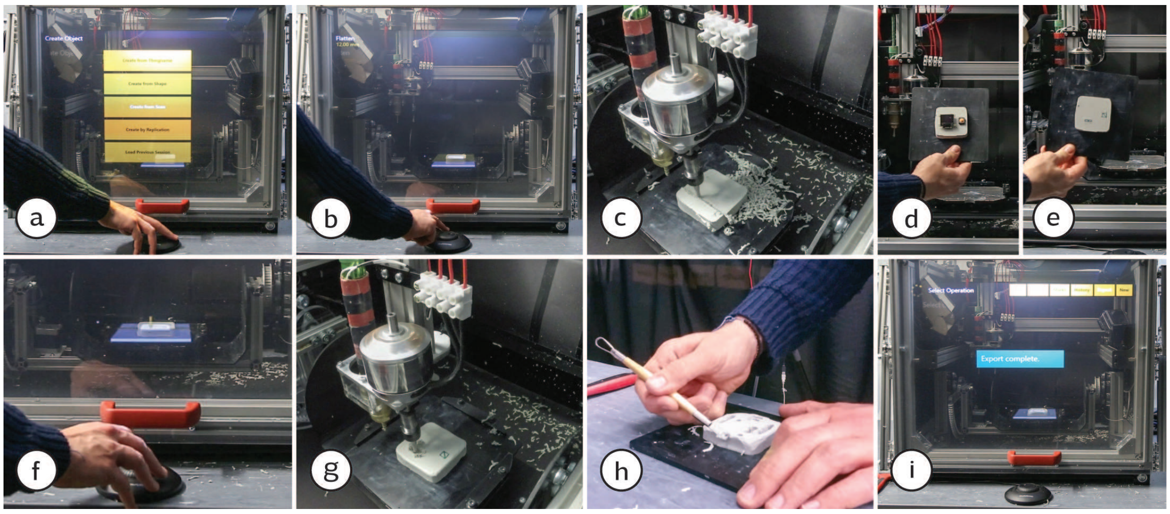


Fig. 6.10 ReForm smartwatch design walkthrough: (a) flattening the top surface (b) ReForm updating the physical object (c) existing components placed on the prototype (d) component positions are annotated (e) using selective extrusion to create the display cavity (f) the physical object is updated (g) the user shapes the watch to their liking (h) the final model is exported for 3D printing.

6.5 Application Examples

We describe two application examples to demonstrate ReForm's features and benefits. Both examples highlight how ReForm blends digital modeling with physical shaping and demonstrate turn-taking in the bidirectional fabrication process. The first illustrates a single-user design that combines shape input from physical 'on-body' sculpting with annotations to support the precise insertion of electronic components. The second describes a group-based design of a game-controller that starts with a physically sculpted base-shape and evolves, via multiple turns and iterations, to a 3D-printed artifact. Both demonstrate physical history.

6.5.1 Walkthrough: Smartwatch

In this example we use ReForm to construct a smartwatch prototype that we mold to fit a users wrist, yet precisely hold electronic components. This demonstrates how organic physical shaping and precise digital manipulation are combined in bidirectional fabrication.

We start by creating the body of the watch by manually cutting a block of clay. After warming the clay it becomes malleable and we can shape the watch body. During this process we can try the prototype on our wrist to see if it fits as a watch and will be comfortable to wear. When the rough shape is complete, we place the object on a build-plate, insert it into ReForm and create a new model by scanning the clay object (Figure 6.10, a).

After ReForm scanned the model, we use the *global flatten* operation (Figure 6.10, b) to smooth the top surface of the watch. Once we confirm the desired cutting height, ReForm updates the digital model, and its physical counterpart (Figure 6.10, c).

Next we make space for the display and electronics. To this end we use the display and physically press it into the soft clay (Figure 6.10, d). With a marker, we then mark the created impression (Figure 6.10, e) to select the area which we want to carve out. The object is re-inserted into ReForm and scanned. ReForm detects the annotations and offers the *selective extrusion* feature (Figure 6.10, f) which we use to create the display cavity. Once confirmed, ReForm updates the physical object and carves out the material as designed (Figure 6.10, g).

We take out the updated object, place the display inside the cavity and try the prototype on our wrist. With the components placed as desired, we finalize the watch design by shaping the watch body (Figure 6.10, h). Using sculpting tools we directly manipulate the physical clay object.

Once all components are placed and the shape of the smartwatch is as desired, we can take the digital model (Figure 6.10, i) and 3D print it in a more suitable material i.e. PLA. This produces the final prototype which is subsequently assembled and used (Figure 6.9, a).

6.5.2 Walkthrough: Game Controller

In this example we develop a new game controller and experiment with different designs and button configurations. Here, we demonstrate how ReForm can be used in a collaborative setting where multiple users can make changes to the design by modifying the physical object in-turn.

Using existing controllers as a guide, we begin with selecting a block of clay from which we form the basic shape of the controller. We use our hands to approximate the curvature and geometry of the design (Figure 6.11, a); creating a depression in the center and sides which follows the crease of the hands. Using tools we carve away chunks of clay. Where we remove clay accidentally, or make changes feel wrong, we push the clay back into place. This process is repeated until we are happy with the base shape.

Next, the controller is placed into the machine. We select ‘Create From Scan’ to start a new ReForm session using this object. Once the scan is complete, we select the ‘Flatten’ command to create a flush working surface. Using the augmented-reality preview we locate a cut height that will leave no troughs on the surface before confirming the operation. No

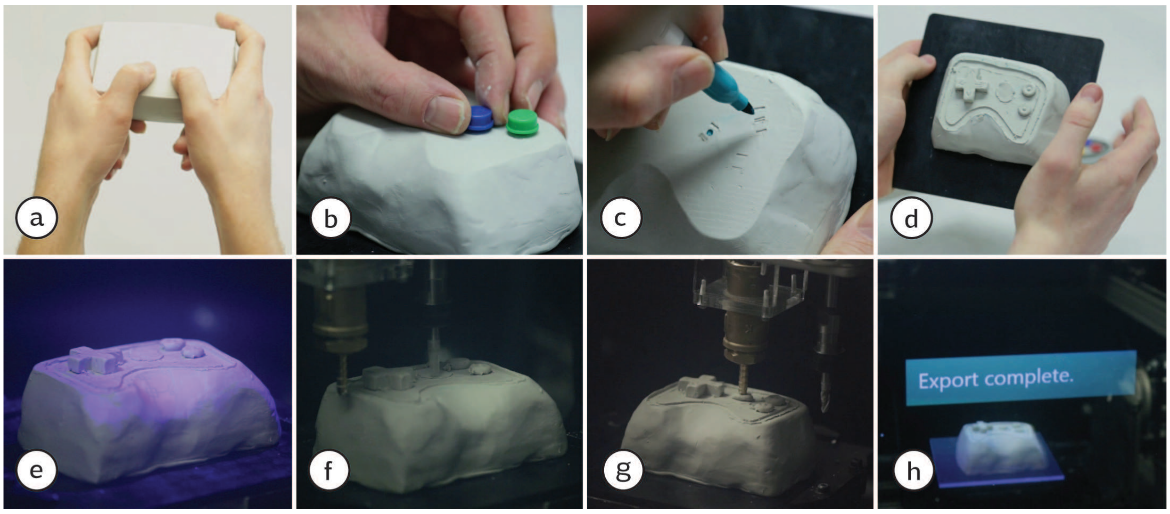


Fig. 6.11 ReForm game-controller design walkthrough: (a) shaping a game controller (b) placing buttons on the prototype (c) annotating the button positions (d) user damaged the prototype (e) damaged object is scanned (f, g) ReForm repairs the object (h) the final model is exported for 3D printing.

measuring is required. The digital model (thus the physical object) is then flattened and updated accordingly.

Once complete, we take the clay out of the machine and place it amongst a set of available interface components. Everyone in the group holds the prototype, passes it around, and alters the button placement (Figure 6.11, b); discussing the merits of alternatives—a person with smaller hands uses a fingernail to score a line in the clay that illustrates their constraints. A three-button configuration is agreed and the final button positions are marked out with a pen (Figure 6.11, c). The annotated clay is placed back into the machine, which after scanning, detects the marks. We select each the mark and instruct the machine to extrude to the specified height and radius.

We then remove the object from the machine and add a directional-pad to the right-hand-side of the model. A few people try the design to ensure the pad can comfortably be reached. During this process, we accidentally smudge out a button (Figure 6.11, d). To fix this, we place the object back in the machine (Figure 6.11, e) and select the previous, undamaged version. This causes the machine to perform a local milling operation to clear away the damage (Figure 6.11, f), followed by extrusion to replace the button (Figure 6.11, g).

Lastly, we take the repaired model from the machine and draw a pen line around the edge of the shape to describe a lip. Back in the machine, the drawn path is detected and extruded down 2 mm. We then select 'export' (Figure 6.11, h) and send the finished model to a 3D printer for fabrication (Figure 6.9, a). The clay prototype is recycled.

6.6 Discussion

Through our implementation and design walkthroughs (see Section 6.5) we learn several practical lessons from realizing bidirectional fabrication. As both walkthroughs illustrate, ReForm is particularly well suited for design tasks that require a combination of precise and organic modeling. The game controller (see Section 6.5.2), for example, combines its organic and ergonomic shape with the precise button placement. Through ReForms synchronization of the digital model with the physical object, the physical rendition can be experienced in context (P4), which enables an iterative style of design. Both aspects lend themselves to product design, particularly when creating interactive devices.

ReForm's synchronization takes time however. In its current implementation, it took about one hour to design the smartwatch. Most of that time was spent synchronizing model and object (scanning, milling, or adding respectively). More efficient means of synchronization (see next section) or alternative implementations (see Section 6.6.2) could alleviate this increased time-demand.

6.6.1 Technical Limitations

Our current implementation solves model-object registration by fixing the object to a build-plate, thus enforcing a fixed reference frame. While this approach simplifies implementation, it also limits what users can do with the physical object — e.g. the side attached to the build-plate can not be modified. To do away with the buildplate, one could use the Iterative-Closest Point algorithm [150] or apply infrared registration markers to the model e.g. spraying a random dot pattern. However, being able to externally machine an object requires it to be held firmly in position.

The accuracy/fabrication-time trade-off can be tuned at runtime of the system, making it more flexible. If high accuracy is required, the more precise of the two fabrication methods can be used and the machine can move slower. If short fabrication times are desired, a more coarse fabrication method is used at higher speeds. For example, in our prototype subtractive operations are more precise than additive ones. Thus if accuracy is required, we can refine additively fabricated features subtractively. Due to tolerances of the fabrication process, we scan the object after each physical update and update the digital model accordingly. This can lead to an accumulative error, thus make the model degrade over time. A relaxed object/model correspondence, where only desired changes are integrated into the digital model [161], would remedy this problem.

Optical 3D scanners require all parts of the 3D model to be visible to them. Thus, concavities and hollow areas are difficult to capture. By integrating multiple 3D scanners, we could capture the physical object to a greater extent. Similarly, the digital fabrication stage is limited by what it can physically reach. Using all five axes for fabrication would increase the set of fabricable shapes, but also increase the algorithmic toolpath generation complexity.

6.6.2 Alternative Implementations

Other forms of implementing ReForm and bidirectional fabrication are possible. If only one fabrication method were automated, the other method could be performed manually e.g. computer controlled milling and manual material addition similar to *Sculpting by Numbers* [52]. Bidirectional fabrication could also be implemented by combining automated construction kit assembly (e.g. LEGO®) utilizing automated brick layout algorithms [162], and some shape-sensing capabilities integrated into the construction kit.

Multi-material printers could be used to implement a bidirectional fabrication process offering a whole new range of interactions. Malleable and hard materials in the same object could be used to express constraints. Built-in curvature sensors using printed optics [48] would make the artifact itself interactive, or even enable them to sense their own shape.

6.7 Summary

In this chapter we introduced *bidirectional fabrication*; a concept whereby digital and physical objects are entangled so that updates to one always propagate to the other. This enables users to design objects using precise repeatable digital operations, intuitive expressive physical actions, and combinations of both. By enabling designers to model through direct physical manipulation, we offer an intuitive form of interaction (P1) and foster direct engagement with the material (P5). Because the object-under-design exists not only as virtual model, but also as physical entity, users can integrate other existing physical objects into their designs (P3). Further, as the virtual 3D model is continuously synchronized with a physical rendition, users are afforded a sense of size (P2) and can explore the object-under-design in its target context (P4). In particular, we enable users to undo physical changes, bringing a feature which was previously exclusive to the digital domain, into the physical realm.

To evaluate the *bidirectional fabrication* concept, we built ReForm: a design system that blends digital modeling and physical shaping practice. We have implemented a ReForm prototype and shown application examples, demonstrating the novel interactions and benefits offered by our system. In lieu of computationally bidirectional materials, we have implemented external modification of objects. ReForm shows how bidirectional fabrication can be applied today, allowing researchers to explore interactions with such materials. In summary, this chapter contributed the concept of bidirectional fabrication and the ReForm system, including its interaction design and various technical contributions.

7 Discussion

While each individual chapter offers a discussion of the presented concepts and system in isolation, this chapter discusses aspects that bestride them. We discuss the implications of mixed physical/virtual design environments for different user-groups, for design processes and for the resulting artifacts. Further, we generalize the concept developed in this thesis to other domains: fabrication unrelated design, entertainment, and e-commerce.

The discussion of the implications of the concepts and systems developed in this thesis is followed by a reflection on the research methodology we employed. We relate our research-through-design approach to more empirical methods and broader design exploration. This leads us to a discussion of our evaluation methods. We then elaborate how our systems could be combined, and on the effect future technological developments will have on the contributions made in this thesis.

7.1 Implications of Mixed Design Environments

Bringing the physical and virtual world closer together when designing objects for digital fabrication, has implications for design processes, the audiences that employ these processes, and the objects that will be designed through them. In this section, we reflect on the implications for these three subjects.

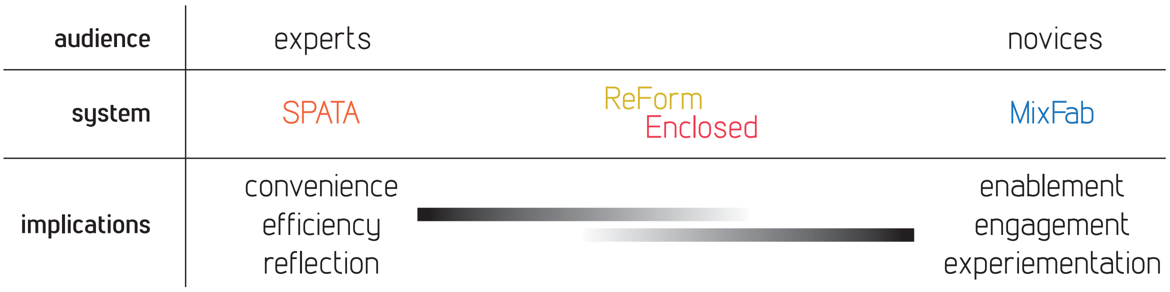


Fig. 7.1 Implications of the concepts/system of this dissertation for different user audiences.

7.1.1 Implications for User Audiences

The concepts we presented in this thesis have implications for different user-groups. Design and engineering experts (e.g., mechanical engineers or product designers) use established virtual design environments (see Section 2.3.1 for an overview). The concept of active spatio-tangible measurement tools for fabrication-aware design developed in chapter 4 integrates these environments. Through bi-directional tools we integrated physical measurement closely in existing virtual design environments, supporting existing workflows within these systems. This makes including physical measurements into new designs more convenient, efficient and less error prone. We have demonstrated the convenience gain in two walkthroughs using different design environments (see Section 4.3). To demonstrate the efficiency benefit of SPATA, we described a walkthrough (see Section 4.3.1) comparing the use of SPATA-based calipers with using traditional analog calipers in a mechanical design setting. Further, because SPATA tools can physically output virtual measurements, they enable the tangible reflection on previously made design decisions. Designers can e.g., use the SPATA calipers to get a sense of the size of the object-under-design.

On the other end of the spectrum are novices (see Figure 7.1), who have no/little training in design, engineering or digital fabrication. MixFab (see Chapter 5) enables this audience to design new objects, by interacting in a mixed-reality space. This *mixed-reality design for digital fabrication* approach introduces some familiarity to the design environment, as objects within MixFab can be directly manipulated, through gestures resembling the manipulation of physical entities. This direct form of interaction greatly eases the interaction with the design environment, and fosters engagement – as shown in the user-study, see Section 5.5. Further, users can effortlessly integrate physical objects in their designs, enabling them to experiment with existing items as they design new ones. As such, MixFab enables a new user-group to design for digital fabrication, while offering an engaging experience that invites experimentation and playfulness.

Enclosed (see Chapter 3) is aimed at non-fabrication experts developing prototype encasings using digital fabrication, but is also useful for experienced users who prefer convenience over expressive power. It enables novices to design enclosures by focusing the interaction on the components that need to be enclosed (making them *reference objects*, see Section 1.2). Our system ensures that the design result is fabricable, which enables users to engage in experimentation as fabrication-specifics are not longer a concern. Users can focus on the enclosure shape, rather than its fabrication. To that end, Enclosed can automatically generate the laser-cutting outlines necessary to fabricate the encasing. This generation adds the inter-panel connectors necessary to assemble the device once produced. For non-experts that removes the need to know about those connectors, lowering the barrier for using digital fabrication. For expert users, it makes designing laser-cut prototypes more convenient, as the tedious task of designing finger-joint connectors is automated.

The concept of *Bidirectional Fabrication* (see Chapter 6) is not specific to either audience. It has implications for skilled users, and for novices alike. Bidirectional fabrication continuously synchronizes a digital model with its physical rendition, and vise versa. Through this synchronization, we enable users to perform modeling actions in physical or digital space, depending on where they best performed. This is why the concept can be applied to the whole spectrum of user-audiences: the actions performed in physical or digital space will vary depending on the user group, however. Our ReForm system implementation is aimed towards novices, through its reduced AR-based UI. It offers simple, yet suggestive, digital operations which are easy to understand, but may be unsuited for expert users. Due to the clay material used by ReForm, audiences that are not familiar with digital modeling, are afforded an intuitive method of interacting with the object-under-design. Expert users, on the other hand are likely to benefit from the bidirectional synchronization as it enables iterative and participatory design (see next section). If ReForm were integrated in sophisticated CAD environments, similar to SPATA, users knowledgeable in those systems would benefit from the best of both worlds: powerful digital modeling, and iterative, physical, spatially intuitive, participatory bidirectional fabrication. Through ReForm, experts would be able to switch between both spaces as needed: operate in the physical world for organic modeling, use operations in the digital space for precise modification.

7.1.2 Implications for Design Processes

On one hand, SPATA and Enclosed take up and enhance existing, engineering-type design processes. SPATA makes current fabrication-aware design processes more convenient and efficient by easing the integration of physical aspects into new designs. This enables designers to more readily design around existing objects, as their measurement is less tedious. The physical output capabilities of *SPATA* enable more reflection, as the object-under-design is no longer a purely virtual entity, but in part becomes tangible. *Enclosed*, eases prototype development processes by integrating physical design aspects into the existing toolchain, and by removing fabrication-specific boilerplate tasks. This enables developers, designers and makers to focus on the shape of the device they are developing, rather than having to concern themselves with fabrication-specific details.

MixFab and *ReForm*, on the other hand, emphasize an iterative and participatory design process that is more playful than traditional engineering approaches. Both systems, and their underlying concepts, foster the direct engagement with the material and object-under-design. The former focuses on the integration of existing physical objects, which enables the playful combination of existing artifacts into new ones – during the MixFab user study, *U1* described this effortless replication of existing objects as "fun" (see Section 5.5.3). ReForm renders the object-under-design in a physical medium that affords playful and iterative interaction: clay. Through the game-controller walkthrough we have illustrated how ReForm's bidirectional synchronization fosters and enables participatory iteration. In that example, a group of designers creates a game-controller and physically experience it at different stages during the design process. Because there is no more hard separation between digital model and physical prototype, iteration and co-located collaboration becomes easier.

The playful and iterative design processes enabled by MixFab and ReForm might lead to a trial-and-error approach towards design, rather than a precise engineering one. During the early stages of design, when ideas are rough and malleable, that is a desired situation. However, such a process might lead to "design by coincidence"[1] where the design solution happens to work for one particular instance, but is not based on a thoroughly developed rationale that would ensure broader validity.

Embedding Design Values

The design environments we use to design objects not only dictate how we design, but also enforce a set of values on our final design solutions. Parametric design environments, for example favor smooth and regular shapes over chaotic organic ones. To those who create design environments, being able to encode values offers a chance to encourage "good" design. This idea is similar to Gross' notion of using code as a carrier for design knowledge: our design environments become carriers for design values. We can extend this notion to

[1]This term is used in analogy to "programming by coincidence" [163] where a program undergoes small, incremental changes or permutations, all of which are tested on a single, specific use-case.

two areas: the design process, and the resulting artifact.

Mehalik and Schunn present insight in what constitutes “good design” based on a meta-analysis of design processes [164]. Our own work is predominantly concerned with such good design process aspects. We emphasize the use of appropriate (physical) representations, and the exploration of measurement issues, integrate design constraints and enable the exploration of existing artifacts. This particular view however, ignores “meta-values” that do not directly affect the outcome of the design process, but its byproducts and execution. Material consumption is a good example: if we need to refabricate our object-under-design for each action along the design process, we will consume an insufferable amount of material, rendering the process unsustainable. This is a common criticism of interactive fabrication systems [80, 81]. We have addressed this particular meta-value in ReForm where, material removed during fabrication, can be reused again.

We can also influence the final artifact through our design environments. The aforementioned sustainability aspect for example, could be encouraged through design environments that nudge users into making material-saving decisions – as SPATA does when warning about increased fabrication resource demands (see Section 4.2.1). Using computational design methods we can support users in exploring the design space searching for more efficient solutions; efficient with respect to the objects fabrication (material used and fabrication time), as well as its final function. Enclosed, for example, attempts to minimize material use by optimizing the placement of the computed outlines on the material sheet (see Section 3.2.4).

7.1.3 Implications for Resulting Artifacts

All design systems prescribe the set of objects that can be designed with them. For example, with parametric design environments, it is next to impossible to model a human face accurately. Conversely, with mesh-based modeling tools creating accurate complex mechanical arrangements is very challenging. Besides the set of objects that can be designed, the design processes, such design environments incorporate and support, also prescribe a certain aesthetic of the resulting artifacts [4]. In this section we discuss what objects can be designed with the systems and concepts described in this thesis[2], and reflect on the prescribed aesthetics of these objects.

[2]This discussion also relates to the system usefulness i.e., that useful objects can be designed with them. As previously noted, we consider usefulness an evaluation criterion (see Section 1.3).

Object Classes and Constraints

The actions available to a designer in a given design environment govern what classes of objects can be designed. In addition, the concrete implementation of interaction concepts and design environments constrains the set of expressible objects further. For each of the four systems we presented in this thesis, we aimed to give an intuition as to what kind objects can be designed with them. Mainly through example objects, we demonstrate that these systems are expressive enough to produce meaningful objects. In the following, we want to discuss the constraints that are imposed predominantly by the implementation of design environments, and those inherent to the respective concepts which are thus unlikely to be removed entirely. As we move design environments closer to physical space, the constraints imposed by physicality will affect these environments (as opposed to entirely virtual environments which are devoid of physical constraints).

The closer we situate design environments to physical space, the more they will be affected by *spatial constraints*, meaning that size of objects that can be constructed will be limited. In virtual environments we can zoom to almost arbitrary levels; we can make very small things big enough for them to have discernible features, and make large structures small enough to view them to their full extent. Systems that maintain a one-to-one mapping between design space and physical space (e.g. MixFab or ReForm) can not offer this ability, and thus limit designers to objects sized so that they can still be manipulated. For example, a house would be far too big to be modeled in a meaningful way, as much as something of the size of a human cell would be far too small to be manipulated directly. In a similar vain are *resolution constraints*, which are imposed by the implementation of a system. To situate digital features in physical space, we need to render those features e.g., through augmented reality or digital fabrication. The concrete rendering method used, and its technical implementation enforce spatial limits e.g., we can not render decorative elements in arbitrary detail as we are limited by the available pixels or extrusion nozzle diameter of our render method implementation.

Our systems implement specific representations of the object-under-design. Enclosed presents an arrangement of flat surfaces and objects placed on them. SPATA, adds a tangible representation to the rendition offered by the design environments it integrates into. MixFab and ReForm present objects as atomic entities (without inner complexity or function). We opportunistically choose these representations to support the interaction concepts we are interested in. In case of the former two, the representation is prescribed by their design process: to place objects on fabricable planar panels, we must display those planar panels and represent the enclosure as such. SPATAs goal is to support design environments, and thus we adopt their representations. The latter two could also be implemented with more varied object representations, if the technology to capture them would be available. MixFab and ReForm both capture the shape of physical objects. Current 3D scanners can capture the outer surface of an object, but not its interior structure (with noted exceptions [123]). As such, we represent objects the way can capture them: as atomic solids. One could however, imagine a hierarchical representation where the outer surface maintains a link to the physical, and internal structure is defined independently and retrofit to the exterior. We hinted at such a relaxed model/object correspondence earlier, see 6.6.1.

Aesthetics and Style

The style and aesthetics of objects under design is determined by the process that is used to design them [4]. Two of our interaction concepts – mixed-reality design for digital fabrication and bidirectional fabrication – offer new, previously unknown design processes. The *reference object* concept does not impose a specific design process (and thus aesthetics), but our implementation does. As we limit the design to planar panels and make fabrication specific design decisions (such as panel connector placement) through an algorithmic process (we generate the fabrication outlines automatically, see Section 3.2.4), we prescribe a particular appearance of the resulting enclosures.

On one hand, objects designed with MixFab are fairly regular in nature (see Figure 5.1, c). They consist of flat surfaces approximating round shapes. Part of these looks can be attributed to the implementation, others are artifacts of the actions offered by MixFab's design process. The approximation of round shapes through flat surfaces is a result of MixFabs internal, mesh-based shape representation. To execute the CSG based modeling operations in real-time, we keep the complexity of the mesh models to a minimum, thus we approximate round shapes with few vertices. The regularity of the shapes however, is likely a conceptual artifact. MixFab offers three shape-defining operations: creation of cylinders & boxes, planar cuts and the additive/subtractive combination with existing objects. This set of operations is expressive (see Section 7.1.3), but taxes the creation of complex shapes as many steps have to be performed to create them. Thus, shapes designed with MixFab are likely to be unvaried and regular in nature, yet meaningful.

On the other hand, objects designed with ReForm are more organic and imprecise in nature (see Figure 6.9). They are characterized by flat top surfaces, and otherwise varied and irregular shapes. Again, we can hold the implementation of ReForm, as well as its concept, accountable. The imprecision, of for example cavities or holes created with ReForms digital operations, is a result of the prototypical implementation. For example, the implementation of the toolpath generation algorithm imposes several inaccuracies on the physical update process (see Section 6.4.3). The organic shapes of objects designed with ReForm can largely be attributed to the bidirectional fabrication design process, except for the smooth appearance which is likely due to the clay we used to implement ReForm. This new design process enables users to shape objects with their bare hands or sculpting tools, thus the resulting shapes are more likely to be organic, varied and imprecise as compared to a purely computer-modified shape.

Enclosed offers an interesting trade-off it that it prescribes two partially separable aesthetic aspects: the design process itself, and the computational design component that generates the outlines. The design process of Enclosed is focused on components being placed on planar panels. As such it prohibits the creation of e.g., round surfaces. This results in very edgy and mechanical looking enclosures. We, as the creators of Enclosed have thus become "meta-designers" in that we design the available aesthetics of future devices ahead of time. The panel connectors (finger joints) that connect the laser-cut panels also have an influence on the resulting enclosures appearance. As we discussed to some extend in section 3.4.2, other types of connectors could be used. Alternatively, we could implement connector placement strategies that try to hide connectors as much as possible.

7.1.4 Generalization to Other Domains

The concepts presented in this thesis generalize to domains other than digital fabrication. In the following we discuss the application of these concepts to other areas of design, entertainment (specifically games), and e-commerce.

The need for implementation-specific knowledge is prevalent in many areas of design. We have presented a system that frees users from that need when designing prototype enclosures (see Chapter 3). Our system automatically generates the laser-cutting outlines required to build an enclosure; laser-cutting outlines being the implementation of the enclosure. This idea of automatically generating an implementation from a higher-level description can be found in many design areas. In software engineering, model-driven development [165] frees users from implementation-specific knowledge by generating source-code from high-level models. In web-design, What-You-See-Is-What-You-Get (WYSIWYG) editors generate the HTML/CSS implementation of websites based on graphical descriptions thereof. Enclosed (see Chapter 3) is in the same vain, as it is a domain-specific version of the general “generate implementation from high-level description” concept.

We have developed two concepts/environments in which digital content and physical world co-exist. Such environments could also be used for entertainment purposes, specifically games. Using an immersive mixed-reality environment, where digital objects can interact with physical items – such as the one created by MixFab (see Chapter 5) – one could build new video-game platforms. Rather than producing images on a screen placed in front of the user, in such a mixed-reality space, users can interact with the content they are shown, directly. For example, in *Super Mario Bros.* players could reach into the game world and move obstacles out of the way. Conversely, existing objects could be introduced into games, to build and enrich the game world (similar to I.Ge [166]). E.g., users could place a cup in the mixed-reality game world to enable their character to reach the game goal. ReForm (see Chapter 6), through bidirectional fabrication, enables tangible modification of digital data. We could use this environment to create e.g., puzzles that are based on physical modification. In the bidirectional fabrication environment, a maze would be fabricated, which users have to modify using a predispensed amount of clay material, so that when virtual water (through the AR interface) is poured into the maze, the water does not flow out. Using the shape-input of bidirectional fabrication, we could capture, analyze and simulate the users solution. Using the AR interface, we would show the simulated water flowing into the maze. This is a new form of games, enabled through the bidirectional modification of physical material and digital data.

Integrated, active physical measurement tools could be used for online-shopping. For example, shoe-shops could give out dedicated foot measurement devices that can measure peoples feet, and submit that data to the online store. This would enable a more error-proof shopping experience, as customers could get shoes made to measure. In return, the device could instil a sense of size or shape of the shoes one is about to buy. Rather than just seeing a photograph, users could employ a mixed-reality presentation that combines AR with tangible, active measurement tools. A similar concept could be applied to fashion. Complementing optical estimation of body-sizes (which suffers from low precision [167]), connected measurement tools could offer a better shopping experience. Again, such experience could be combined with a mixed-reality approach to preview the clothes [167].

7.2 Research Methodology and Approach

We have opted for a research through design methodology. This choice greatly influences our contributions (i.e., concepts and systems, rather than studies and data), and our evaluation. In the following, we reflect on these methodological choices, as well as the combination of our concepts, and the impact of future technological development on the results of this thesis.

7.2.1 Reflections on Research Methodology

In this thesis we followed a research through design approach [13], which entails engineering and interaction-design work. Through this approach we developed novel concepts situated along the virtuality continuum. We implemented these concepts in prototype systems, yielding interaction concepts, technical innovation and working artifacts. Our engineering-focused approach allowed us to create four unique concepts. Each concept investigates the fabrication-design related problems created by the physical and digital world being disconnected. Other methodological stances, would lead to different outcomes and perspectives on the problems we set out to address. In the following, we reflect on alternative methodologies: empirical studies, and broader design exploration.

An empirical, reductionist treatment of this thesis' subject matter would produce narrower, but also deeper results (compared to the contributions we presented). Some of the physical/virtual disconnection problems (see Section 1.1) lend themselves well to controlled studies, others are more elusive. Investigating the spatial understanding of users in virtual environments for example, in fields other than HCI, is approached through empirical methods e.g., psycho-physical studies [168]. Such psycho-physical studies could be used to gain a deeper understanding of the impact of spatial misunderstanding during fabrication-aware design. One could investigate the "space compression effect" [7] observed in room-scale virtual environments with respect to digital design. Is this effect present in small-scale (1 m^3) environments? What effect do different display technologies have (stereoscopic vs. monoscopic rendering, flat screens vs. head-worn display)? Next to understanding such fundamental concerns, an qualitative approach could be used to understand users on a higher level. For example, finding out what objects novices want to design, so that we could build design environments to support those tasks.

Broader design exploration would yield a wider overview of the problem space, but at the expense of solutions for these issues. Applying design-led research methods, would give us an understanding of problems that elude empirical treatment. We could use cultural probes and ethnography to gauge the target contexts users design for, the requirements of these contexts and the persona of people involved. Our understanding of target-context would be extended from a spatial and technical perspective, by the cultural aspects users design for. Novices designing for a household setting, for example, will have different requirements for the design process, and for exploring the objects in said household, than experts designing parts for a robot. The first will likely favor aesthetics over function, the second will likely value function more. To investigate a different problem, using participatory design methods with artists (e.g., workshops, focus groups or structured interviews), would likely broaden the scope of what direct engagement with material during design, entails. Applying the same methods together with engineers might give us a better understanding of the role of existing objects in current design processes, and the scenarios in which their integration currently is a hindrance. Using the knowledge of domain experts, gained through such methods, we could also provide specialized design environments that enable non-experts to create complex objects (e.g., SketchChair which enables novices to design functional chairs [61]). Lastly, using interaction-design methods (e.g., wireframes, low-fidelity prototypes) we could explore a wider range of fabrication-design specific interaction concepts for virtual design environments. Specifically aiming at non-expert user-groups, such exploration would help to lower barriers for digital fabrication adoption.

Our engineering-focused research through design approach is situated in-between a empirical methodology and design exploration. We develop design-led solutions to the problems, but also provide empirical evidence towards the solutions utility. To choose which spatio-tangible tools to develop, we conducted a survey among practitioners (see Section 4.1). We designed the gesture-set used for the MixFab system through a user-defined gesture study (see Section 5.4). Our design solutions are developed in an integratory manner (meaning that we do not solve one problem after the other, but develop inclusive concepts that address a subset of problems). Through developing these solutions, we broaden our understanding of the problems, as each concept (and subsequent implementation) demonstrates how the respective problems can be addressed. Through walkthroughs and user-studies, we gain deeper insight into how to solve the problems at hand.

7.2.2 Validation and Evaluation

We evaluated the four concepts we developed in this thesis, by implementing them in prototype systems. Through this implementation, we develop the concepts themselves and demonstrate that the they are realizable. We evaluated the systems through walkthroughs and application examples. These methods demonstrate the benefits, but also weaknesses of the concepts and systems. They provided evidence that the systems are useful, as the systems can produce useful objects. For mixed-reality design environments (and its MixFab implementation, see Chapter 5), we claimed that it lowers the barrier for design novices. To back this claim, we evaluated the system in a user-study. More generally though, the implementation quality inherent in prototypes, renders usability evaluation unsuitable. User studies with our systems would likely yield implementation deficits, rather than insight in the underlying concept [14] – a trend that we have observed in the MixFab user study.

However, combining our system with an empirical approach could answer more specific questions however – for details refer to the previous discussion.

7.2.3 Combining the Systems

The concepts and systems developed in this thesis have been described atomically, mitigating their individual set of problems caused by the physical/digital divide. While we presented these solutions as indivisible, parts of these systems are interchangeable or can be combined.

Venturing into computational design, the generation of fabrication-specific aspects as found in *Enclosed* (see Chapter 3), generalizes beyond domain-specific design environments. Current CAD tools, for example support dedicated design wizards that encode engineering knowledge into code. Autodesk Inventor, for example, offers a range of "design accelerators" that can design bolted connections, gear arrangements and mechanical shafts. Such wizards could be offered for digital-fabrication specific aspects e.g., generate to the connectors between two planar components.

The active spatio-tangible measurement tools (see Chapter 4) are prime candidates to be combined with our other systems. In MixFab, the SPATA calipers and protractor would enable a form of tactile feedback. Using the calipers inside the mixed-reality volume, for example, could physically output the size of the object-under-design enabling tactile exploration of the otherwise solely visually rendered object. Utilizing the SPATA tools as tangible prop, we would do away with mid-air gestures and offer a mixed-reality interface that relies on tangible user-interaction, rather than symbolic one. Technically, a more precise caliper implementation could be used to refine the at the moment crude 3D scanning abilities offered by current depth sensors. Accurately remeasuring specific points of a previously attained crude 3D scan should make the scanned shape more precise.

Similarly, the SPATA tools could be integrated in *ReForm* (see Chapter 6). Whenever users need to input precise dimensions, these tools could be used. For example, when drilling a hole, the user would mark the location of the hole on the physical object, and subsequently use the SPATA calipers to measure the diameter of the hole based on either or an existing item, or their own liking. Another example is flattening off objects: the calipers would first move to the current height of the object, and could then be used to set the new flattening height. In both cases, we would further the integration of existing physical dimensions (P3) and allow for exploration of the design choice (P4) before it is made.

7.2.4 Impact of Future Technological Development

We have presented concepts, and their implementations. The former will benefit from ongoing technological development. The latter, participates in said development, and will eventually be superseded by it; much like any technology it will become obsolete. It is thus the concepts, and the technological advancements (not the technology itself), and the insights we have gained from their development that will persist.

The fabrication-aware design concepts will be refined themselves, and so will their implementation become more sophisticated. Current research topics illustrate this trend: low-fi fabrication [86] for example, lowers the time required to fabricate an object. Using this technology, we could improve the implementation of *bidirectional fabrication* (see Chapter 6) and lower the time required to synchronize the virtual model and physical object. Advances in augmented-reality displays will enhance mixed-reality design environments (see Chapter 5), as will better gesture recognition and faster 3D scanning. With future sensors and actuators, Spatio-Tangible Tools (see Chapter 4) could be implemented for a multitude of physical properties e.g., reflectance, elasticity/hardness, even function and mechanical movement. This development would coincide with the development of, and the support for, multi-material printers. Concepts closer to the virtual side of the virtuality continuum (see Section 1.2) will also benefit technological advancement. As computational design methods evolve, so will our abilities to encode expert design knowledge in software.

8 Conclusion

We set out to explore the closer connection of the physical world into digital fabrication-aware design processes and environments. Following this aim we developed new interaction concepts (see Section 1.2) aligned with the virtuality continuum. We implemented these interaction concepts in novel systems, demonstrating the concepts' viability and making various technical contributions. Each concept, and subsequent implementation, addresses problems caused by the disconnection of the physical and virtual space. In this chapter we summarize the contributions of this thesis, from a systems perspective, and from a design process view. We conclude by discussing future work that arises from this thesis.

8.1 Contributions

This thesis has made four major contributions to the field of HCI in the domain of design environments for digital fabrication. These contributions are aligned with the concepts and their implementations presented herein. These are:

1. **The concept and implementation of *reference objects*** which relates virtual models to physical objects at the designers disposal, which aids spatial judgment (P2) of designers. We developed the *Enclosed* system, which eases the design of prototype enclosures. This system closely integrates the components that need to be enclosed, as reference objects. To free users from the for fabrication-specific knowledge, we

contribute a novel laser-cutter outline generation algorithm for encasings. Prior to this work, designers had to manually integrate the components they want to enclose in their design process (e.g., through manual measurements). Further, this tool frees users from having to manually design the joints necessary to assemble the 3D encasing from the 2D laser-cut parts.

2. **The concept and implementation of *Active Spatio-Tangible Measurement Tools* (SPATA tools)** which integrates physical measurement tools into digital design environments. These tools can automatically transfer physically measured values to virtual design environments, and vise versa: virtually measured values can be tangibly output in physical space. We contribute two such tools: a digital adaptation of calipers and of bevel protractors. Further, did we contribute the integration of both tools into three design environments commonly used for fabrication-aware design: parametric modeling, mesh-based modeling and 2D design (e.g., laser-cutting or circuit board design). Lastly, we demonstrate both tools and their integration in three application examples that highlight the benefits of the bi-directional information transfer and design environment specific task support.

3. **The concept and implementation of *mixed-reality design for digital fabrication*** which situates the design of objects in a hybrid space, where digital and physical objects co-exist. We proposed and implemented implement an immersive mixed-reality environment by combining an augmented reality setup, gesture recognition and 3D scanning capabilities. Further, did we contribute a set of user-defined gestures for 3D modeling obtained through a study in which we observe how users would perform basic tasks unconstrained by any system or augmentation. We then presented MixFab's design environment, which is based on these gestures. It is centered around direct and natural interaction with virtual artifacts, effortless integration of physical objects into the design process and a self-explanatory interface. Lastly, we evaluated MixFab's design decisions in a user study and provided evidence that, this system enables novices to design meaningful objects.

4. **The concept and implementation of *bidirectional fabrication***, which continuously synchronizes a digital model and physical object. On top of this core concept, we built ReForm, a system to design objects using manual physical modification, as well as precise digital operations. We contributed the first *bidirectional Fabrication* implementation, which combines additive and subtractive fabrication, a structured-light 3D scanner, a custom-built computer-numerically controlled (CNC) five-axis motion plat-

form actuating a custom clay mill and extruder, and a purpose-built, back-projected AR display. We made numerous technical contributions, specifically the use of two-state polymer clay for interactive fabrication systems and a novel toolpath generation algorithm to update the physical object. Lastly, we demonstrated the benefits afforded by ReForm through application examples.

8.2 Progress Towards Addressing the Problems

In this thesis we explored different concepts the connect the physical world and virtual space in digital fabrication design environments. The prior disconnection of physical and digital space caused several problems (with respect to fabrication-aware design; see Section 1.1). In the following we reflect on the various concepts presented in this thesis which address the individual problems – thus, progress we have made towards alleviating those problems:

(P1) ***Difficult interaction with virtual spaces*** Interacting with 3D objects in virtual spaces is difficult, as it is often unclear how to manipulate them. We have presented a range of different options towards easing the interaction with virtual design environments. Enclosed eased interaction by reducing the set of available actions, focusing on domain-specific operations (e.g., place component on enclosure) that yield fabricable results. The UI is comprised of four views (top/bottom, left/right, front/back, 3D) which reduces the need for navigating the virtual space. The Active Spatio-Tangible Measurement Tools can be used as tangible proxies for object being designed in virtual space, thus easing that virtual object's manipulation. Mixed-Reality Design for Digital Fabrication employs direct gestural interaction, with the gestures derived from a user-study. It offers intuitive manipulation of virtual objects. Lastly, Bidirectional-Fabrication affords the direct tangible modification of a physical clay rendition of digital models, thus takes interaction out of the virtual realm.

(P2) ***Spatial understanding is hampered*** because spatial judgment, depth perception as well as size and depth perception are difficult in virtual environments. Towards this problem, we have explored reference objects, spatio-tangible output of length and angle, augmented reality and physical fabrication. Reference Objects ease spatial judgment, as users can investigate the relationship between virtual objects using their physical counterparts. Active Spatio-Tangible Measurement Tools physically reproduce dimensions and angles of virtual objects, enabling users to obtain a sense of size. A mixed-reality design environment creates a space that is identified with the physical world, so that virtual objects being designed are displayed in their correct size. Existing objects interact with virtual ones, and thus serve as reference. Bidirectional Fabrication replicates object-under-design in physical space, where spatial understanding is not a problem.

The concepts illustrated above offer apparent benefits regarding spatial understand as all of them, except the mixed-reality approach, tangibly relate spatiality back to the physical space. Yet, we did not study the effects of our concepts in an empirical manner, with the exception of the mixed-reality concept where we performed a user-study. Gathering empirical data about the spatial understanding of users when given physical references (e.g., through passive reference objects or active measurement tools) would enable the refinement of the concepts listed above.

(P3) ***Lack of physical artifact integration*** makes design decisions based on existing objects difficult. We have demonstrated a spectrum of concepts that integrate physical artifacts into the fabrication-aware design process: from annotated 3D models (see Chapter 3), to automated physical measurements (see Chapter 4), to 3D scanning (see Chapters 5 and 6). These methods differ in their flexibility (3D models are prescribed, 3D scanning integrates shapes from arbitrary objects), implementation complexity (3D models are simple, 3D scanning is difficult), infrastructure requirements (3D models require disk storage space, 3D scanning requires specialized hardware), and run-time requirements (3D models can be loaded quickly, 3D scanning requires minutes to complete).

(P4) ***In-context exploration is limited*** which impedes the exploration design decisions that are to be made and hinders reflection on previous choices. Through our engineering-led approach, we have developed concepts that enable exploration at varying levels of fidelity. Using Spatio-Tangible Measurement Tools designers can physically output dimensions and angles of the object-under-design, and evaluate those choices in the physical space, also with respect to existing artifacts. Bidirectional Fabrication produces the object-under-design as physical object, which is not bound to a location, and thus can be inspected and evaluated in its target space.

We have demonstrated a broad range of in-context exploration concepts, primarily focused on physicality as generic target space. However, a design-based exploration of target contexts (what spaces do users design for?) would offer a broader understanding of specific target environments and their requirements. For example, how does one explore /evaluate designs when creating furniture compared to creating toys for children? The former, might involve visualizing the furniture in a living room, placing existing objects on it, and seating people around. The latter, would be on a smaller scale and likely require require tangible exploration. Investigating the specific target environments users design for, will lead to better in-process exploration.

(P5) ***Lack of direct engagement with the material*** hinders an engaging design experience. We have explored two ways of creating engagement with the object-under-design, both by enabling its direct modification. In chapter 5 we used gestures to allow users to directly modify objects displayed through an AR interface. While users described this method as immersive and engaging (see Section 5.5.3), it remained intangible. Bidirectional Fabrication creates a physical rendition of the object-under-design. This enables tangible engagement with the object in a material that affords direction modification: clay.

Next to our integrative approach, artistic exploration of this problem [10, 83, 169] will yield further insight in how this problem can be alleviated, drawing from the experience of manual fabrication practitioners.

(P6) ***Digital-Fabrication and engineering knowledge is required*** which presents a barrier for beginners, and an inconvenience for experts. Enclosed can automatically produce the fabrication-plans (laser-cutting outlines) for prototype enclosures enclosures. This includes the inter-panel joints and material-thickness compensation which designers would otherwise have to know about and design themselves.

This is the problem we least explored within this thesis. Integrating fabrication-specific knowledge into design environments, is for the better part, an algorithmic challenge. Computational design methods enable us to assert, extend and create fabrication-specific properties of user's designs. For additive manufacture, for example, methods exist that asses if a 3D model can be fabricated that way [101]. Future algorithmic treatment of this problem, combined with interaction concepts that integrate them, will alleviate this issue further. In section 8.3.2 we discuss future work that incorporates these aspects.

8.3 Future Work

This thesis opens up and explores the idea of bringing digital design closer to the physical world, and vise versa. We have designed and implemented four concepts guided by the virtuallity continuum [11]. Besides these concepts – or guiding principles – there are other possibilities to bring by such integration. This section outlines research programs that would explore other avenues of bringing digital design world and physical space closer together.

8.3.1 Shape-Change and Digital Fabrication

Shape changing devices are artifacts which have a surface that is in some way articulate, meaning that its spatial domain can be modulated. The most common form of articulation is linear Z-actuation where the device surface is approximated by poles that can move along one axis. A sufficient number of such poles, when arranged densely enough, can manipulate physical objects [170] and closely approximate 3D objects. Other forms of actuation more directly approximate surfaces through an actuated regular tiling of a surface [171].

This class of devices is, much like digital fabrication, a means for the programmatic manipulation of matter. Unlike digital fabrication however, that manipulation is not permanent, but rather a dynamic process. For example, the articulated poles of *Emerge* [172] can dynamically render multiple physical bar charts after another, whereas with digital fabrication new material is required for each physicalized bar chart [173]. However, in essence both concepts (digital fabrication and shape change) belong to the same class of concepts, bringing us closer towards the *universal computer* [174] that can manipulate not only symbols, but also the matter around it.

It thus seems like an interesting avenue to explore the direct combination of digital fabrication and shape-change. Both concepts can be intertwined at various stages of the digital fabrication process: for fabrication-aware design, during fabrication itself, or to produce new shape-changing abilities.

For Fabrication-Aware Design

The obvious use of shape-changing displays in fabrication-aware design processes is displaying the object under design. As shape-changing displays not only show color, but also shape, the object being designed is effectively made physical reality without being fabrication (subject to the display's fidelity and abilities). *MixFab* (see Chapter 5) and *ReForm* (see Chapter 6) are based on that very principle (physically represent the object being designed). Using for example *inFORM* [170], users could tangibly inspect the model and even form it by directly interacting with the poles that physically render the model. In order to develop a better understanding of how to utilize shape-change in fabrication-aware design processes, we would need to examine questions such as: what the shape-change properties that matter for the design process are – resolution/fidelity alone? What kinds of interactions are enabled by this technology, and how do they support design processes for which user groups?

In lieu of the self-actuating materials required for high-resolution shape-change, hybrid fabrication environments that combine additive and subtractive fabrication such as *ReForm*, could be considered shape-changing devices. Compared to other shape-changing systems we do not actuate rigid components, or assemble atomic building blocks, but rather shape reusable material in a cycle of adding and removing material to "render" each frame. Such a system has a very low frame rate (in the order of frames per hour) and requires environmental augmentation (namely the system itself), but offers very high fidelity. We are not bound by clumsy actuators, but can shape matter with the quality of digital fabrication machines.

During Fabrication

Shape-change could be used to make fabrication faster. For example a print bed that supports the FDM printing process (see Section 2.2.1) would speed up the process considerably, as no support material would have to be printed. At the same time, such a system would only be applicable to overhangs where there is no printed material in the volume beneath them, as otherwise the poles of the build platform could not reach the object. Non-linear support structures alleviate this problem [106], and such the question becomes what forms of shape-change best support and implement such structures.

Using high-fidelity shape-change can enable new forms of digital fabrication. Suppose we had a device that could (with high fidelity) produce molding structures e.g., through articulated surfaces forming the walls of the mold. With such a device, we could make mold-based fabrication techniques (such as injection molding [175] or rotation molding [176]) part of the digital fabrication repertoire. Currently, mold-based fabrication processes require the prior fabrication of the mold itself. Replacing such a static fixture with a dynamic shape-changing device would make such processes more flexible and faster. Key challenges will be the mechanical strength (injection molds have to withstand high pressure) and required fidelity of the shape-changing device.

New Shape-Changing Abilities

Additive fabrication methods will likely bring about new forms of shape-change. Recent advances in these fabrication processes enable new devices, not only to be built using digital fabrication, but to be built at all. Multi-material printers can combine various materials at fabrication time to produce varying physical properties in the same object. For example, one can print an object that has well-defined elastic behavior when bending along one axis, but remains stiff along another. Digital fabrication machines can embed electronics into additively fabricated objects[1]. Walters and McGoran have embedded shape-memory alloys [177] into 3D printed objects.

Computational design methods for efficiently navigating the design space of such future artifacts might even enable domain and application specific shape-changing solutions. While the goal of *programmable matter* [178] still seems far away, the combination of algorithmic design and new fabrication techniques will enable specialized shape-changing devices of high quality, dexterity and fidelity.

8.3.2 Interactive Computational Design

Design is the search of a satisfying solution to a constrained problem [4]. This search is often in the form of exploration of a design space through the designer [179]. The systems that we have presented in this thesis support designers in this endeavor by easing design decisions and iteration. However, designers still must observe the difference between the current state and the desired situation , and develop a strategy to bring about the target state. For example, if we want to have our phone standing upright on our desk[2] we need to describe an object that holds the phone in place; all using the actions and operations offered by our design environment of choice.

Computational design methods (some of which we survey in section 2.3.7) analyze the initial situation and compute a series of actions that results in the desired one: they algorithmically encode a design process. This is utmost convenient as now we are no longer left with describing a solution that brings about our desired target state, but we merely need to describe that target state itself (e.g., that the phone should be held upright, rather than describing the object that does so). Note, that the aesthetics and other properties of the computed solutions depend on the encoded design process. We could consider e.g. aesthetics a part of the target state (e.g., the phone be held upright by a spherical artifact) and it is up to the creators of such computational design processes to decide where that border lies. In analogy, consider imperative vs functional programming; the former describes the procedure to solve a problem, the latter describes the problem solution itself. Initially two completely schools of thought, their concepts have since been combined to new classes of programming languages where developers can move between both forms of specifications.

[1]http://www.voxel8.co/

[2]The current state is that the phone can not stand upright unless we hold it in that position. In the desired state the phone is held in place at a certain angle by an object other than our own hand.

Combining computational design with tangible user interfaces or mixed-reality, in a personal fabrication context, will give rise to new interaction paradigms. Freeing users from having to devise design strategies, but enabling them to easily describe the state they want to create, will enable new user groups to utilize digital fabrication. Consider the phone dock example: after specifying that the phone should come to rest in a certain position (e.g., by holding it in place and having the system capture that), and possibly specifying other constraints (e.g., desired max size, colors, material, texture) the system would generate a range of designs from which users can then choose. Their choice in turn imposes further design constraints. Research questions revolve around adequate constraint specification methods, solution representations, and search strategies. Such a research program would likely borrow greatly from the existing body of knowledge in the respective domains (HCI, computational design, operations research, design research).

Acronyms

ABS Acrylonitrile Butadiene Styrene. 17

AR Augmented Reality. 11, 90, 92, 101–103, 121, 127, 134, 136

CAD Computer Aided Design. 3, 11, 13, 15, 22, 24, 34, 35, 50, 52, 67, 69, 82, 90, 92, 121, 130

CSG Constructive Solid Geometry. 71, 125

FDM Fused Deposition Modeling. 16, 31, 34, 138

HCI Human-Computer Interaction. v, 11, 13, 22, 34, 35, 132

IDE integrated development environment. 35–38, 45

PLA Polylactic Acid. 17

SLA Stereolithohraphy. 17

SLS Selective Laser Sintering. 17

TUI Tangible User Interface. 11, 14, 15, 34

UI user interface. 6, 10, 11, 34, 40, 41, 49, 71, 92, 121, 134

WYSIWYG What-You-See-Is-What-You-Get. 126

References

[1] C. G. Jung, *Psychologische Typen.* Walter Verlag, Olten und Freiburg im Breisgau, 13th ed., 1978.

[2] F. Rengier, A. Mehndiratta, H. von Tengg-Kobligk, C. Zechmann, R. Unterhinninghofen, H.-U. Kauczor, and F. Giesel, "3D printing based on imaging data: review of medical applications," *International Journal of Computer Assisted Radiology and Surgery*, vol. 5, no. 4, pp. 335–341, 2010.

[3] L. Sun, T. Fukuda, T. Tokuhara, and N. Yabuki, "Differences in spatial understanding between physical and virtual models," *Frontiers of Architectural Research*, vol. 3, no. 1, pp. 28 – 35, 2014.

[4] H. A. Simon, *The Sciences of the Artificial (3rd Ed.).* Cambridge, MA, USA: MIT Press, 1996.

[5] D. Wynn and J. Clarkson, "Models of designing," in *Design process improvement* (J. Clarkson and C. Eckert, eds.), pp. 34–59, Springer London, 2005.

[6] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst, "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques," *Computer Graphics Forum*, vol. 17, no. 3, pp. 41–52, 1998.

[7] E. Kruijff, J. Swan, and S. Feiner, "Perceptual issues in augmented reality revisited," in *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pp. 3–12, Oct 2010.

[8] M. McCullough, *Abstracting Craft: The Practiced Digital Hand.* Cambridge, MA, USA: MIT Press, 1996.

[9] D. Ramduny-Ellis, J. Hare, A. Dix, and S. Gill, "Exploring Physicality in the Design Process," in *Undisciplined! Design Research Society Conference 2008*, July 2009.

[10] A. Zoran and J. A. Paradiso, "FreeD: a freehand digital sculpting tool," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, (New York, NY, USA), pp. 2613–2616, ACM, 2013.

[11] P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *IEICE Transactions Information Systems*, vol. E77-D, pp. 1321–1329, Dec. 1994.

[12] J. Zimmerman, J. Forlizzi, and S. Evenson, "Research Through Design As a Method for Interaction Design Research in HCI," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, (New York, NY, USA), pp. 493–502, ACM, 2007.

[13] W. Gaver, "What Should We Expect from Research Through Design?," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 937–946, ACM, 2012.

[14] S. Greenberg and B. Buxton, "Usability Evaluation Considered Harmful (Some of the Time)," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, (New York, NY, USA), pp. 111–120, ACM, 2008.

[15] J. A. Jacko, *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Third Edition.* Boca Raton, FL, USA: CRC Press, Inc., 3rd ed., 2012.

[16] M. Buchenau and J. F. Suri, "Experience Prototyping," in *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '00, (New York, NY, USA), pp. 424–433, ACM, 2000.

[17] H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, (New York, NY, USA), pp. 234–241, ACM, 1997.

[18] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell, "Passive Real-world Interface Props for Neurosurgical Visualization," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, (New York, NY, USA), pp. 452–458, ACM, 1994.

[19] B. Fröhlich and J. Plate, "The Cubic Mouse: A New Device for Three-dimensional Input," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, (New York, NY, USA), pp. 526–531, ACM, 2000.

[20] G. Pangaro, D. Maynes-Aminzade, and H. Ishii, "The Actuated Workbench: Computer-controlled Actuation in Tabletop Tangible Interfaces," in *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, UIST '02, (New York, NY, USA), pp. 181–190, ACM, 2002.

[21] J. Lee, R. Post, and H. Ishii, "ZeroN: mid-air tangible interaction enabled by computer controlled magnetic levitation," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, (New York, NY, USA), pp. 327–336, ACM, 2011.

[22] Excalibur Electronics, Inc., *Phantom Force - Auto-Motion Intelligent Response Chess.* Excalibur Electronics, Inc., Excalibur Electronics, Inc., 13755 SW 119th Av, Miami, Florida 33186 U.S.A., 2015.

[23] D. Nowacka and D. Kirk, "Tangible Autonomous Interfaces (TAIs): Exploring Autonomous Behaviours in TUIs," in *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '14, (New York, NY, USA), pp. 1–8, ACM, 2013.

[24] H. Ishii, D. Lakatos, L. Bonanni, and J.-B. Labrune, “Radical Atoms: Beyond Tangible Bits, Toward Transformable Materials,” *interactions*, vol. 19, pp. 38–51, Jan. 2012.

[25] S. Greenberg and C. Fitchett, “Phidgets: Easy Development of Physical Interfaces Through Physical Widgets,” in *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST ’01, (New York, NY, USA), pp. 209–218, ACM, 2001.

[26] R. Ballagas, M. Ringel, M. Stone, and J. Borchers, “iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’03, (New York, NY, USA), pp. 537–544, ACM, 2003.

[27] B. Hartmann, S. R. Klemmer, M. Bernstein, L. Abdulla, B. Burr, A. Robinson-Mosher, and J. Gee, “Reflective physical prototyping through integrated design, test, and analysis,” in *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST ’06, (New York, NY, USA), pp. 299–308, ACM, 2006.

[28] S. E. Hudson and J. Mankoff, “Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil and masking tape,” in *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST ’06, (New York, NY, USA), pp. 289–298, ACM, 2006.

[29] N. Villar, J. Scott, S. Hodges, K. Hammil, and C. Miller, “.NET Gadgeteer: A Platform for Custom Devices,” in *Pervasive Computing* (J. Kay, P. Lukowicz, H. Tokuda, P. Olivier, and A. Krüger, eds.), vol. 7319 of *Lecture Notes in Computer Science*, pp. 216–233, Springer Berlin Heidelberg, 2012.

[30] A. Knörig, R. Wettach, and J. Cohen, “Fritzing: a tool for advancing electronic prototyping for designers,” in *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI ’09, (New York, NY, USA), pp. 351–358, ACM, 2009.

[31] J. Harrop and R. Gordon, *3D Printing 2015-2025: Technologies, Markets, Players*. IDTechEx, 2015.

[32] N. Gershenfeld, “How to Make Almost Anything: The Digital Fabrication Revolution,” *Technology and Foreign Policy*, vol. 91, pp. 43–57, 2012.

[33] D. Pham and R. Gault, “A comparison of rapid prototyping technologies,” *International Journal of Machine Tools and Manufacture*, vol. 38, no. 10-11, pp. 1257 – 1287, 1998.

[34] J. G. Tanenbaum, A. M. Williams, A. Desjardins, and K. Tanenbaum, “Democratizing Technology: Pleasure, Utility and Expressiveness in DIY and Maker Practice,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, (New York, NY, USA), pp. 2603–2612, ACM, 2013.

[35] R. Shewbridge, A. Hurst, and S. K. Kane, “Everyday Making: Identifying Future Uses for 3D Printing in the Home,” in *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS ’14, (New York, NY, USA), pp. 815–824, ACM, 2014.

[36] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, and A. Bowyer, “RepRap–the replicating rapid prototyper,” *Robotica*, vol. 29, no. 01, pp. 177–191, 2011.

[37] W. H. Oskay and L. M. Edman, “The CandyFab Project,” 2008.

[38] B. Hopkin, “Benefits of positional five-axis machining,” *Moldmaking Technology*, 2005.

[39] D. Dragomatz and S. Mann, “A classified bibliography of literature on NC milling path generation,” *Computer-Aided Design*, vol. 29, no. 3, pp. 239 – 247, 1997.

[40] S. Mueller, B. Kruck, and P. Baudisch, “LaserOrigami: Laser-cutting 3D Objects,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, (New York, NY, USA), pp. 2585–2592, ACM, 2013.

[41] K. Fukuchi, K. Jo, A. Tomiyama, and S. Takao, “Laser cooking: a novel culinary technique for dry heating using a laser cutter and vision technology,” in *Proceedings of the ACM multimedia 2012 workshop on Multimedia for cooking and eating activities*, CEA ’12, (New York, NY, USA), pp. 55–58, ACM, 2012.

[42] J. Schoning, Y. Rogers, and A. Kruger, “Digitally Enhanced Food,” *IEEE Pervasive Computing*, vol. 11, pp. 4–6, July 2012.

[43] R. A. Khot, R. Pennings, and F. F. Mueller, “EdiPulse: Turning Physical Activity Into Chocolates,” in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’15, (New York, NY, USA), pp. 331–334, ACM, 2015.

[44] S. E. Hudson, “Printing Teddy Bears: A Technique for 3D Printing of Soft Interactive Objects,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’14, (New York, NY, USA), pp. 459–468, ACM, 2014.

[45] H. Peng, J. Mankoff, S. E. Hudson, and J. McCann, “A Layered Fabric 3D Printer for Soft Interactive Objects,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, (New York, NY, USA), pp. 1789–1798, ACM, 2015.

[46] M. Vázquez, E. Brockmeyer, R. Desai, C. Harrison, and S. E. Hudson, “3D Printing Pneumatic Device Controls with Variable Activation Force Capabilities,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, (New York, NY, USA), pp. 1295–1304, ACM, 2015.

[47] V. Savage, C. Chang, and B. Hartmann, “Sauron: embedded single-camera sensing of printed physical user interfaces,” in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, UIST ’13, (New York, NY, USA), pp. 447–456, ACM, 2013.

[48] K. Willis, E. Brockmeyer, S. Hudson, and I. Poupyrev, “Printed optics: 3D printing of embedded optical elements for interactive devices,” in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST ’12, (New York, NY, USA), pp. 589–598, ACM, 2012.

[49] E. Brockmeyer, I. Poupyrev, and S. Hudson, “PAPILLON: designing curved display surfaces with printed optics,” in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, UIST ’13, (New York, NY, USA), pp. 457–462, ACM, 2013.

[50] V. Savage, R. Schmidt, T. Grossman, G. Fitzmaurice, and B. Hartmann, “A Series of Tubes: Adding Interactivity to 3D Prints Using Internal Pipes,” in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST ’14, (New York, NY, USA), pp. 3–12, ACM, 2014.

[51] A. Zoran, “Hybrid basketry: interweaving digital practice within contemporary craft,” in *ACM SIGGRAPH 2013 Art Gallery*, SIGGRAPH ’13, (New York, NY, USA), pp. 324–331, ACM, 2013.

[52] A. Rivers, A. Adams, and F. Durand, “Sculpting by Numbers,” *ACM Transactions on Graphics*, vol. 31, pp. 157:1–157:7, Nov. 2012.

[53] A. Rivers, I. E. Moyer, and F. Durand, “Position-correcting tools for 2D digital fabrication,” *ACM Transactions on Graphics*, vol. 31, pp. 88:1–88:7, July 2012.

[54] A. A. G. Requicha and J. R. Rossignac, “Solid Modeling and Beyond,” *IEEE Computer Graphics and Applications*, vol. 12, pp. 31–44, Sept. 1992.

[55] R. Woodbury, *Elements of parametric design*. Routledge, 2010.

[56] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge, “Sketch-based modeling: A survey,” *Computers & Graphics*, vol. 33, no. 1, pp. 85 – 103, 2009.

[57] K. D. Willis, J. Lin, J. Mitani, and T. Igarashi, “Spatial sketch: bridging between movement &#38; fabrication,” in *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, TEI ’10, (New York, NY, USA), pp. 5–12, ACM, 2010.

[58] G. Johnson, M. Gross, E. Y.-L. Do, and J. Hong, “Sketch it, make it: sketching precise drawings for laser cutting,” in *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, CHI EA ’12, (New York, NY, USA), pp. 1079–1082, ACM, 2012.

[59] K. Hildebrand and M. Alexa, “Sketch-based pipeline for mass customization,” in *ACM SIGGRAPH 2013 Talks*, SIGGRAPH ’13, (New York, NY, USA), pp. 37:1–37:1, ACM, 2013.

[60] M. D. Gross, “Now More Than Ever : Computational thinking and a science of design,” *Special issue of Japanese Society for Science of Design*, vol. 16, no. 2, pp. 50–54, 2009.

[61] G. Saul, M. Lau, J. Mitani, and T. Igarashi, "SketchChair: an all-in-one chair design system for end users," in *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, (New York, NY, USA), pp. 73–80, ACM, 2011.

[62] Y. Mori and T. Igarashi, "Plushie: an interactive design system for plush toys," *ACM Transactions on Graphics*, vol. 26, July 2007.

[63] M. Skouras, B. Thomaszewski, P. Kaufmann, A. Garg, B. Bickel, E. Grinspun, and M. Gross, "Designing Inflatable Structures," *ACM Transactions on Graphics*, vol. 33, pp. 63:1–63:10, July 2014.

[64] B. Thomaszewski, S. Coros, D. Gauge, V. Megaro, E. Grinspun, and M. Gross, "Computational Design of Linkage-based Characters," *ACM Transactions on Graphics*, vol. 33, pp. 64:1–64:9, July 2014.

[65] A. Schulz, A. Shamir, D. I. W. Levin, P. Sitthi-amorn, and W. Matusik, "Design and Fabrication by Example," *ACM Transactions on Graphics*, vol. 33, pp. 62:1–62:11, July 2014.

[66] G. Saul, C. Xu, and M. D. Gross, "Interactive paper devices: end-user design & fabrication," in *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, TEI '10, (New York, NY, USA), pp. 205–212, ACM, 2010.

[67] J. McCrae, N. Umetani, and K. Singh, "FlatFitFab: Interactive Modeling with Planar Sections," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, (New York, NY, USA), pp. 13–22, ACM, 2014.

[68] D. Anderson, J. L. Frankel, J. Marks, A. Agarwala, P. Beardsley, J. Hodgins, D. Leigh, K. Ryall, E. Sullivan, and J. S. Yedidia, "Tangible interaction + graphical interpretation: a new approach to 3D modeling," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, (New York, NY, USA), pp. 393–402, ACM Press/Addison-Wesley Publishing Co., 2000.

[69] K.-J. Wu and M. D. Gross, "TOPAOKO: interactive construction kit," in *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, (New York, NY, USA), pp. 3619–3624, ACM, 2010.

[70] Y. Huang and M. Eisenberg, "Easigami: virtual creation by physical folding," in *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, (New York, NY, USA), pp. 41–48, ACM, 2012.

[71] S. Schkolne, M. Pruett, and P. Schröder, "Surface drawing: creating organic 3D shapes with the hand and tangible tools," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, (New York, NY, USA), pp. 261–268, ACM, 2001.

[72] M. Lau, M. Hirose, A. Ohgawara, J. Mitani, and T. Igarashi, "Situated modeling: a shape-stamping interface with tangible primitives," in *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI '12, (New York, NY, USA), pp. 275–282, ACM, 2012.

[73] D. Anderson, J. L. Frankel, J. Marks, D. Leigh, E. Sullivan, J. Yedidia, and K. Ryall, "Building virtual structures with physical blocks," in *Proceedings of the 12th annual ACM symposium on User interface software and technology*, UIST '99, (New York, NY, USA), pp. 71–72, ACM, 1999.

[74] S. Follmer and H. Ishii, "KidCAD: digitally remixing toys through tangible tools," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 2401–2410, ACM, 2012.

[75] S. Follmer, D. Carr, E. Lovell, and H. Ishii, "CopyCAD: remixing physical objects with copy and paste from the real world," in *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, (New York, NY, USA), pp. 381–382, ACM, 2010.

[76] M. Reed, "Prototyping digital clay as an active material," in *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, (New York, NY, USA), pp. 339–342, ACM, 2009.

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

[78] S. Cho, Y. Heo, and H. Bang, "Turn: a virtual pottery by real spinning wheel," in *ACM SIGGRAPH 2012 Emerging Technologies*, SIGGRAPH '12, (New York, NY, USA), pp. 25:1–25:1, ACM, 2012.

[79] M. Gannon, T. Grossman, and G. Fitzmaurice, "Tactum: A Skin-Centric Approach to Digital Design and Fabrication," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), pp. 1779–1788, ACM, 2015.

[80] K. D. Willis, C. Xu, K.-J. Wu, G. Levin, and M. D. Gross, "Interactive fabrication: new interfaces for digital fabrication," in *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, TEI '11, (New York, NY, USA), pp. 69–72, ACM, 2011.

[81] S. Mueller, P. Lopes, and P. Baudisch, "Interactive construction: interactive fabrication of functional mechanical devices," in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, (New York, NY, USA), pp. 599–606, ACM, 2012.

[82] H. Song, F. Guimbretière, C. Hu, and H. Lipson, "ModelCraft: capturing freehand annotations and edits on physical 3D models," in *Proceedings of the 19th annual ACM symposium on User interface software and technology*, UIST '06, (New York, NY, USA), pp. 13–22, ACM, 2006.

[83] A. Zoran, R. Shilkrot, and J. Paradiso, "Human-computer interaction for hybrid carving," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, UIST '13, (New York, NY, USA), pp. 433–440, ACM, 2013.

[84] S. Mueller, T. Mohr, K. Guenther, J. Frohnhofen, and P. Baudisch, “faBrickation: Fast 3D Printing of Functional Objects by Integrating Construction Kit Building Blocks,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’14, (New York, NY, USA), pp. 3827–3834, ACM, 2014.

[85] D. Beyer, S. Gurevich, S. Mueller, H.-T. Chen, and P. Baudisch, “Platener: Low-Fidelity Fabrication of 3D Objects by Substituting 3D Print with Laser-Cut Plates,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, (New York, NY, USA), pp. 1799–1806, ACM, 2015.

[86] S. Mueller, S. Im, S. Gurevich, A. Teibrich, L. Pfisterer, F. Guimbretière, and P. Baudisch, “WirePrint: 3D Printed Previews for Fast Prototyping,” in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST ’14, (New York, NY, USA), pp. 273–280, ACM, 2014.

[87] B. Koo, W. Li, J. Yao, M. Agrawala, and N. J. Mitra, “Creating Works-like Prototypes of Mechanical Objects,” *ACM Transactions on Graphics*, vol. 33, pp. 217:1–217:9, Nov. 2014.

[88] J. S. Sadar and G. Chyon, “3D Scanning and Printing As a New Medium for Creativity in Product Design,” in *Procedings of the Second Conference on Creativity and Innovation in Design*, DESIRE ’11, (New York, NY, USA), pp. 15–20, ACM, 2011.

[89] F. Blais, “Review of 20 years of range sensor development,” *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 231–243, 2004.

[90] G. Taubin, D. Moreno, and D. Lanman, “3D Scanning for Personal 3D Printing: Build Your Own Desktop 3D Scanner,” in *ACM SIGGRAPH 2014 Studio*, SIGGRAPH ’14, (New York, NY, USA), pp. 27:1–27:66, ACM, 2014.

[91] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, “KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera,” in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST ’11, (New York, NY, USA), pp. 559–568, ACM, 2011.

[92] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[93] M. Lau, G. Saul, J. Mitani, and T. Igarashi, “Modeling-in-context: User Design of Complementary Objects with a Single Photo,” in *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM ’10, (Aire-la-Ville, Switzerland, Switzerland), pp. 17–24, Eurographics Association, 2010.

[94] K. Xu, H. Zheng, H. Zhang, D. Cohen-Or, L. Liu, and Y. Xiong, “Photo-inspired model-driven 3D object modeling,” in *ACM SIGGRAPH 2011 papers*, SIGGRAPH ’11, (New York, NY, USA), pp. 80:1–80:10, ACM, 2011.

[95] T. Weyrich, J. Lawrence, H. Lensch, S. Rusinkiewicz, and T. Zickler, “Principles of Appearance Acquisition and Representation,” in *ACM SIGGRAPH 2008 Classes*, SIGGRAPH ’08, (New York, NY, USA), pp. 80:1–80:119, ACM, 2008.

[96] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger, “Real-time Non-rigid Reconstruction Using an RGB-D Camera,” *ACM Transactions on Graphics*, vol. 33, pp. 156:1–156:12, July 2014.

[97] D. K. Pai, K. van den Doel, D. L. James, J. Lang, J. E. Lloyd, J. L. Richmond, and S. H. Yau, “Scanning Physical Interaction Behavior of 3D Objects,” in *ACM SIGGRAPH 2005 Courses*, SIGGRAPH ’05, (New York, NY, USA), ACM, 2005.

[98] D. Ceylan, W. Li, N. J. Mitra, M. Agrawala, and M. Pauly, “Designing and Fabricating Mechanical Automata from Mocap Sequences,” *ACM Transactions on Graphics*, vol. 32, pp. 186:1–186:11, Nov. 2013.

[99] J. I. Echevarria, D. Bradley, D. Gutierrez, and T. Beeler, “Capturing and Stylizing Hair for 3D Fabrication,” *ACM Transactions on Graphics*, vol. 33, pp. 125:1–125:11, July 2014.

[100] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev, “3D Self-portraits,” *ACM Transactions on Graphics*, vol. 32, pp. 187:1–187:9, Nov. 2013.

[101] Q. Zhou, J. Panetta, and D. Zorin, “Worst-case structural analysis,” *ACM Transactions on Graphics*, vol. 32, pp. 137:1–137:12, July 2013.

[102] A. Telea and A. Jalba, “Voxel-based assessment of printability of 3D shapes,” in *Proceedings of the 10th international conference on Mathematical morphology and its applications to image and signal processing*, ISMM’11, (Berlin, Heidelberg), pp. 393–404, Springer-Verlag, 2011.

[103] O. Stava, J. Vanek, B. Benes, N. Carr, and R. Měch, “Stress relief: improving structural strength of 3D printable objects,” *ACM Transactions on Graphics*, vol. 31, pp. 48:1–48:11, July 2012.

[104] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, and X. Liu, “Cost-effective Printing of 3D Objects with Skin-frame Structures,” *ACM Transactions on Graphics*, vol. 32, pp. 177:1–177:10, Nov. 2013.

[105] L. Lu, A. Sharf, H. Zhao, Y. Wei, Q. Fan, X. Chen, Y. Savoye, C. Tu, D. Cohen-Or, and B. Chen, “Build-to-last: Strength to Weight 3D Printed Objects,” *ACM Transactions on Graphics*, vol. 33, pp. 97:1–97:10, July 2014.

[106] J. Dumas, J. Hergel, and S. Lefebvre, “Bridging the Gap: Automated Steady Scaffoldings for 3D Printing,” *ACM Transactions on Graphics*, vol. 33, pp. 98:1–98:10, July 2014.

[107] R. Hu, H. Li, H. Zhang, and D. Cohen-Or, “Approximate Pyramidal Shape Decomposition,” *ACM Transactions on Graphics*, vol. 33, pp. 213:1–213:12, Nov. 2014.

[108] L. Luo, I. Baran, S. Rusinkiewicz, and W. Matusik, “Chopper: partitioning models into 3D-printable parts,” *ACM Transactions on Graphics*, vol. 31, pp. 129:1–129:9, Nov. 2012.

[109] K. Hildebrand, B. Bickel, and M. Alexa, “crdbrd: Shape Fabrication by Sliding Planar Slices,” *Comp. Graph. Forum*, vol. 31, pp. 583–592, May 2012.

[110] X.-Y. Li, C.-H. Shen, S.-S. Huang, T. Ju, and S.-M. Hu, “Popup: automatic paper architectures from 3D models,” *ACM Transactions on Graphics*, vol. 29, pp. 111:1–111:9, July 2010.

[111] S. Xin, C.-F. Lai, C.-W. Fu, T.-T. Wong, Y. He, and D. Cohen-Or, “Making burr puzzles from 3D models,” *ACM Transactions on Graphics*, vol. 30, pp. 97:1–97:8, July 2011.

[112] Y. Zhou, S. Sueda, W. Matusik, and A. Shamir, “Boxelization: Folding 3D Objects into Boxes,” *ACM Transactions on Graphics*, vol. 33, pp. 71:1–71:8, July 2014.

[113] C. Schüller, D. Panozzo, and O. Sorkine-Hornung, “Appearance-mimicking Surfaces,” *ACM Transactions on Graphics*, vol. 33, pp. 216:1–216:10, Nov. 2014.

[114] M. Lau, A. Ohgawara, J. Mitani, and T. Igarashi, “Converting 3D furniture models to fabricatable parts and connectors,” *ACM Transactions on Graphics*, vol. 30, pp. 85:1–85:6, July 2011.

[115] K. Vidimče, S.-P. Wang, J. Ragan-Kelley, and W. Matusik, “OpenFab: a programmable pipeline for multi-material fabrication,” *ACM Transactions on Graphics*, vol. 32, pp. 136:1–136:12, July 2013.

[116] D. Chen, D. I. W. Levin, P. Didyk, P. Sitthi-Amorn, and W. Matusik, “Spec2Fab: a reducer-tuner model for translating specifications to 3D prints,” *ACM Transactions on Graphics*, vol. 32, pp. 135:1–135:10, July 2013.

[117] B. Bickel, M. Bächer, M. A. Otaduy, H. R. Lee, H. Pfister, M. Gross, and W. Matusik, “Design and Fabrication of Materials with Desired Deformation Behavior,” *ACM Transactions on Graphics*, vol. 29, pp. 63:1–63:10, July 2010.

[118] M. Skouras, B. Thomaszewski, S. Coros, B. Bickel, and M. Gross, “Computational design of actuated deformable characters,” *ACM Transactions on Graphics*, vol. 32, pp. 82:1–82:10, July 2013.

[119] M. Bächer, B. Bickel, D. L. James, and H. Pfister, “Fabricating Articulated Characters from Skinned Meshes,” *ACM Transactions on Graphics*, vol. 31, pp. 47:1–47:9, July 2012.

[120] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, “Computational design of mechanical characters,” *ACM Transactions on Graphics*, vol. 32, pp. 83:1–83:12, July 2013.

[121] R. Prévost, E. Whiting, S. Lefebvre, and O. Sorkine-Hornung, “Make it stand: balancing shapes for 3D fabrication,” *ACM Transactions on Graphics*, vol. 32, pp. 81:1–81:10, July 2013.

[122] M. Bächer, E. Whiting, B. Bickel, and O. Sorkine-Hornung, “Spin-it: Optimizing Moment of Inertia for Spinnable Objects,” *ACM Transactions on Graphics*, vol. 33, pp. 96:1–96:10, July 2014.

[123] K. D. D. Willis and A. D. Wilson, “InfraStructs: fabricating information inside physical objects for imaging in the terahertz region,” *ACM Transactions on Graphics*, vol. 32, pp. 138:1–138:10, July 2013.

[124] T. Oster and J. Borchers, “VisiCut: An Application Genre for Lasercutting in Personal Fabrication,” tech. rep., RWTH Aachen University, 2011.

[125] D. Saakes, T. Cambazard, J. Mitani, and T. Igarashi, “PacCAM: material capture and interactive 2D packing for efficient material usage on CNC cutting machines,” in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, UIST ’13, (New York, NY, USA), pp. 441–446, ACM, 2013.

[126] M. Gardiner, C. Lindinger, R. Haring, H. Hörtner, H. Ogawa, and E. Ogawa, “Social brainstorming via interactive fabrication,” in *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, ACE ’11, (New York, NY, USA), pp. 76:1–76:2, ACM, 2011.

[127] H. Ogawa, M. Mara, C. Lindinger, M. Gardiner, R. Haring, D. Stolarsky, E. Ogawa, and H. Hörtner, “Shadowgram: a case study for social fabrication through interactive fabrication in public spaces,” in *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, TEI ’12, (New York, NY, USA), pp. 57–60, ACM, 2012.

[128] S. Houben and C. Weichel, “Overcoming Interaction Blindness Through Curiosity Objects,” in *CHI ’13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’13, (New York, NY, USA), pp. 1539–1544, ACM, 2013.

[129] I. Posch, H. Ogawa, C. Lindinger, R. Haring, and H. Hörtner, “Introducing the FabLab as interactive exhibition space,” in *Proceedings of the 9th International Conference on Interaction Design and Children*, IDC ’10, (New York, NY, USA), pp. 254–257, ACM, 2010.

[130] N. Gershenfeld, *Fab: The Coming Revolution on Your Desktop–from Personal Computers to Personal Fabrication*. Basic Books, 2005.

[131] B. Nissen and J. Bowers, “Data-Things: Digital Fabrication Situated Within Participatory Data Translation Activities,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, (New York, NY, USA), pp. 2467–2476, ACM, 2015.

[132] H. W. Lin, L. Aflatoony, and R. Wakkary, “Design for One: A Game Controller for a Quadriplegic Gamer,” in *Proceedings of the Extended Abstracts of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI EA ’14, (New York, NY, USA), pp. 1243–1248, ACM, 2014.

[133] A. Lodi, S. Martello, and D. Vigo, “Recent Advances on Two-dimensional Bin Packing Problems,” *Discrete Appl. Math.*, vol. 123, pp. 379–396, Nov. 2002.

[134] J. E. G. Coffman, M. R. Garey, D. S. Johnson, and R. E. Tarjan, “Performance Bounds for Level-Oriented Two-Dimensional Packing Algorithms,” *SIAM Journal on Computing*, vol. 9, no. 4, pp. 808–826, 1980.

[135] S. Schneegass, A. Sahami Shirazi, T. Döring, D. Schmid, and A. Schmidt, "NatCut: An Interactive Tangible Editor for Physical Object Fabrication," in *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, (New York, NY, USA), pp. 1441–1446, ACM, 2014.

[136] J. Lee, V. Su, S. Ren, and H. Ishii, "HandSCAPE: A Vectorizing Tape Measure for On-site Measuring Applications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, (New York, NY, USA), pp. 137–144, ACM, 2000.

[137] J. Ou, L. Yao, D. Tauber, J. Steimle, R. Niiyama, and H. Ishii, "jamSheets: Thin Interfaces with Tunable Stiffness Enabled by Layer Jamming," in *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '14, (New York, NY, USA), pp. 65–72, ACM, 2013.

[138] R. Niiyama, L. Yao, and H. Ishii, "Weight and Volume Changing Device with Liquid Metal Transfer," in *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '14, (New York, NY, USA), pp. 49–52, ACM, 2013.

[139] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson, "HoloDesk: direct 3d interactions with a situated see-through display," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 2421–2430, ACM, 2012.

[140] H. Benko, R. Jota, and A. Wilson, "MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 199–208, ACM, 2012.

[141] H. Nishino, K. Utsumiya, and K. Korida, "3D Object Modeling Using Spatial and Pictographic Gestures," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '98, (New York, NY, USA), pp. 51–58, ACM, 1998.

[142] I. Llamas, B. Kim, J. Gargus, J. Rossignac, and C. D. Shaw, "Twister: a space-warp operator for the two-handed editing of 3D shapes," in *ACM SIGGRAPH*, pp. 663–668, 2003.

[143] C. Holz and A. Wilson, "Data Miming: Inferring Spatial Object Descriptions from Human Gesture," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, (New York, NY, USA), pp. 811–820, ACM, 2011.

[144] H. Kim, G. Albuquerque, S. Havemann, and D. W. Fellner, "Tangible 3D: Hand Gesture Interaction for Immersive 3D Modeling," in *Proceedings of the 11th Eurographics Conference on Virtual Environments*, EGVE'05, (Aire-la-Ville, Switzerland, Switzerland), pp. 191–199, Eurographics Association, 2005.

[145] K. Oka, Y. Sato, and H. Koike, "Real-time fingertip tracking and gesture recognition," *Computer Graphics and Applications, IEEE*, vol. 22, no. 6, pp. 64–71, 2002.

[146] V. I. Pavlovic, R. Sharma, and T. S. Huang, “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 677–695, July 1997.

[147] A. D. Wilson, “Using a Depth Camera As a Touch Sensor,” in *ACM International Conference on Interactive Tabletops and Surfaces*, ITS ’10, (New York, NY, USA), pp. 69–72, ACM, 2010.

[148] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112–122, 1973.

[149] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly Media, Incorporated, 2008.

[150] Z. Zhang, “Iterative Point Matching for Registration of Free-form Curves and Surfaces,” *Int. J. Comput. Vision*, vol. 13, pp. 119–152, Oct. 1994.

[151] S. Rusinkiewicz, “Estimating Curvatures and Their Derivatives on Triangle Meshes,” in *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2Nd International Symposium*, 3DPVT ’04, (Washington, DC, USA), pp. 486–493, IEEE Computer Society, 2004.

[152] G. Taubin, “A Signal Processing Approach to Fair Surface Design,” in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’95, (New York, NY, USA), pp. 351–358, ACM, 1995.

[153] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The Quickhull Algorithm for Convex Hulls,” *ACM Trans. Math. Softw.*, vol. 22, pp. 469–483, Dec. 1996.

[154] T. Starner, B. Leibe, D. Minnen, T. Westyn, A. Hurst, and J. Weeks, “The perceptive workbench: Computer-vision-based gesture tracking, object tracking, and 3D reconstruction for augmented desks,” *Machine Vision and Applications*, vol. 14, no. 1, pp. 59–71, 2003.

[155] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn, “FingARtips: Gesture Based Direct Manipulation in Augmented Reality,” in *Proceedings of the 2Nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, GRAPHITE ’04, (New York, NY, USA), pp. 212–221, ACM, 2004.

[156] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, “User-defined Gestures for Surface Computing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’09, (New York, NY, USA), pp. 1083–1092, ACM, 2009.

[157] D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis, and P. Olivier, “Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor,” in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST ’12, (New York, NY, USA), pp. 167–176, ACM, 2012.

[158] G. C. Loney and T. M. Ozsoy, "NC machining of free form surfaces," *Computer-Aided Design*, vol. 19, no. 2, pp. 85 – 90, 1987.

[159] P. Kulkarni and D. Dutta, "An accurate slicing procedure for layered manufacturing," *Computer-Aided Design*, vol. 28, no. 9, pp. 683 – 697, 1996.

[160] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, vol. 96, pp. 226–231, AAAI Press, 1996.

[161] A. Hattab and G. Taubin, "3D Modeling by Scanning Physical Modifications," in *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*, Aug 2015.

[162] R. Testuz, Y. Schwartzburg, and M. Pauly, "Automatic Generation of Constructable Brick Sculptures," in *Eurographics 2013 - Short Papers* (M.-A. Otaduy and O. Sorkine, eds.), The Eurographics Association, 2013.

[163] A. Hunt and D. Thomas, *The Pragmatic Programmer: From Journeyman to Master*. Pearson Education, 1999.

[164] M. Mehalik and C. Schunn, "What constitutes good design? A review of empirical studies of design processes," *International Journal of Engineering Education*, vol. 22, no. 3, p. 519, 2007.

[165] D. C. Schmidt, "Guest editor's introduction: Model-driven engineering," *Computer*, vol. 39, no. 2, pp. 25–31, 2006.

[166] P. Oswald, J. Tost, and R. Wettach, "I.Ge: Exploring New Game Interaction Metaphors with Interactive Projection," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, (New York, NY, USA), pp. 733–738, ACM, 2015.

[167] M. Yuan, I. Khan, F. Farbiz, S. Yao, A. Niswar, and M.-H. Foo, "A Mixed Reality Virtual Clothes Try-On System," *Multimedia, IEEE Transactions on*, vol. 15, pp. 1958–1968, Dec 2013.

[168] H. H. Bülthoff and H. A. van Veen, "Vision and Action in Virtual Environments: Modern Psychophysics in Spatial Cognition Research," in *Vision and Attention* (M. Jenkin and L. Harris, eds.), pp. 233–252, Springer New York, 2001.

[169] H. Peng, A. Zoran, and F. V. Guimbretière, "D-Coil: A Hands-on Approach to Digital 3D Models Design," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), pp. 1807–1815, ACM, 2015.

[170] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii, "inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, (New York, NY, USA), pp. 417–426, ACM, 2013.

[171] A. Roudaut, A. Karnik, M. Löchtefeld, and S. Subramanian, “Morphees: Toward High "Shape Resolution" in Self-actuated Flexible Mobile Devices,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, (New York, NY, USA), pp. 593–602, ACM, 2013.

[172] F. Taher, J. Hardy, A. Karnik, C. Weichel, Y. Jansen, K. Hornbæk, and J. Alexander, “Exploring Interactions with Physically Dynamic Bar Charts,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, (New York, NY, USA), pp. 3237–3246, ACM, 2015.

[173] Y. Jansen, P. Dragicevic, and J.-D. Fekete, “Evaluating the Efficiency of Physical Visualizations,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’13, (New York, NY, USA), pp. 2593–2602, ACM, 2013.

[174] I. E. Sutherland, “The Ultimate Display,” in *Proceedings of the IFIP Congress*, pp. 506–508, 1965.

[175] D. Rosato and M. Rosato, *Injection Molding Handbook*. Springer US, 2012.

[176] R. Todd, D. Allen, and L. Alting, *Manufacturing Processes Reference Guide*. Industrial Press, 1994.

[177] P. Walters and D. McGoran, “Digital fabrication of "smart" structures and mechanisms-creative applications in art and design,” in *NIP & Digital Fabrication Conference*, pp. 185–188, Society for Imaging Science and Technology, 2011.

[178] S. Goldstein, J. Campbell, and T. Mowry, “Programmable matter,” *Computer*, vol. 38, pp. 99–101, May 2005.

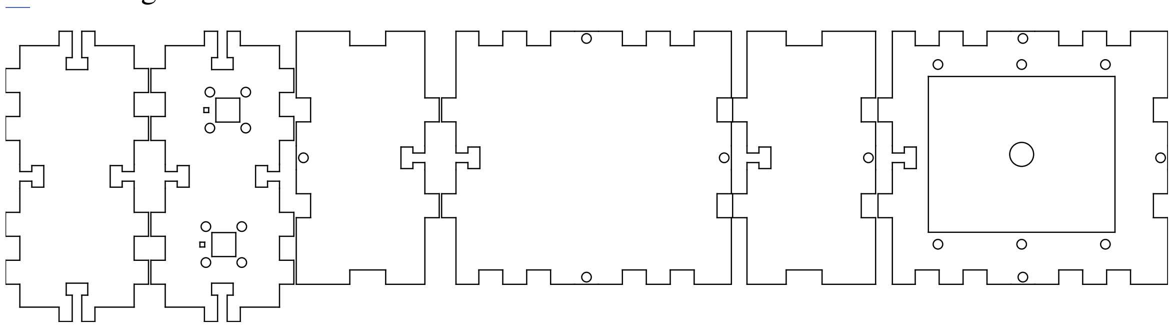
[179] R. F. Woodbury and A. L. Burrow, “Whither Design Space?,” *Artif. Intell. Eng. Des. Anal. Manuf.*, vol. 20, pp. 63–82, Apr. 2006.

Supplementary Material

A.1 Enclosed

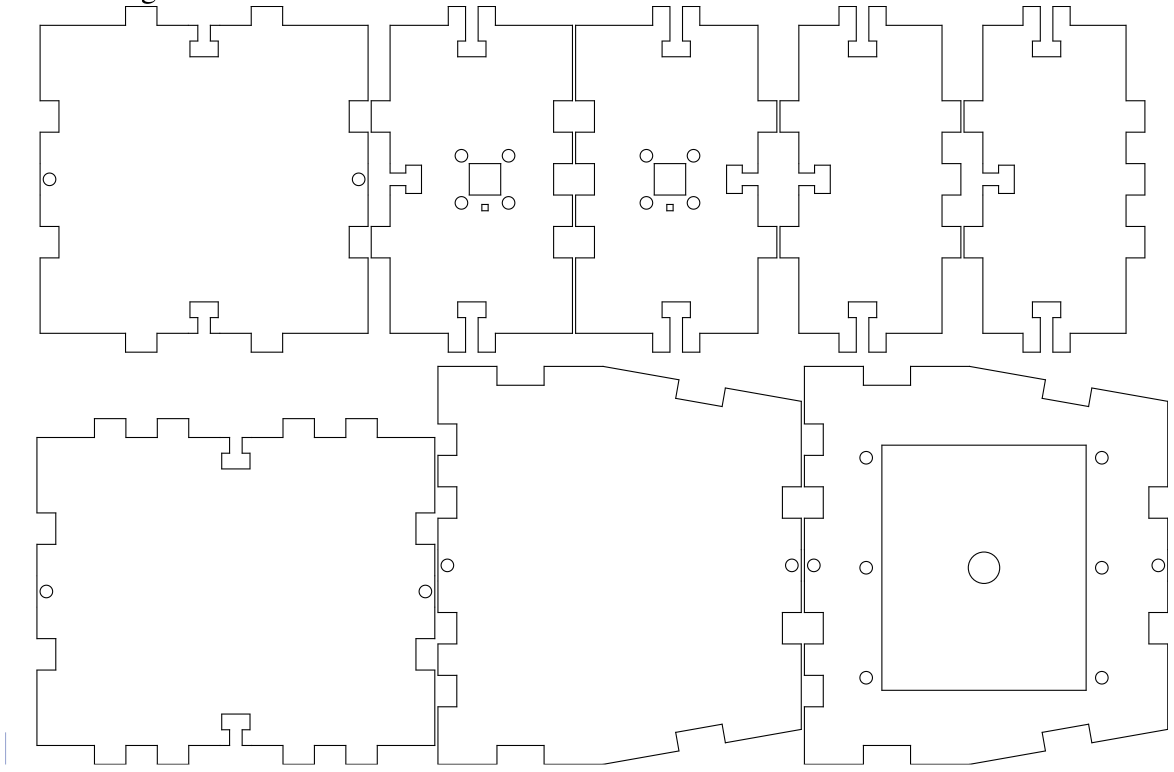
A.1.1 Generated Outlines: Brick

These are the laser-cutter outlines for the *Brick* prototype (see Figure 3.6) as automatically generated by our Enclosed system. The blue line is for scale reference, and is originally 10 mm long.



A.1.2 Generated Outlines: Pool

These are the laser-cutter outlines for the *Pool* prototype (see Figure 3.6) as automatically generated by our Enclosed system. The blue line is for scale reference, and is originally 10 mm long.



A.2 SPATA

A.2.1 Middleware Source Code Example

In the following, we aim to give an intuition of the nature of SPATA's middleware layer. To this end, we list excerpts from the integration of SPATA calipers into Autodesk Inventor.

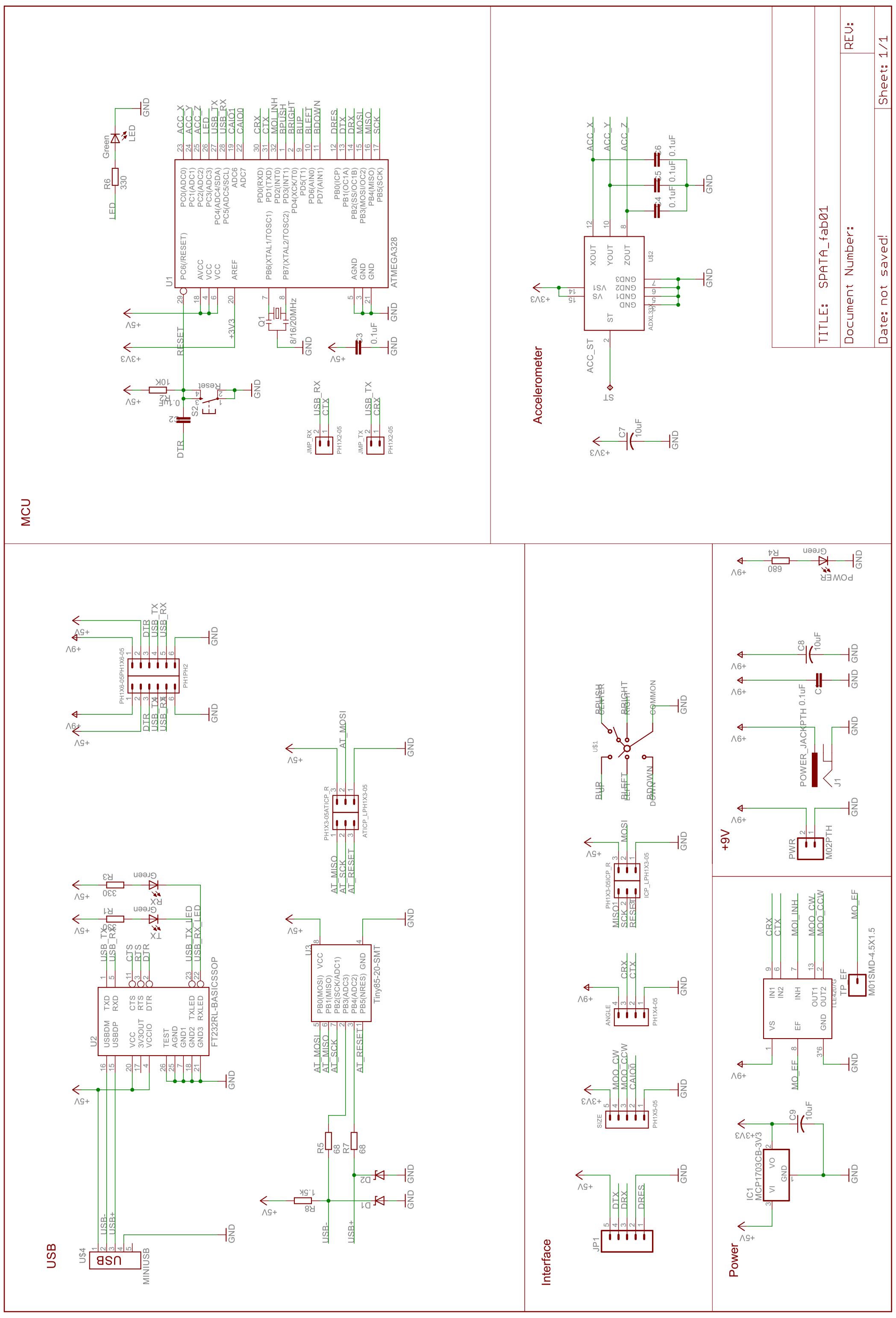
```
// All tool/design environment connecting classes implement this connector interface,
// and are then automatically set up using our middleware layer (setup method is called).
public class CalipersAndInventor : IToolAndEnvironmentConnector<Calipers, InventorClient>
{
    private Calipers calipers;
    private InventorClient inventor;

    // Sets up the connection between calipers and Autodesk Inventor
```

```
public void Setup(Calipers calipers, InventorClient inventor)
{
    this.calipers = calipers;
    this.inventor = inventor;

    // Add a workflow to Inventor. This will show up as button on the SPATA toolbar ribbon
    // our Inventor client implementation creates.
    inventor.AddWorkflow("Create Box", OnInventorCreateBox);

    // Register for tool orientation changes, so that we update the model view accordingly.
    calipers.OrientationChanged += OnToolOrientationChanged;
}

// Called when the user selects the create box workflow from within Inventor.
// This method runs in its own thread, so it can block without affecting any
// other system behavior.
public void OnInventorCreateBox()
{
    var doc = inventor.GetActiveDocument();
    var startingPlane = inventor.QueryUserForPlane();
    bool continueCreatingBoxes = true;

    while(continueCreatingBoxes)
    {
        // the tools do not constantly send their measurements, to not clog up
        // the serial communication channel. We enable, and disable the position
        // updates as needed.
        calipers.SetPositionUpdateEnabled(true);

        // The devices can store multiple images, each with a unique ID. This is
        // a global namespace that has to be managed by implementors. We choose to
        // give each application a range (e.g., Inventor has from 10 - 30). The
        // images available to each environment are stored as constants in an enum.
        calipers.Display.ShowImage((int) Calipers.DisplayImages.CreateBoxWidth);

        // WaitForButtonConfirmation() is a method provided by all SPATA tools. It
        // blocks until the user presses the button down, and then returns the last
        // measurement value of the tool (if position updates were previously enabled).
        var width = calipers.WaitForButtonConfirmation();

        calipers.Display.ShowImage((int) Calipers.DisplayImages.CreateBoxHeight);
        var height = calipers.WaitForButtonConfirmation();
        calipers.Display.ShowImage((int) Calipers.DisplayImages.CreateBoxLength);
        var length = calipers.WaitForButtonConfirmation();
        calipers.SetPositionUpdateEnabled(false);

        // Now we create the box on the current starting plane
        var size = new Vector3(width, height, length);

        // After creating the box, its top face becomes the new starting plane
        startingPlane = inventor.CreateBox(doc, startingPlane, size);

        // Lastly, we check if the user wants to continue creating boxes
```

```
            calipers.Display.ShowImage((int) Calipers.DisplayImages.ContinueCreatingBoxes);
            continueCreatingBoxes = calipers.WaitForButtonInput() == Button.Down;
        }

        calipers.Display.Clear();
    }

    // This method would update the view orientation accordingly (if appropriate)
    protected void OnToolOrientationChanged(Quaternion orientation)
    {
    }

}
```

A.2.2 Common Electronics Schematics

Both our SPATA tools (see Section 4.1) share the same hardware platform. The following are the schematics of the printed circuit board the tools are based on.


USB
MCU
Interface
Power
+9V
Accelerometer
FT232RL-BASICSSOP
Tiny85-20-SMT
ATMEGA328
ADXL330
MCP1703CB-3V3
TLE4207G
MINIUSB
TITLE: SPATA_fab01
Document Number:
REV:
Date: not saved!
Sheet: 1/1