Rapid Mold Prototyping: Creating Complex 3D Castables From 2D Cuts

by

Hanieh Shakeri

BSc., Simon Fraser University, 2018

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Rapid Mold Prototyping: Creating Complex 3D Castables From 2D Cuts

submitted by Hanieh Shakeri in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Examining Committee:

Karon MacLean, Computer Science
Supervisor

Robert Xiao, Computer Science
Supervisor

Alla Sheffer, Computer Science
Examining Committee Member

Abstract

Designers, makers, and artists prototype physical products by iteratively ideating, modeling, and realizing them in a fast, exploratory manner. A popular method of bringing 3D designs to life is through casting. Casting is the process of pouring a material into a mold, such that once the material sets, the target object is created. Currently, the process of turning a digital design into a tangible product can be a difficult one. One reason for this is that building the mold - for example by 3D printing it - can take hours, slowing down the prototyping process. This can be particularly true when prototyping molds for casting interactive (sensate and actuated) or geometrically complex (curvy) objects.

To this end, we developed two mold-making techniques intended to facilitate different, complementary needs for rapid prototyping. Both rely on computational support: as an intrinsic element of the first, and as an empowering add-on for the second.

The first technique we introduce is Silicone I/O, a making method based on Computer Numerical Control (CNC) that enables the molding of *sensate, actuated* silicone devices. This making method uses stacked laser-cut slices of wood bound together with molten wax in order to create cheap, accessible, one-time-use molds that are quick and easy to assemble. The Silicone I/O devices are pneumatically actuated using air channels created through lost-wax casting, and made sensate by mixing carbon fibre with silicone. We demonstrate the performance of these sensors with machine learning through a small user study (n=10). The second technique that we describe is FoldMold, which allows *curvy* molds to be rapidly built out of paper and wax. This approach is based on "unfolding" a 3D object, cutting the 2D layout, and using papercraft techniques to reassemble the mold.

Beyond the physical challenges of rapid mold-making, digitally designing mold patterns from 3D objects poses a bottleneck in the design process. We contribute the FoldMold Blender Add-on, a computational tool that turns 3D positives into CNC-ready papercraft mold patterns. This add-on unwraps the 3D model into flat designs, placing joinery along the edges, adding scores on the pattern where bends or curves occur, and providing patterns for external pieces that prevent the deformation of the paper mold under the weight of the casting material.

This thesis contributes within two different broad approaches to increasing increasing speed in mold prototyping. The first method is by creating flat, laser-cuttable mold patterns, significantly speeding up the actual mold creation process. The second method is by automating mold design, off-loading much of the tricky and tedious design work to a computer software that can help a maker design a mold very quickly.

Lay Summary

Design is an iterative process, often requiring multiple versions of an artifact to be tested and improved before completion. Quickly iterating on the design of castable objects (objects created by pouring a material into a mold) can be challenging due to the time-consuming nature of existing mold-creation techniques. This challenge is even more significant for designers and makers with limited access to the machinery, materials, and expertise required to make sophisticated molds. Current low-cost molding techniques lack the speed necessary for rapid iteration, and there are certain kinds of objects that remain unsupported by these techniques. In this thesis, we investigate how molding techniques can better support the rapid casting of curvy objects, objects that can sense touch, and objects that can move. We explore the automation of a mold-creation technique through software, considering the needs of a maker.

Preface

This research was done in collaboration with members of the SPIN Lab, especially Paul Bucci. Together, we supervised a team of undergraduate researchers, including Hannah Elbaggari, Anita Shah, Qianqian Feng, Bryan Lee, Eileen Ong, QiQi Li, and Liam Butcher. Though we collaborated heavily on all parts of our work, I was the intellectual lead on the Silicone I/O project (Chapter 3), especially in the designing, building, and evaluation of sensors. For the FoldMold project (Chapter 4), Paul led the design of the technique, which I contributed to by building and iterating on prototypes in hopes of validating and algorithmically modeling the process. Through this iterative process, we found many ways of changing and improving the usability, fidelity, and repeatability of the technique. I led the development of the FoldMold computational tool (Chapter 5), including making the primary design decisions and personally developing all aspects of the tool that are reported on in this thesis. I took collaborative input from Paul Bucci, Qianqian Feng, and Hannah Elbaggari on the design, requirements, and development of the computational tool throughout the process of creating it.

Table of Contents

Abstract . . . . **iii**

Lay Summary . . . . **iv**

Preface . . . . **v**

Table of Contents . . . . **vi**

List of Tables . . . . **ix**

List of Figures . . . . **x**

Acknowledgements . . . . **xii**

1 Introduction **1**
1.1 Background and Motivation 2
1.2 Objectives 3
1.3 Approach 3
1.4 Contributions and Overview 5

2 Background **7**
2.1 Rapid Shape Prototyping as Context for DIY Mold Making 7
2.1.1 Shape Fabrication Methods 7
2.1.2 3D Prototyping 8
2.2 Rapid Casting 9
2.2.1 Making Molds by Direct-Printing Object Negatives 9
2.2.2 Making Molds From 2D Materials 9
2.2.3 Making Molds From Object Positives 10
2.2.4 Materials Used in Rapid Casting 10
2.2.5 Computational Mold Creation 10
2.3 Interactive Castable Objects 11

2.3.1 Sensing . . . 11
2.3.2 Actuation . . . 13
2.4 Curvy Castable Objects . . . 14

3 Silicone I/O - Rapid Mold Prototyping with Laser-Cut Stacks . . . 16
3.1 Introduction . . . 16
3.2 Design Approach . . . 18
3.2.1 Design Exploration . . . 19
3.2.2 Parametric Design . . . 19
3.3 Molding Technique . . . 21
3.4 Making Actuators . . . 23
3.4.1 Process . . . 23
3.4.2 Demonstration . . . 24
3.5 Making Sensors . . . 25
3.5.1 Process . . . 25
3.5.2 Sensing . . . 26
3.5.3 Demonstration . . . 27
3.6 Gesture Recognition Study . . . 29
3.6.1 Method . . . 29
3.6.2 Results . . . 30
3.6.3 Conclusion . . . 31

4 Computational Support for Rapid Mold Prototyping . . . 32
4.1 Introduction . . . 32
4.2 Background: FoldMold Technique and Workflow . . . 33
4.2.1 Bending . . . 33
4.2.2 Joining . . . 34
4.2.3 Ribbing . . . 36
4.2.4 Process . . . 36
4.3 Computational Tool Requirements and Features . . . 38
4.3.1 FoldMold Feature Support . . . 38
4.3.2 Full Automation . . . 39
4.3.3 User Intervention . . . 39
4.3.4 System (Computational and Cutter) Compatibility . . . 40
4.4 Implementation . . . 41
4.4.1 Workflow . . . 41
4.4.2 Joinery . . . 42

4.4.3 Bending 43
4.4.4 Ribbing 44
4.4.5 UV Unwrapping 45
4.4.6 User Interface 46
4.5 Demonstration: Using the FoldMold Blender Add-on for a Kitchen Grip Mold 47
4.5.1 Creating the 3D Model 48
4.5.2 Scoring 48
4.5.3 Choosing Seams and Joinery 48
4.5.4 Ribbing 49
4.6 Pilot Study 49
4.6.1 Method 49
4.6.2 Results 50
4.7 Discussion on Computational Tool Performance 51
4.7.1 FoldMold Feature Support 52
4.7.2 Full Automation and User Intervention 52
4.7.3 System Computational and Cutter Compatibility 52
4.8 Conclusion 53

5 Discussion 54
5.1 Speed and Usability 54
5.1.1 Parametric Constraints Are Helpful in Rapid Exploration 55
5.1.2 Mold Design is Highly Automatable 55
5.2 Accessibility for a DIY Maker 56
5.2.1 Mold Material Choice Lowers the Barrier to Iteration 56
5.2.2 Leveraging Well-Known Software Lowers the Learning Curve 56
5.3 Production of High-Performance Prototypes 57
5.3.1 Wax Improves Surface Finish 57
5.3.2 Layered Molds Support Placement of Internal Components 57
5.3.3 Papercraft and Wax Enable Curvy Mold Creation 58

6 Conclusion and Future Work 59
6.1 Performance - Rapid Lost Wax Casting 60
6.2 Speed - Rapid 3D Design Exploration 60
6.3 Sensate, Actuated, *and* Curvy Castables 61
6.3.1 Making Methods 61
6.3.2 Computational Support For Designing Internal Components 62

Bibliography **63**

List of Tables

3.1 Our recipe for conductive silicone, mixing Ecoflex silicone with carbon fibre prepared with isoproprly. 26

4.1 Strengths and limitations of the FoldMold technique for scale, curvature, and fidelity. 38

List of Figures

3.1 Design Space: (A) Attainable geometries: our system can be used for geometries that are expressible in 2.5D and non-overhung 3D, and complex combinations therein; where we define 2.5D to be any arbitrary surface where all portions are describable as extrusions from a plane normal to the top of the mold pot. (B) Impossible geometries: molding methods prohibit overhang, leading to A's limits. 20
3.2 Constraint-based parametric design paradigms facilitate a structured approach to exploring a design space, where constraints are used to direct the design search,and parameterized relationships between geometric elements make variations on a design quick to produce. (A) Steps 1-7 show how to navigate the design space using developed parametric techniques. 21
3.3 The four kinds of silicone components that compose an interactive device. . 22
3.4 The Silicone I/O molding process. First, the designer discretizes the desired 3D model into flat sheets, which they parametrically design and laser-cut. They then reassemble the flat sheets in the correct order, sealing and smoothing the mold with molten wax. The casting material can then be poured in and cured. 22
3.5 Steps to creating an actuated sensate object. (1) Laser-cut beeswax air channels. (2)Stack the air channels with the other layers to elevate them to the right height. (3) Place sensing pads. (4) Cure silicone around everything. . 24
3.6 The final product of our actuator casting demonstration, showing the cast air channels being inflated. 25
3.7 Electronic circuits for pressure (left), touch and hover (right) sensing. Pressure sensing is done via piezoresistive sensing, where the change in resistance is measured, while touch and hover use capacitive sensing with two different strengths of resistors. For these tests, Rp=330, Rt=1M, Rh=16M. 27
3.9 The end product of our sensor casting demonstration - a fully soft SNES game controller prototype 29

3.10 Gestures used to train the silicone sensors. A Random Forest classifier was trained on capacitive data for these 10 gesture states. 30
3.11 Confusion matrix for gesture recognition for leave-one-out cross validation, percent accuracy averaged across runs (N=10); each run included 12 trials for each of 9 gestures for a total of 108 trials. 31

4.1 Molds can be bent in multiple ways, including simple scores (for 1D curves or sharp edges) or by using a kerfing pattern (for a specific two-directional curve). Joinery features allow for 3D reassembly and ribbing elements to reinforce the shape. 34
4.2 Three different kerfing styles. Each enables the material to bend in different ways (arrows). While the middle and right patterns have the same direction of bending, the middle pattern is more flexible. Patterns from [1]; laser-cut onto illustration board, a dense thick paper. 35
4.3 Join types. FoldMold uses four join types, and a pour-hole feature. Joins utilize pressure fitting for secure joining and to maintain alignment. 35
4.4 The FoldMold Blender Add-on. 41
4.5 Using the FoldMold Blender Add-on to create a mold. Steps 1-4 show digital steps involving the Add-on, and steps 5-6 include physical cutting, assembly, and casting. 42
4.6 An object created in the FoldMold Blender Add-on with ribbing along the x, y, and z axes for maximal support. These ribbing sheets are designed to slot into one another for easy assembly. 44
4.7 Cutting slots into ribbing slices along different axes allows them to interlock easily. 45
4.8 The user interface of the FoldMold Blender Add-on is based on three panels. The first panel (Mold Prep Panel) allows the user to specify a material, apply seams and joinery types, and create scores. The second panel (Ribbing Panel) supports the generation and conforming of ribbing elements, allowing the user to specify ribbing density. The third panel (Unfold) exports the mold into 2D patterns. 46
4.9 Our demonstration of the FoldMold Blender Add-on for creating a kitchen grip mold, showing the process from 3D model to a physical mold. 47
4.10 We presented this 3D model (a drinking glass) to the participant and asked them to create a mold design for casting it. 50

Acknowledgements

I thank my supervisors Dr. Karon MacLean and Dr. Robert Xiao for all of their insight, advice, and support over the course of my Master's degree. I am very grateful for their mentorship and for the many things they have taught me.

I am very thankful to Dr. Alla Sheffer for the time and effort she put into reading this work. I greatly appreciate the valuable new perspective she introduced.

I would also like to thank the members of the SPIN Lab and MUX for their ideas, feedback, and constant willingness to talk through problems I came across in this work. I especially thank Paul Bucci, Hannah Elbaggari, Anita Shah, and Qianqian Feng for their enthusiasm and never-ending stream of ideas, which made this project a joy to work on.

Finally, I thank my family for their love, support, and patience, and for always trying to help in whatever way they can.

Chapter 1

Introduction

We often think of designing tangible products as a slow, tedious process. From carving intricate stone statues to knitting a sweater, making physical objects takes time and effort. While the effort we put into this process is part of the joy and journey of creating memorable objects, there are often situations where speed is essential. Learning to quickly shape materials has been a struggle for thousands of years, and we have been designing solutions for just as long. Traditional sculpture techniques rely on various stone materials which are gradually carved to reveal the final intended form - a process that typically takes several hundred hours and requires extensive training, making it inaccessible to the average maker. An alternative solution is casting, which is the process of pouring a liquid into a mold and allowing it to solidify. Over time, these casting processes have become increasingly sophisticated. Early casting work most commonly involved molten metals such as copper and iron, cast by artisans to create farm equipment, religious items, and weaponry. Around 4000 BC, lost wax casting was invented as a way to create hollow cavities in these cast objects [2, 3].

With the rise of the Industrial Revolution in the 18th century, manufacturing processes became a point of great importance, with the use of machines for production becoming the basis of many factories. Over time, casting has become a widespread method of manufacturing products - especially plastics [4].

In the mid-19th century, illustrated self-help books began to portray at-home making as an enjoyable activity [5]. With the economic hardships of the early 20th century, making at home became a necessity. World War II brought even more importance to being self-sustaining, and by the 1950s the phrase “Do It Yourself” was a popular one [6]. Technology has been a momentous part of this do-it-yourself (DIY) movement, with the internet allowing makers to share their knowledge on platforms such as Instructables [7] and YouTube [8], and 3D printers becoming accessible enough for many to have in their homes.

Rapid prototyping is a design workflow that relies on quick ideation, iteration, and testing of products. It is a crucial part of many kinds of design work and fabrication. Advances in technology have allowed some making processes to be much faster and more accessible to the average DIY maker. However, to this day, making some kinds of tangible artifacts takes significant amounts of time and effort. While this is sometimes a worthwhile tradeoff for high precision and fidelity, it can also create a bottleneck for designers who need to rapidly prototype. Currently, the process of turning a digital design into a tangible product is one that can take hours, slowing down the prototyping process. This is felt most strongly when prototyping molds for casting complex objects.

Here, we use “complex” to mean that the tangible object is either interactive (sensate and actuated) or geometrically challenging to build (curvy). We focused on these two properties because objects exhibiting them present particular challenges in prototyping with current molding processes. Interactive objects require specialized techniques for incorporating internal components, and curvy objects are often discretized in order to be rapidly prototyped, resulting in a loss of curvature. Further, creating molds that have overhanging curves or concavities can be challenging with additive fabrication methods such as 3D printing.

1.1 Background and Motivation

The need for this research arose from our lab's prototyping endeavors; in our attempts to build soft, silicone-based robots that moved expressively in response to different human touch patterns, we were hindered by our inability to quickly generate and iterate on designs in a DIY manner.

Initially, we worked with a 3D printer to make molds for our castable robots, and we faced a print time of several hours for each piece. Taking into consideration multi-part molds, large models, and failed prints, this added up to a latency of hours to days. This cost meant that a significant design time had to go into each model, with the goal of ensuring that a minimal number of iterations had to be reprinted. Of course, good design relies on iteration, so prototyping with slow or minimal iteration was a roadblock.

Because we needed to very quickly explore many different design directions, 3D printing a mold for each iteration was impractical. Our rapid prototyping approach meant that each mold was only used once, as the designs changed with each iteration. This resulted in an expensive, high-waste process.

We needed to prototype molds for objects that were highly curved and/or interactive, which the previous techniques did not support. Highly curved objects were challenging to mold, as the techniques that were able to accomplish smooth, precise curvature were not also rapid. Interactive objects require internal components integrated within the bodies of the cast objects, such as pneumatic actuation channels or sensors. Accomplishing this level of complexity while remaining fast and accessible was a challenge. Of course, there is a trade-off - to create precise, polished objects, time and sophisticated equipment are required. Our goal, however, was not to achieve high precision and polish; rather, as prototypes, the cast objects should allow designers to sufficiently test their ideas.

1.2 Objectives

We aimed to develop mold-making workflows for complex castable objects that leverage the speed of two-dimensional (2D) Computer Numerical Control (CNC) cutters to make rapid, accessible prototyping possible for the DIY maker. These techniques needed to support:

1. *Speed and usability*: minimizing design time, making time, and amount of manual work and assembly, such that it enables rapid prototyping.

2. *Accessibility for a DIY maker*: using low-cost, low-waste materials that are easy to find and non-toxic, and equipment/resources that can be found in a typical workshop. Our goal is to create one-time-use molds, as we focus on a highly iterative rapid prototyping process where a single design is unlikely to be duplicated in exactly the same way.

3. *Production of high-performance prototypes*: capturing the intended object complexity, whether in enabling interactivity or smooth curvature and surface finish. We don't intend to compete with the precision that a sophisticated technology such as 3D printing - perfected over four decades [9] - can achieve; rather, we hope to support the rapid creation of objects that meet the designer's prototyping goals.

1.3 Approach

Our approach was iterative; with each iteration, new needs were exposed and informed our next effort. As outlined in 1.1, we started with the goal of making it easier to rapidly generate soft tangible objects of wide ranges of shape and material that can also support embedded sensate and actuating elements. This led to the development of the *Silicone I/O* technique (Chapter 3). "I/O" here refers to input/output, meaning that the cast objects can sense touch (input) and actuate (output). This technique is based on stackable laser-cut wooden parts sealed with wax and included a layered construction workflow to support both structural and embedded interactive elements, such as pneumatic actuation channels made with lost-wax casting, and capacitive touch sensing pads made of conductive silicone.

Silicone I/O's development involved exploration both in mold-making and in materials, e.g. encapsulating touch sensing within soft silicone molded objects, and supported the creation of re-usable molds.

In the process of developing Silicone I/O, we encountered limits on surface resolution and texture imposed by the stacked-sheet paradigm, as well as on scalability in size; it also precluded some interesting geometries, such as shapes without at least one flat side. However, when creating the Silicone I/O technique we were fascinated with the critical role of wax in our process. Wax (which can usually be purchased at most North American grocery stores) not only sealed a porous material such as wood, it also impacted the structure of the molds by holding the stacks together. Beyond that, it created a smooth surface finish, somewhat dulling out the stacked texture created by the molds.

With the aim of overcoming these limitations in surface resolution, we focused our next efforts on enabling the molding of curvy objects with a smooth surface finish. We explored a number of other approaches which included the inspiration of wax combined with folded and cut paper craft — origami and kirigami. From here, we realized that paper folding offers a completely different possibility for mold making, with benefits that complement those of a stackable-layer approach. Based on this insight, our group developed the FoldMold technique.

Like Silicone I/O, the FoldMold concept constructs molds for 3D geometries out of 2D media. However, rather than building it up through stacked layers, a FoldMold conceptually "wraps" an object positive; and critically, use of bendable media such as paper and cardboard allows for accommodation of curvature which is smooth rather than discretized. A major difference between the two techniques is that Silicone I/O requires the designer to directly design the mold (the object negative) while FoldMold allows the designer to focus on the object positive.

While conceptually simple, a FoldMold can be mathematically complex. Except for simple geometries (which other techniques can accommodate), generating the paper mold from the 3D positive model is not a human-friendly task. From the start, the importance of computational support was obvious. One reason for this is that a designer typically focuses on the design of the target object (the object positive) rather than the design of its mold (the object negative). Converting between the two in such a way that allows the mold to be laser-cut requires complex spatial thinking, especially as objects become more complicated. Further, with our FoldMold paper-based technique, support pieces that ensure structural integrity must also be designed, and this requires careful planning and engineering. Calculating the placement of support pieces and joinery is challenging and time-consuming even for experts, but amenable to computational automation given some fairly general heuristics.

We thus developed the FoldMold computational tool, which allows the designer to focus on the created object positive, and automates generation of laser-cuttable mold patterns and supports, while supporting designer intervention at every stage. We started with FoldMold for this enhancement rather than Silicone I/O because the gains were most obvious: FoldMold has more components and is generally more tedious and difficult for makers to design, but those components are theoretically possible to compute mathematically. Additionally, computational tools for stack-based molds have previously been developed [10], while computational support for this kind of paper-craft based molding has been relatively unexplored. There is a significant challenge in creating the interface that allows a maker to provide the most valuable input and customization, which we hoped to investigate.

Pursuing these two similar yet different techniques allowed us to isolate the challenges associated with interactivity and curvature, tackling each separately with the goal of characterizing their solutions. The effort of developing computational support for FoldMold gave insight into the requirements and challenges of automating rapid mold-creation in general, which can help to guide computational support for many different molding techniques.

1.4 Contributions and Overview

In this thesis, we specifically contribute the following. While development of both the Silicone I/O and FoldMold techniques were group efforts, just the first is claimed as a contribution here, while FoldMold will be summarized in this thesis for context.

1. The Silicone I/O making method that enables the molding of sensate, pneumatically actuated silicone devices.
    - Detailed demonstrations of the process of making sensate and actuated silicone devices.
    - A user study evaluating the sensing (machine learning) performance of the end products.
2. The FoldMold Blender Add-on, a computational tool to support the process of making highly curved castable objects with wax-stiffened paper molds.
    - Requirements for rapid mold-making software.
    - A demonstration of how our Blender Add-on can be used to create a cast object.

- An informal evaluation of how well the FoldMold Blender Add-on meets our requirements.

We start with background on the concepts and techniques that led to this work (Chapter 2). In Chapter 3, we present the first technique - Silicone I/O - including demonstrations and an evaluation of the sensing capabilities of the cast devices. Next, in Chapter 4 we cover the FoldMold making technique as background for the requirements of the FoldMold computational tool, which we describe in detail. We demonstrate the use of the computational tool and reflect on how well it meets the requirements. Then we close with discussion and conclusions jointly about all of these efforts, in Chapters 5 and 6.

Chapter 2

Background

In this section, we describe the various fields we drew inspiration from and refer to throughout this work, contextualizing our research at their intersection. We provide an overview of rapid prototyping systems and how they relate to casting, tools and techniques for computational mold creation, a brief overview of how interactive castable objects have been made in previous work, and information on how curvy castables can be made using papercraft and woodworking techniques.

2.1 Rapid Shape Prototyping as Context for DIY Mold Making

Rapid prototyping is a framework that relies on quick ideation, iteration, and testing of designs. Systems that support rapid prototyping often introduce design tools to aid in the reduction of design time, or contribute new methods of using computer-controlled machines to reduce the physical production time.

2.1.1 Shape Fabrication Methods

In the 1980s, *stereolithography* was introduced. This fabrication technique involved layers of liquid photosensitive polymers being extruded and cured by a irradiation light source, building up to a 3D shape [11]. Over the past few decades, this technology has grown into modern-day 3D printing. Most commonly, 3D printers are used with polymer filaments that are heated, extruded, and cooled. Certain 3D printers can achieve high resolution and precision, at the expense of speed, cost, and available material options [12]. A limitation of 3D printing for shape fabrication is that due to its layer-by-layer process, overhanging geometries are challenging or impossible to print. Some overhangs can be created by printing support material that scaffolds the overhangs, creating the potential for deformation and using excess material [13].

Another approach to shape fabrication is through CNC (Computer Numerical Control) machining. In the beginning, CNC machines would cut materials based on points defined on punch cards [14]. Today, most CNC machines are computationally controlled by G-code [15]. CNC machining technologies include drills and laser-cutters which can create 2D artifacts, and lathes and milling machines which can create 3D artifacts, among others. Of these laser cutting stands out as being suited to rapid shape fabrication due to its relative inexpensiveness, speed, and precision [16].

2.1.2 3D Prototyping

Additive and Subtractive Methods

Methods for rapid shape prototyping are generally either additive or subtractive.

Additive methods gradually build a 3D object up layer-by-layer. Over the past few years, 3D printers have become a popular additive tool in the rapid prototyping community [9], due to their accessibility, breadth of material choices, quality, and speed relative to previous rapid prototyping techniques such as photosculpture [17] and directed light fabrication [18].

Subtractive methods create 3D objects by cutting material away until the desired shape is achieved. A popular subtractive method is laser-cutting 2D object sections which will later be combined into a 3D whole, which similarly to 3D printing, has gained traction due to its speed, accessibility, precision, and safety relative to techniques such as plasma-cutting [19] or water jets [20].

DIY Mold Reusability

Molds can be disposable [10] or reusable [21], depending on the casting goals. Reusable molds are generally used to cast the exact same object a number of times; they become increasingly valuable for repetition of an established design; usually this is later in a development cycle. Mold re-use has the benefit of reducing waste as well as the costs entailed in making the molds. These molds are typically made of durable materials such as silicone that do not deform over time, and are often very high-fidelity. Creating a reusable mold is usually a slow process, including extensive design work (to reduce the need for iteration) and slow but precise fabrication methods. However, if the mold is meant to be used multiple times, the slow one-time process of making the mold is worthwhile.

Often, the casting goal is not object replication. In contexts where rapid iteration is needed, cast objects are used as tests of a design that are meant to be changed, meaning that the same design is unlikely to be cast twice. In these cases, creating a mold from a durable material is not only unnecessarily expensive, it can actually be detrimental if the material is not easily disposed of and biodegradable/non-toxic. In rapid mold iteration use cases, the mold must be fast to make, cheap [22], and easy to destroy [23].

2.2 Rapid Casting

2.2.1 Making Molds by Direct-Printing Object Negatives

One way to create molds for casting in a DIY manner is through 3D printing. This process involves designing a 3D model of the object negative, leaving space for the casting material to be poured in. Two geometric challenges exist with this approach. The first challenge is that overhangs can be challenging to print, requiring excess scaffolding material that can fill up the mold [13]. The second challenge is that extracting a complex cast object from a 3D printed mold requires the mold to have several separate parts that can be opened up; as the number of segments of the target object increase, this becomes infeasible to keep sealed. Further, due to the layered surface texture of 3D printed molds, casting materials such as silicone can seep into crevices and become hard to remove - especially for very fine geometries that can break under the pressure of pulling away from the mold.

Relative to the speed of the rapid iteration process, 3D printing can be slow, becoming an obstacle; a single project may take hours to days to print. Some techniques, e.g WirePrint, have aimed to improve printing time by directly modifying the digital 3D model design to reflect a mesh version of the desired solid model [24]. However, these techniques cause a decrease in the amount of material used, resulting in geometries that cannot be used as molds (because they have holes).

2.2.2 Making Molds From 2D Materials

Thin 2D shapes can be produced relatively quickly, e.g. with planar CNC devices such as laser cutters. Thus, some have sought speed by cutting 2D patterns that will be folded or assembled into 3D objects [10, 25, 26, 27].

FlatFitFab and Field-Aligned Mesh Joinery allow the user to create 2D laser cut pieces that, when aligned and assembled, form 3D approximations of the object [25, 26] that can be quickly assembled, while other methods (Joinery, SpringFit) focus on laser cutting assembly techniques from 2D cutouts [28, 29]. Many of the above methods combine a quicker prototyping method such as laser cutting for larger volumes, with smaller 3D printed parts for areas of the model that need finer details [10, 30].

In this work, we draw inspiration from these previous laser-cutting approaches, and combine them in order to apply laser-cutting to a paper-craft based molding context. We also explore the computational automation of generating laser-cuttable patterns.

Other efforts have focused on prototyping the mold-making system itself. For example, StackMold is a system for casting multi-material parts that uses stacked laser-cut pieces of wood to create molds [10]. StackMold incorporates lost-wax cast parts to create internal structures such as overhangs or cavities. Metamolds [21] is a system that uses a 3D printed mold to produce a second silicone mold, which is then used to cast objects.

While previous work has made significant progress in exploring rapid molding, creating interactive or curvy objects with rapidly-created molds is still not completely possible. StackMold begins to look at incorporating ready-made sensing components in cast objects [10]. We expand on this work to include the molding of the sensing components themselves, and study their sensing capabilities, as well as integrating actuation abilities. We also investigate methods for creating a smooth, non-stacked surface finish in our molds.

2.2.3 Making Molds From Object Positives

Molds that are created for casting object duplicates are often created from the target object positive. This can be done by placing the object positive into a molding material such as silicone, allowing the molding material to cure around the object, preserving the imprint of the positive [31]. A particular utility of casting in rapid prototyping is access to a wider range of materials like silicone or plaster for creating an exact copy of an target object positive.

2.2.4 Materials Used in Rapid Casting

The choice of casting material strongly impacts the appropriate molding method. For example, for high-temperature casting (e.g. metal, glass), it is important to choose a mold material with a higher melting temperature, such that the mold does not deform once cast.

For objects cast from certain materials (such as plastic), directly printing the object positive is possible and potentially worthwhile. For materials like silicone or plaster, this cannot be easily accomplished. For example, when rapidly prototyping a soft robot, it may be important for the prototype to reflect the desired softness and malleability – simply prototyping the object from another material would not accomplish this. While research has been done on direct silicone printing [32], this is still experimental and not accessible to a DIY maker.

2.2.5 Computational Mold Creation

Computationally-supported design can make complex geometric tasks more accessible to designers [33, 34, 35, 36, 37, 38, 39]. An example of this is LASEC, a system presented by

Groeger et al. that allows for simplified production of stretchable circuits through the use of a design software and laser cutter [40]. Though many of these design tools include software that automates part of the process, others are computationally-supported frameworks or design approaches [41]. Our work draws from and extends this area in our parametric design approach.

When designing tangible objects, designers will often begin by creating digital 3D models of the intended objects using CAD software. To mold the modeled object, the complement of the object (the object negative) must then be computationally generated, and converted to physical patterns that can be re-assembled into a mold. This is where mold-design software can aid in making the process faster and better suited to a rapid prototyping workflow. Stackmold slices the object negative into laser-cuttable slices [10]. Metamolds helps users optimize silicone molds [21].

These previous mold-design tools inform our approach to creating an interface for mold creation. Our computational approach is fundamentally different because our molding technique relies on paper-craft based techniques. Essentially, in our computational tool the 3D model of the object positive is "unfolded" and elements are added to reassemble the 2D patterns into a structurally sound mold. Similar "unfolding" algorithms have been used in existing tools, such as the Unwrap function in Fusion 360 [42] - though not in a mold-making context.

2.3 Interactive Castable Objects

Prototyping tangible objects that can sense and move is particularly challenging, requiring additional hardware design, programmatic implementation, and careful engineering of how components can be fully embedded within the bodies of castable objects. In an attempt to overcome this challenge, we directly integrate the sensing and actuation structures within the cast material itself. In this section, we describe previous research on casting sensate and actuated objects, which we draw from in building our interactive tangibles.

2.3.1 Sensing

Sensing Materials

Existing work in this area includes sensate fabric [43, 44, 45], hydrogel [46, 47], and indium tin oxide films [48]. However, while indium tin oxide is bendable, fabric is flexible, and hydrogel is stretchable, these are all varieties of sensate films that must be positioned on top of some object in order to achieve sensing. This is not only a difficult mapping to make physically, but achieving accurate sensing with these films overtop of complex geometries may also be challenging.

We overcome this by incorporating the sensate material into the actual structure of highly deformable objects. We accomplish this through the use of silicone, which is extremely soft, stretchable, and deformable. Others have explored silicone-based sensors [49, 50], but this generally involves skin-like silicone films rather than the use of silicone as a structural element.

To make silicone sensate, it must be combined with an electrically conductive material, as pure silicone is a dielectric. Previous work has explored inserting sheets of indium tin oxide-coated conductive flexible polyethylene terephthalate between layers of silicone [51], but this limits the flexibility and stretchability of the overall material. Rather than embedding conductive materials inside silicone, it is also possible to dope silicone with materials that make the silicone itself conductive, such as carbon black [52] or carbon fibre [53]. Here, we used carbon fibre because it is much cheaper and more accessible than sufficiently conductive carbon black.

Sensing Techniques

Silicone doped with carbon fibre is both capacitive and piezoresistive. When a material with either of these properties is deformed, the current flowing between strategically located electrodes changes in a deterministic albeit complex way. These changes can be sensed and recognized with machine learning in order to identify gestures. We summarize here how others have implemented these sensing approaches, which we have adapted for use in our silicone sensors.

In mutual capacitance sensing, touch [54, 55, 56] and pressure [51] contact cause capacitance to accumulate between parallel charge and ground plates separated by a dielectric, which can be measured. Alternatively, the capacitive sensor can follow a self-capacitive principle, which involves measuring the capacitance between the charge plate and some grounding object, such as a human finger. This technique has been shown to produce larger changes in capacitance at greater distances compared to mutual capacitance sensing [44], making it more suited to measuring proximity and light touch. While mutual capacitance sensing has better touch accuracy and can afford multi-touch sensing, self-capacitive sensors are simpler to design [57], which is an important consideration for our accessible approach. For this reason, we focus on self-capacitive sensing.

Piezoresistive sensing uses conductive material that changes resistance upon being compressed where resistance across a material is measured to determine pressure [58]. Carbon fibre-doped silicone is piezoresistive: as the silicone deforms, conductance changes across the network-like structure of the fibres. We draw from work on the impact of carbon fibre density on silicone conductivity [53], determining a silicone brew that is responsive to pressure, large changes in resistance when the material is deformed.

Signal Processing

Machine learning can be used for recognition of spatially or temporally complex interactions such as touch gestures. ML features calculated from resistive [59, 60, 61] and capacitive [62, 63, 64] signals are commonly used for gesture models: Cang et al. used a piezoresistive fabric sensor to train a gesture recognition system using a Random Forest Classifier [43]; Touche uses swept frequency capacitive sensing to train a Support Vector Machine to detect gestures [65]. Our work draws from this research, as we implement and evaluate a Random Forest Classifier for gesture detection with capacitive signals.

2.3.2 Actuation

Actuation design for soft robotics is an incredibly expansive field. The most common approaches to actuation rely on highly engineered mechanical systems akin to skeletons within the robots [66, 67]. Muscular structures can be simulated through the use of shape-memory alloys [68, 69] or tension cables [70, 71]. For example, one work explores the use of shape memory alloys to actuate a soft earthworm robot [72]. Other approaches to actuating soft robots include electrothermal [73], pyrotechnical [74] and magnetic actuation [75].

For robots made with soft materials, fluidic actuation is a popular approach, as soft materials can often inflate, creating movement. Hydraulic actuation, for example, uses water to create large forces [76, 77]. Due to silicone's deformability and potential for biomimicry, it is excellent for this purpose, and has generated interest in the field of soft robotics. Pneumatic actuation [78, 79, 80, 81, 82, 83] is the most thoroughly explored, and is a good match to silicone because it is lightweight and can create natural-looking movements.

Lost wax casting can be very useful for creating channels for fluidic actuation. Katzschmann et al. use lost wax casting for creating hydraulically-powered fish [77]. Incorporating lost-wax casting into a rapid prototyping workflow poses an interesting challenge that has not been explored in previous literature. The wax structures are usually molded themselves, requiring a tedious process that takes a significant amount of time and design work [84].

We devised a rapid lost-wax-casting technique that uses laser-cut wax sheets incorporated into molds in order to create air channels within silicone structures to enable selective and targeted pneumatic actuation.

2.4 Curvy Castable Objects

Rapidly prototyping curvy objects can be difficult. Using additive prototyping methods, such as 3D printing, curves are broken down into small but discrete layers and gradually built up. To capture a smooth curve, many layers must be printed - which is a time consuming process. Subtractive prototyping methods are also poorly suited to creating curvy objects. Most CNC machines (with the exception of lathes) do not support cutting at an angle, creating a noticeable "staircase" effect on object edges when cut layers are stacked. Instead, we try to leverage the classical techniques of origami, kirigami, joinery, and kerfing to allow flat-cut sheets to curve and be assembled into 3D objects.

Origami involves folding a single sheet of paper repeatedly into a 3-D shape, sometimes with astonishing complexity, fidelity, and number of folds. Like origami, Kirigami is based on folding, but allows cutting to simplify the folding and give accessibility to a broader range of shapes. These folding and cutting techniques reveal fundamental properties of achieving three-dimensionality from a 2D sheet.

Creating Origami/Kirigami Patterns for 3D Objects

Others have sought ways to discretize 3D object surfaces to creating foldable patterns and control deformation. Castle et al. developed a set of transferable rules for folding, cutting, and joining rigid lattice materials [85].

Similar work on kirigami 3D structures showed that specific cuts to flat material can be buckled out of plane by a controlled tension on their connected ligaments [86]. Research and mathematical work on these papercraft techniques have informed cut/fold systems.

Applying this to prototyping, LaserOrigami uses a laser cutter to make cuts on a 2D sheet that then get melted into particular bends, deforming the 2D surface to make a precise 3D object [87].

Kerfing

A wood bending technique that involves creating a pattern of many small cuts on a workpiece, kerfing is used to impart flexibility to rigid material [88]. The width, shape and patterning of these cuts leads to controlled bends and curves.

There are many techniques and designs to achieve specific curves: a pattern of short through-cuts renders a different curvature than cutting partway through. Consideration must be given to cuts' depth, size and proximity. Rigid materials can be cut to augment bending performance [89, 90, 91, 92]; and achievements include manufacturing complex double curved surfaces [89, 92], stretchiness [40], and conformation to preexisting curves for measurements [93]. Kerfed and rigid sections can be combined to create a strong, continuous, and curvy piece of material to structurally reinforce a design [30]. Kerfing has informed our approach to achieving smooth, structurally sound shapes at a high resolution.

Joinery

In joinery, two pieces of wood are cut such that they fit firmly into one another, mechanically strengthening the material bond, which can be reinforced with glue, screws or dowels. It is common in modeling, construction and fine woodworking. There are many joint types, with advantages related to material and needed strength.

One project (Joinery) has developed a parametric joinery design tool specifically for laser cutting to create 3D shapes [29]. Joinery has been used in rapid prototyping literature: Cignoni et al. creates a meshed, interlocked structure approximation of a positive shape to replicate a 3D solid object [26]. Conversely, SpringFit shows how mechanical joins can lock components of an object firmly in place and minimize assembly pieces [28].

We draw on all of these techniques by incorporating them into our paper-craft based molding technique. Our overall method of mold construction is inspired by the cut-and-fold approaches of kirigami and origami, which we combine with joinery to create patterns that can be rapidly assembled and sealed for casting. We use kerfing techniques to enhance the natural flexibility of paper to create molds that are controllably curvy.

Chapter 3

Silicone I/O - Rapid Mold Prototyping with Laser-Cut Stacks

3.1 Introduction

Design exploration often requires castable input/output (I/O) devices to be rapidly built, tested, and iterated upon. Creating arbitrarily-shaped molds in a standard workshop is not a well-supported task; although 3D printers are able to easily print simple molds (within the constraints of no overhangs or support material), even small molds can take many hours to print. Adding other components such as sensors and actuation parts makes for even more complicated 3D printed models, necessitating multi-part designs. For highly engineered final products, the effort to produce complicated molds might be justified, but for rapid prototyping, 3D printing is too slow.

To speed up the prototyping process and simplify the design for components with internal negative space, we developed a system for making molds out of laser-cut wood and wax. Molds are built up out of laser-cut "wafers" and joined with melted wax; wafers can be made out of wood or wax. Wafers are essentially discretizations (slices) of the mold. Their thickness is determined by the thickness of the material from which they are cut; thicker materials lower the object resolution but improve cutting speed.

3D printing is conceptually similar to this approach, with the major difference that the extruded layers of a 3D printed object are *much* thinner than the wafers we use here. For this reason, 3D printing can achieve higher fidelity, but with much lower speed. As an alternative to 3D printing molds for casting silicone, research has explored direct silicone 3D printing [32], but this is still experimental and not accessible to a DIY maker.

Extracting a complex cast object from a 3D printed mold requires the mold to have several separate parts that can be opened up; as the number of segments of the target object increase, this becomes infeasible to keep sealed. Further, due to the layered surface texture of 3D printed molds, casting materials such as silicone can seep into crevices and become hard to remove - especially for very fine geometries that can break under the pressure of pulling away from the mold. In this Silicone I/O making method, these problems are overcome by using wax to seal and smooth the interior of the wooden molds, allowing the cast object to be easily removed. Due to the layered structure of our molds, they can be pulled apart to release the cast object.

In tackling this problem, we decided to focus on silicone as our casting material due to the unique interaction opportunities it presents. Silicone is flexible, stretchable, waterproof, heat-resistant, dielectric (non-electrically-conductive), food safe, and easy to clean (e.g., in a dishwasher), and thus perfect for products that interact with the human body. Simultaneously, the robotics community has learned how to actuate silicone with air [78, 79, 80, 81, 82, 83], explosives [74], and magnets [75], producing delightfully biomimetic soft robots with "infinite" degrees of freedom that are pleasant to interact with via touch. The potential of silicone for naturalistic actuation and its capacity for molding into arbitrary shapes opens up a world of expressive actuated devices.

Although silicone is typically non-conductive, when doped with conductive particles it reflects the electrical properties of the added material. Adding carbon fibre to silicone in varied amounts creates silicones of corresponding resistance. Depending on the mix, it is also possible to create silicones that are piezoresistive [94], i.e., change resistance when they are compressed/stretched. Novel sensors can be created by adjusting the material's conductivity, resistivity, and piezoresistance for sensing a variety of expressively interactive dimensions, including location, pressure, and hover. Sensing and actuation with silicone have both been explored extensively in previous research, but not in a rapid prototyping context.

The layered structure of Silicone I/O molds directly supports the inclusion of internal sensing and actuation components by allowing layers to be successively poured, controlling the positioning of these components within the cast object. For objects without concavities, a similar layered-pouring approach could be taken with a 3D printed mold. However, because a 3D printed mold is fully constructed before pouring, places in the mold that are narrower will prevent layered positioning of internal components, and possibly even prevent them from being inserted into the mold at all.

In this chapter, we explore the making of silicone input/output devices that directly incorporate conductive silicones and actuation channels in one-time-use wood and wax molds. We target cheap, rapid prototyping of these devices to enable non-engineer designers and makers to create expressive and geometrically complex soft interfaces and robots.

Our objective with this work was to create rapid making and sensing techniques for soft silicone touch sensors with complex, expressive geometries. We targeted materials and equipment typically found in a small maker space and affordable for the average maker. All materials used per sensor design can be acquired for well under $100; the most expensive item used was a laser cutter, often available for $0.10/min. The skills required are within most novice makers' grasp: the making techniques require only a basic understanding of CAD programs and laser cutting, whereas the sensing techniques require basic Arduino circuit building and programming. Targeting the eventual streamlining of the techniques we present here, we took a structured computational approach to design, where we used compositional parametric design paradigms common to tools like Solidworks and AutoDesk Inventor. By building our designs out of simple geometric primitives and transformation rules, we produced "recipes" for creating the sensors within a large but constrained 2.5D to complex 3D design space. By challenging ourselves to broadly explore this design space within the constraints of rapid, low-cost prototyping, we could begin to characterize the interaction between device geometries, sensing abilities, and making techniques, reinforced by empirical testing of the gesture detection system.

3.2 Design Approach

To design a technique that eases the mold-making challenges present in silicone device casting, we had to first identify and understand the geometries, structures, and constraints of these objects. In this section, we describe how we explored the design space of soft, sensate silicone structures and identified making-derived constraints and efficiencies, through a process of prototyping and analysis, and arrived at a structured approach to CAD and fabrication through constraint-based parametric design for mold-making techniques.

We iteratively cycled between two investigative approaches: exploratory, in which we sketched a wide range of expressive geometries, and structured, wherein we analyzed the geometry of our sketches and attempted to describe them in terms of a small set of geometric primitives. As we modeled our sketches in CAD software, produced increasingly complex shapes in molds, and developed a computational approach to realizing our sketches in silicone, we attempted a number of ways of structuring the space but found they were not amenable to rapid prototyping. For example, we attempted to use generative computational approaches such as shape grammars: while successful in reproducing the shapes, they offered little in terms of easing mold design. These experiments and the resulting insights eventually led to a constraint-based parametric design paradigm, through which we were able to define design and making methods that reduce the cognitive load, skill and time necessary for producing new devices while retaining a satisfyingly rich set of options.

3.2.1 Design Exploration

Our exploratory investigations took the shape of unconstrained form studies which quickly taught us the limitations of the silicone molding process, whereas our systematic approach focused on defining, modeling and making geometric primitives such as 2D circles and squares and 3D polygons, and laid the groundwork for our eventual parametric design method. We began with a series of ambitious hand-built clay maquettes varied in geometry, complexity and texture. From these, we analyzed the geometries, attempting to create a structured, algorithmic approach to reproducing our maquettes in CAD software and ultimately silicone. Our initial attempts at structuring were based in creating rules that expressed complex shapes in terms of iterated 3D primitives such as prisms, spheres and other polygons. Although we could create CAD models of these geometries, producing molds was more complex and time consuming. When we began mixing conductive/non-conductive silicone and insert air channels for actuation, we found that mold design became infeasible without expertise in engineering.

3.2.2 Parametric Design

We decided to work in a constrained design space that could be accessible to non-engineer makers and designers and moved to a parametric design paradigm through trial and error. Iterating between an exploratory and structured approach exposed modelling problems, forcing us to develop best practices for mold creation within our constrained space, taking inspiration from silicone mold tutorials. However, most do-it-yourself tutorials focus on silicone as the molding media, i.e. the negative, not the material of the final intended positive. For example, injection molding is often used for creating silicone positives, but the complexity of adding conductive/non-conductive parts and air chambers made a direct injection molding approach infeasible. As such, we developed our own approaches using, at first, two-part 3D printed molds. Attempting more complex internal and external geometry exposed making challenges. Similar to the constraints of injection molding, overhangs and internal negative spaces require complex, design-intensive molds to ensure that parts can be removed without being destroyed.

To focus on rapid prototyping, we targeted a constrained design space that still enables a wide variety of possible shapes. Since we are working with real objects in a 3D world, there are no true 2D shapes (see Figure 3.1). Borrowing terminology from game design, we consider a 2.5D shape as one that can be cut out of a plane as with a cookie cutter. In our

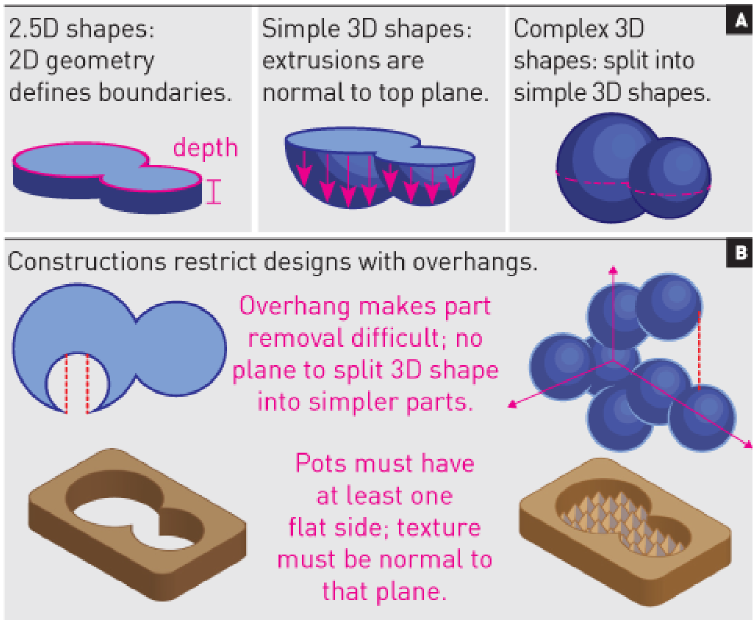

Figure 3.1: Design Space: (A) Attainable geometries: our system can be used for geometries that are expressible in 2.5D and non-overhung 3D, and complex combinations therein; where we define 2.5D to be any arbitrary surface where all portions are describable as extrusions from a plane normal to the top of the mold pot. (B) Impossible geometries: molding methods prohibit overhang, leading to A's limits.

system, a simple 3D shape can be constructed from a 2D shape with perpendicular extrusions from one side, and complex 3D shapes are decomposable into two simple 3D shapes that are "stitched" together. All of the shapes we consider can be made from iterations of molds consisting of a single-sided 'pot'; that is, all molds have at least one flat plane or can be "stitched" together along flat planes.

We found that using a constraint-based, parametric design approach (1) reduces aimless wandering, by providing high-level, 'tested' constraints that can be followed and parameters to be manipulated; and (2) facilitates rapid iteration through similar families of prototypes. We articulated a roadmap for quickly designing and constructing prototypes, illustrated in Figure 3.2. This parametric design system includes defining parameters across multiple stages. First, the designer must choose a bounding shape, such as a circle or a rectangle, in which the device shape will be created (1). Next, primitive shapes must be placed within the bounding shape, and relationships between the shapes should be defined (such as tangency, distance, parallelism, etc) (2). These primitives comprise the "bones" of the design; the final outer shape of the design may not directly adhere to these geometries, but is instead driven by them. The parameters can then be manipulated, and primitives can be added, repeated, and moved until they are in the desired configuration (3). The primitives must

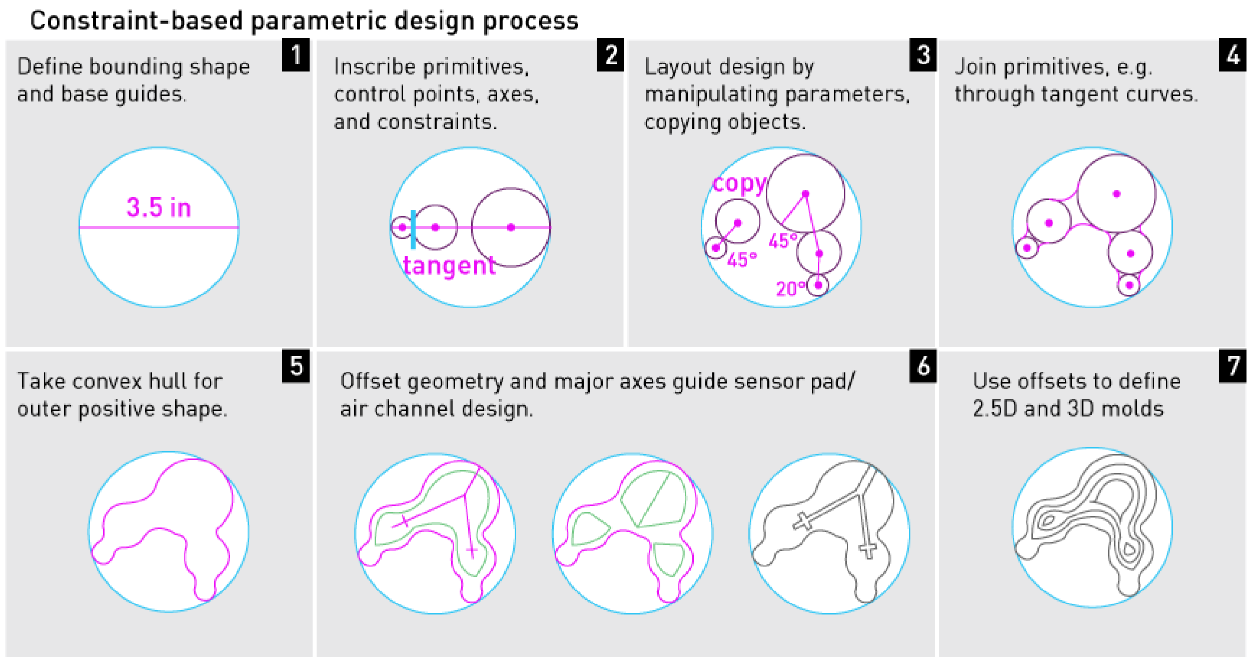

Figure 3.2: Constraint-based parametric design paradigms facilitate a structured approach to exploring a design space, where constraints are used to direct the design search,and parameterized relationships between geometric elements make variations on a design quick to produce. (A) Steps 1-7 show how to navigate the design space using developed parametric techniques.

then be connected, for example through lines or curves (4), and the resulting outline can be created (5). This geometry is now the basis of the exterior shape of the device. To generate internal geometries, an offset of the outline can guide sensor pad placement, and major axes can guide air channel design (6). Further offsets can then be used to design the "stacks" that create 2.5D to 3D shapes (7). This greatly reduces the time spent deciding on where to place pads and channels.

As documented in Figure 3.3, we built our devices from four kinds of silicone components: conductive pads that have carbon fibre mixed in, non-conductive positive parts that comprise the outer structure, non-conductive parts with negative space to enclose conductive pads, and non-conductive parts with negative space to use as air channels for actuation. The parametric system we used makes the design of each part follow easily from high-level constraints defined at the time the structure's exterior envelope is designed.

3.3 Molding Technique

The process of molding a silicone device is shown in Figure 3.4. First, the desired object

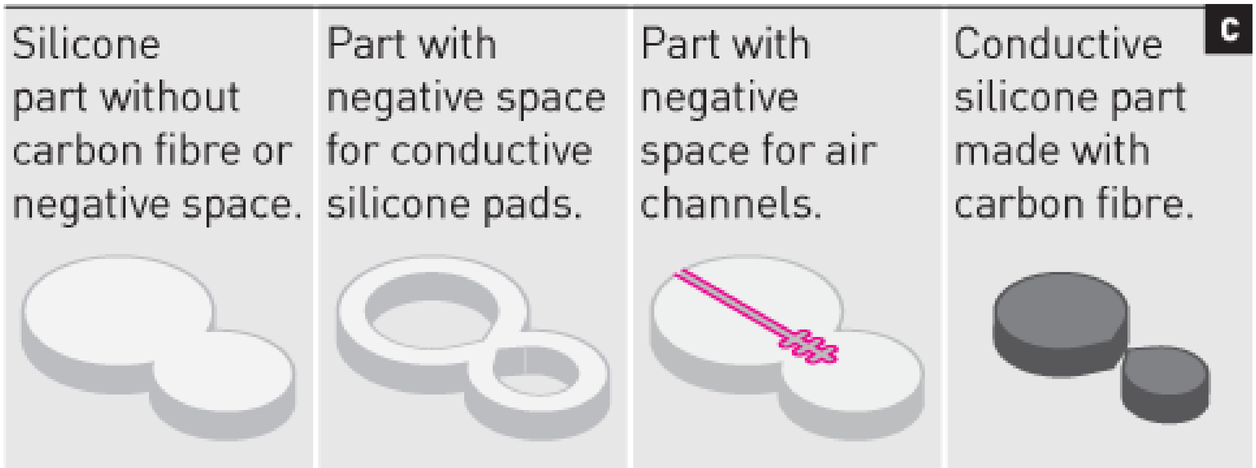

Figure 3.3: The four kinds of silicone components that compose an interactive device.

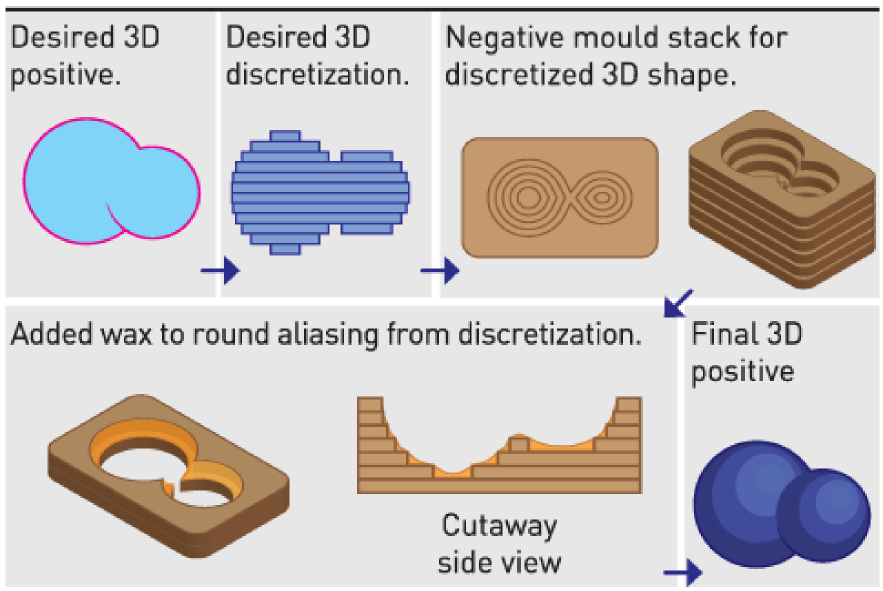

Figure 3.4: The Silicone I/O molding process. First, the designer discretizes the desired 3D model into flat sheets, which they parametrically design and laser-cut. They then reassemble the flat sheets in the correct order, sealing and smoothing the mold with molten wax. The casting material can then be poured in and cured.

must be parametrically designed as “slices” of the mold (the object negative), including slices for air channels and sensing components, using the process shown in Figure 3.2. Next, these slices must be laser-cut. While most slices are cut out of wood, air channels and empty pockets for sensor components must be laser-cut out of wax sheets. These slices are then stacked together in the correct order, such that the internal components are positioned at the correct height. The maker may choose to clamp these stacked layers together to minimize any gaps through which the silicone could later leak. As the layers are stacked together, molten wax can be poured into the mold, soaking into the fibers of the wood and filling in any gaps (to completely seal the mold) as well as smoothing away the “aliasing” effect of the stacked layers. This wax may result in the softening of small or sharp details; to counter this, a hot, narrow-tipped tool (such as a soldering iron) can be used to melt excess wax away in specific areas. A major benefit of this wax is that it allows the silicone object to later be removed from the mold with ease, as it creates a fully smooth surface as opposed to the normally fibrous surface of wood molds.

The mold is then ready to pour. The mixing of different materials and geometries makes it such that some designs can be poured in one step whereas others require multiple pours. An object cast purely out of silicone, or silicone with air channels, may be poured in one step. However, an object with sensing components must be poured to the appropriate height and cured, allowing the components to then be suspended atop the cured silicone at the correct height and cured onto the object. Another layer can then be poured to fully embed the components within the silicone object. Once everything is fully cured, the object can be removed from the mold. Occasionally, bits of wax from the mold may stick to the positive; these are easy to remove manually or by using heat. At this stage, objects that were cast with internal wax components must have their wax melted out, leaving empty channels within the silicone structure.

3.4 Making Actuators

3.4.1 Process

Our approach to actuation was pneumatic: we explored the possibility of casting air channels within a silicone object, which could be in/deflated to achieve movement. A minimal movement possible with this approach is an expandable bubble, which can be actuated with considerable expressivity [95]. In the future, this basic premise can be extended into more complex structures entailing valves and hinged limbs.

To achieve pneumatic actuation, we needed a method for creating air channels within the poured silicone. We attempted two methods: (1) with lost wax casting, a technique where a positive version of the air channel is made out of wax, silicone is cured around it, then the wax is melted away; (2) by splitting the air channel along the bottom plane, curing the rest of the silicone positive, then removing the piece and adhering an airtight seal over top of the negative air channel. After attempting to hand-build some wax channels, we experimented with laser cutting wax to increase the precision of our channels. After some varied success with casting different thicknesses of paraffin wax sheets and attempts at tuning laser cutter settings to accommodate melted wax, we moved to flat beeswax sheets. These had the advantage of being both thin and strong, reducing the amount of melted wax pooling on top of the cuts, and maintaining their shape during molding. By stacking two laser-cut beeswax wafers, we were able to make airtight air channels that were approximately the same thickness as the wood wafers.

We actuated these pneumatic channels in two ways. Since the objects cast with this technique are merely prototypes, we do not embed pumps within the structures of our devices. For most applications, hand-pumping is an appropriate way to test the actuated motion of a design. In these cases, we used a simple hand pump with a tube inserted into the air channel of the silicone device. When automated motion is important to test, for example in cases where a specific programmatic inflating/deflating pattern is needed, a small air pump can be used. We used an affordable DIY-friendly 12V DC air pump motor with an Arduino, and a tube inserted into the air channel [96].

3.4.2 Demonstration

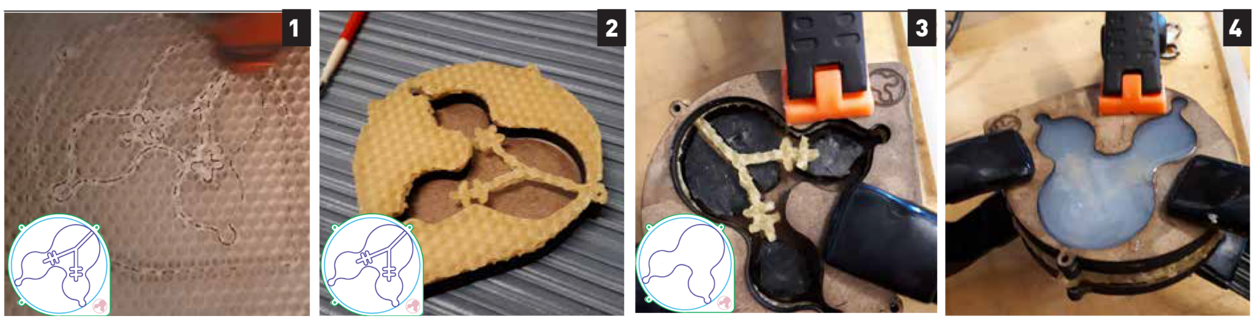

Figure 3.5: Steps to creating an actuated sensate object. (1) Laser-cut beeswax air channels. (2)Stack the air channels with the other layers to elevate them to the right height. (3) Place sensing pads. (4) Cure silicone around everything.

To demonstrate the process of creating an actuated device with our technique, we show how we cast an arbitrarily-shaped device with air channels. The creation of the internal air channel depends on using an extra wax mold layer.

Step 1: CAD Drawing

We began this process by creating a CAD drawing of the various mold layers to be laser-cut. When parametrically designing this geometry, we used the major axes of the shape to guide us in generating a layer for the air channels that are later used for pneumatic actuation.

Step 2: Mold Cutting and Wax Coverage

We then used the drawings to laser-cut the physical mold layers from inch thick wood sheets, which we then coated in paraffin wax in order to prevent the silicone from seeping into the fibers of the wood. We cut the air channel layer from a beeswax sheet, using settings on the laser-cutter that were fast enough to prevent the wax from melting away.

Step 3: Curing and Lost Wax Casting

We placed all of the laser-cut layers together in the correct order, suspending the beeswax air channels at the correct height. We then poured liquid silicone in and left it to cure. Once cured, we removed the object from the mold. We cleared the air channels by heating the object with a heat gun, melting away the wax and leaving empty air channels ready to be actuated—for our prototype, with a hand pump.

Figure 3.6: The final product of our actuator casting demonstration, showing the cast air channels being inflated.

3.5 Making Sensors

3.5.1 Process

The initial recipe for making conductive silicone using carbon fibre came from an Instructables tutorial [97]. To make the conductive silicone pads, we started with Ecoflex silicone [98], which is clear and has a low viscosity, resulting in stretchiness. This silicone mixture comes in two parts (A and B), which remain in liquid form until mixed together. Once mixed, they begin to cure, resulting in solid silicone. The silicone that we used had a 4 hour cure time To make the silicone conductive, we mixed carbon fibre into it. The conductive carbon fibre (CF) silicone pads possess piezoresistive and capacitive properties, while still maintaining some of the flexibility and deformability of silicone. The conductivity of the conductive silicone pad increases as more CF is included, but the material becomes more rigid and fragile. As the proportion of CF to silicone passes a certain threshold, its piezoresistive sensitivity actually decreases, despite the increase in conductivity.

By experimenting with different proportions, we arrived at a balance between electrical sensitivity and softness. We present the recipe for this in Table 3.1. First, the CF must be prepared. We did this by combining 4g of CF with 2mL of isopropyl alcohol in order to dissolve the coating around each piece, encouraging the individual fibres to separate. This combination was mixed by hand until the fibres were separated and the isopropyl had evaporated. By ensuring that the fibres are separate, the network of fibres becomes more complex, as the individual fibres can better contact one another, increasing the overall sensitivity of the pad. Next, we added 28 mL of Ecoflex silicone part A, and 28 mL of part B, and mixed until the mix appeared even. Finally, this was poured into a mold, and cured for the amount of time necessary for the silicone to solidify.

Ingredient	Amount
Carbon Fibre	4g
Isopropyl alcohol	2mL
Ecoflex Part A	28mL
Ecoflex PArt B	28mL

Table 3.1: Our recipe for conductive silicone, mixing Ecoflex silicone with carbon fibre prepared with isoproryl.

3.5.2 Sensing

We focused on four touch sensing modalities (pressure, touch, hover, gesture detection) delivered through a variety of easy-to-implement sensing techniques using a basic microprocessor interface such as an Arduino. This includes capacitive sensing for touch and hover, piezoresistive sensing for pressure, and gesture detection using a random forest classifier.

Pressure can be sensed with resistive sensing, due to the piezoresistive qualities of the carbon fibre networks in the silicone. Under pressure, the fibres come into contact with one

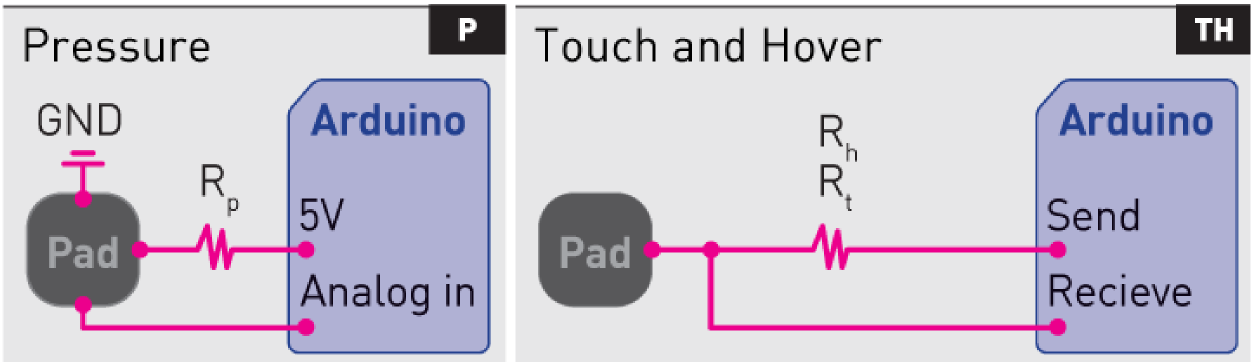

Figure 3.7: Electronic circuits for pressure (left), touch and hover (right) sensing. Pressure sensing is done via piezoresistive sensing, where the change in resistance is measured, while touch and hover use capacitive sensing with two different strengths of resistors. For these tests, Rp=330, Rt=1M, Rh=16M.

another, creating changes in the overall resistance of the material. By taking advantage of this phenomenon, we can sense pressure through the circuit shown in Figure 3.7.

Touch and hover follow the same sensing principle, but differ in the strength of the projected electric field. Both can be sensed through self capacitance. We implemented this by using the Arduino CapacitiveSensor library [99] and the circuit shown in Figure 3.7. The CapacitiveSensor library works by converting two Arduino digital pins into a capacitive sensor with one send pin and one receive pin, where capacitance is measured by the amount of time taken for the receive pin to match the state of the send pin. The only difference between the sensing techniques for touch and hover is the resistance of the circuit; touch sensing requires a smaller valued resistor.

3.5.3 Demonstration

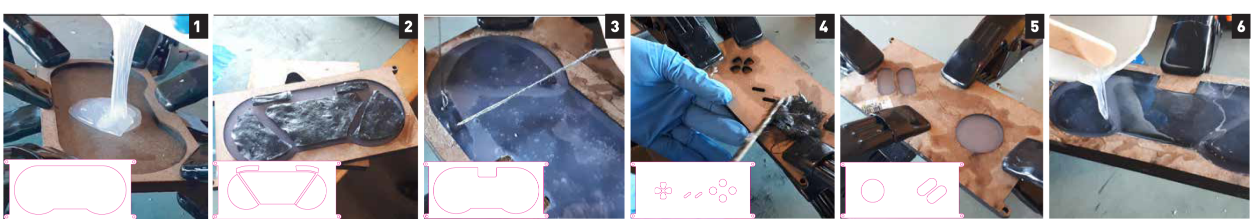

Figure 3.8: Steps to creating a soft sensate game controller. (1) Cure the first layer of dielectric silicone. (2) Place sensing pads. (3) Sew conductive thread into the cured silicone. (4) Cure the controller buttons. (5) Cure the button extrusions on top of the controller. (6) Assemble and cure all sub-components together.

We designed and implemented a soft hand-held game controller, following the design of the Super Nintendo Entertainment System (SNES) controller [100]. The process steps are illustrated in Figure 3.5.3.

Step 1: CAD Drawing

We started this process with the CAD design of the laser-cut mold layers. We included all original buttons of the SNES controller, which we took into consideration by creating a layer with button designs and extrusions on the top of the controller. In addition to the original functionality, we created an embedded layer of pressure sensing pads. We designed the molds using our parametric system where geometric primitives defined controller geometry, and offsets from the outline geometry defined the embedded layer of pressure sensing pads.

Step 2: Mold Cutting and Wax Coverage

We then used the drawings to laser-cut the physical mold layers from inch thick wood sheets. Due to the porosity of the wood, it needed to be sealed in order to prevent the silicone from soaking into the mold while curing, making it more challenging to remove. We accomplished this sealing by covering the inside of each mold layer with a film of molten paraffin wax.

Step 3: Curing and Wiring Process

During the CAD step, we had created a conductive silicone pad mold layer as well. In order to embed these within the larger dielectric silicone body, we first separately cured the pads, and extracted them from the mold. In the meantime, we also cured one layer of dielectric silicone from the main controller body. Once this was fully cured, we placed the pressure-sensing pads on top. We sewed two conductive threads through each pad (one for power and the other for ground), and positioned them so that they were not touching one another. The threads, once sewn in, were further secured through the use of silicone glue - a fast-curing one-component silicone adhesive. We then poured the next two layers of dielectric silicone into the controller body mold and left them to cure. We placed the top layer of touch-sensitive buttons on top, and sewed one conductive thread through each button (for self-capacitive sensing). These threads were again secured in place with silicone glue. Finally, we poured the top layer of dielectric silicone on top and left it to cure. We removed the cured controller from the mold. To prevent conductive threads from contacting one another, we applied heat-shrink tubing to the portion of the threads that were outside of the controller.

Step 5: Sensing

For the top layer of buttons, we used self-capacitive sensing to capture touch information for the direction controls; and for the pads underneath the controller, resistive sensing captured pressure or deformation.

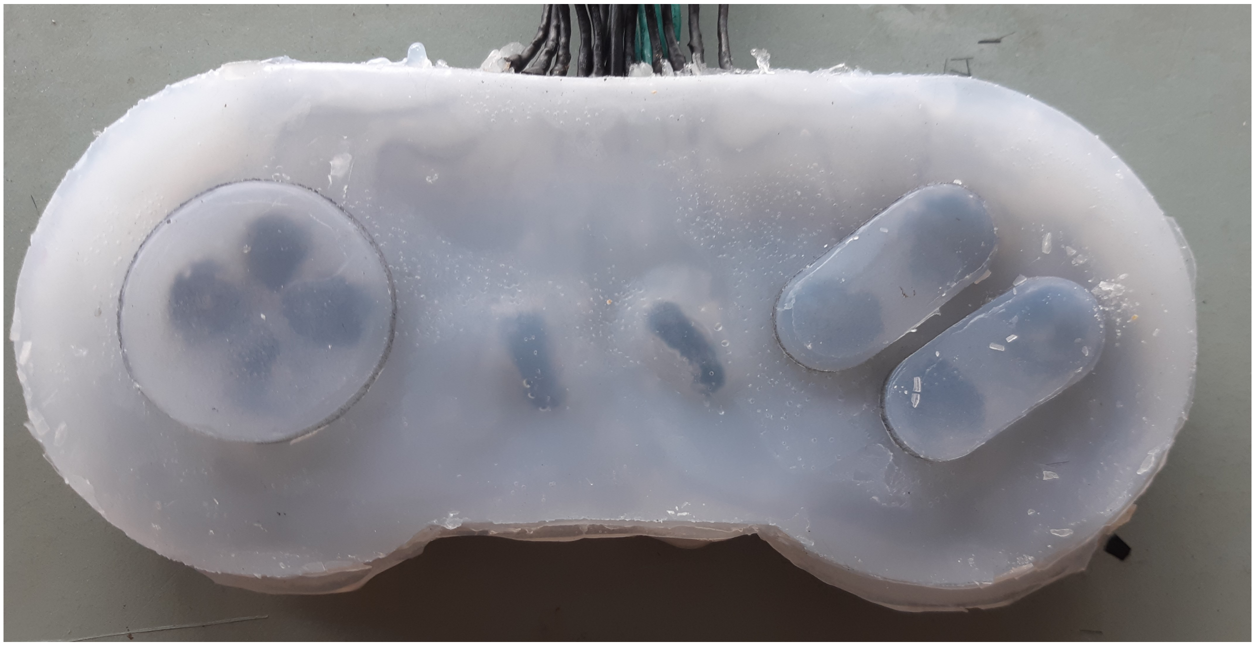

Figure 3.9: The end product of our sensor casting demonstration - a fully soft SNES game controller prototype

3.6 Gesture Recognition Study

3.6.1 Method

Machine learning can be used with data derived from any of these sensing techniques to recognize more complex interactions. To demonstrate, we trained a Random Forest classifier with capacitive data (100 iterations, batch size of 100). The model was trained on 9 gestures + no touch for a total of 10 classes. We defined a gesture set with properties that reflected the range of gestural complexity which we observed in individuals informally interacting with our prototypes. The simplest gesture is to touch an individual sensor pad; this requires the gesture detection system to be able to differentiate the pads from one another. At a slightly greater complexity, we chose gestures such as “bending all arms of the object” and “touching the whole sensor”, where both gestures create large changes in the object’s capacitance and we predicted that differentiation might be more challenging. Finally, the shape of the object

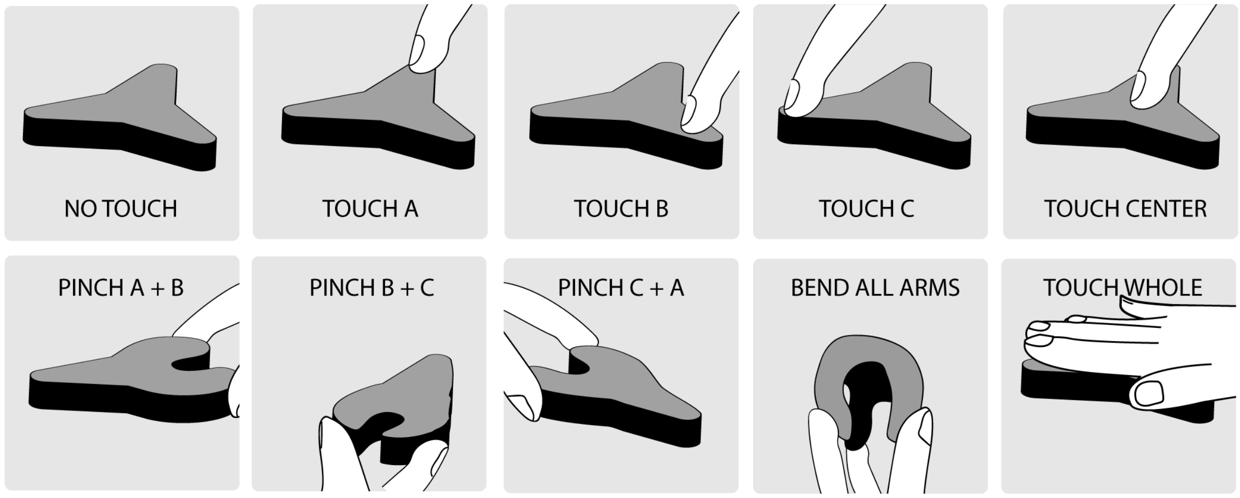

Figure 3.10: Gestures used to train the silicone sensors. A Random Forest classifier was trained on capacitive data for these 10 gesture states.

afforded pinching two arms together, which we predicted would be the most challenging to correctly classify, as it has similarities to touching either of the sensor pads or bending all of the arms. The gestures we used are shown in Figure 3.10. The training data consisted of data collection sessions with four users, each performing 90 gestures (each of the 9 gestures repeated 10 times, in a randomized order).

We conducted a study with 10 participants (7 male, 3 female, ages between 21-40). Each study was 30 minutes long and consisted of three sessions; in each session, the user was asked to perform each of the nine gestures four times, leading to a total of 36 gestures in each session (randomized), and 108 gestures per participant. Between each of the three sessions, the participant was given a one minute break - this was simply in order to allow them to reposition their bodies, introducing more capacitive variation and noise into the recorded data. Within each session, upon the participant performing a requested gesture, the researcher conducting the study would hit a button; recording the last 30 samples read by the Arduino, as well as the live prediction of the system and the "true" classification. This data was automatically recorded into a CSV file for each session. The live predictions of the system were made with the pretrained model (N=4) previously described.

3.6.2 Results

Over all of the sessions (three sessions per participant, 10 participants), the average accuracy score was 92.13%, with the lowest session score being 77.78% and the highest being 100%. Using data collected from all of the sessions, we performed Leave-One-User-Out cross validation. This gave an averaged accuracy score of 95.56%.

	touch arm B	touch arm C	bend all arms	touch center	touch arm A	pinch C+A	pinch B+C	touch whole	pinch A+B
touch arm B	0.94	0.00	0.00	0.00	0.01	0.01	0.03	0.00	0.02
touch arm C	0.00	0.98	0.01	0.00	0.01	0.01	0.00	0.00	0.00
bend all arms	0.00	0.00	0.87	0.00	0.01	0.06	0.00	0.01	0.06
touch center	0.00	0.00	0.00	0.99	0.01	0.00	0.00	0.00	0.00
touch arm A	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
pinch C+A	0.00	0.01	0.02	0.02	0.01	0.94	0.00	0.01	0.00
pinch B+C	0.00	0.01	0.00	0.00	0.00	0.00	0.98	0.01	0.00
touch whole	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.99	0.00
pinch A+B	0.02	0.00	0.07	0.00	0.01	0.00	0.00	0.00	0.91

Figure 3.11: Confusion matrix for gesture recognition for leave-one-out cross validation, percent accuracy averaged across runs (N=10); each run included 12 trials for each of 9 gestures for a total of 108 trials.

In Figure 3.11, we present the confusion matrix for gesture classification. This confusion matrix shows a comparison of the detected gesture type with the true gestures. Red cells show the percentage of true positives, green cells show true negatives, and yellow cells show false positives. Based on this, we can see that the two classifications that were most frequently mistaken for the other are "bend all arms" and "pinch A+B". Another common error was between the "bend all arms" and "pinch C+A" gestures. This is to be expected, as pinching two arms together creates a spike in capacitance that can be mistaken for the spike created by pinching three arms together.

3.6.3 Conclusion

Our goal was to enable designers and makers to create soft, sensate, expressive silicone objects using rapid prototyping methods. Along the way, we explored different ways of describing and structuring the design space, and eventually arrived at our systematic parametric design paradigm, which afforded varied and expressive geometries. Through careful process-level design driven by our attempts at making complex and expressive devices, we arrived at our laser-cut wood/wax molding process and recipes for sensing with silicone. We believe that these methods can help make soft sensor/actuator prototyping more accessible to designers.

Chapter 4

Computational Support for Rapid Mold Prototyping

4.1 Introduction

Turning a 3D model into a laser-cuttable mold is difficult. In Chapter 3, we showed a parametric design approach where the designer must mentally calculate what the “slices” of the object negative (i.e the mold) look like, drawing out each slice with a CAD tool for laser-cutting. This is in contradiction to the way that a 3D object is typically designed. Normally, a designer focuses on the design of the target object (the object positive) rather than the design of its mold (the object negative). Converting between the two in such a way that allows the mold to be laser-cut requires complex spatial thinking, especially as objects become more complicated. Even for an expert, imagining and designing the placement of 3D components in 2D designs is non-trivial. While these 3D to 2D conversions are difficult for a human to do quickly, they are mathematically calculable. This suggests that a computational tool that handles the generation of laser-cuttable patterns from a 3D positive could majorly aid in the design process, making it more possible to rapidly prototype.

In this chapter, we work with a mold-making approach called FoldMold, a method for creating single-use molds from paper and wax, based on paper patterns which can be cut by a 2D numerically controlled cutter. These patterns are joined with papercraft and woodworking techniques, then soaked in wax to solidify them into a smooth-surfaced mold. Through the combination of woodworking techniques and wax smoothing, this method allows for smooth, curvy objects to be cast. Additionally, due to the folding approach of FoldMold (as opposed to the stacking approach of Silicone I/O), less material is used in each mold, and cutting is faster. While the FoldMold technique is not the contribution of this chapter, we present it here in overview to provide context for the requirements and workflow surrounding the computational tool we contribute.

Without computational support, the FoldMold technique requires manual design of all components. This is a task that is not only extremely complicated (as we describe above), but also error-prone, tedious, and time-consuming.

Here, we introduce the FoldMold Blender Add-on, a computational tool incorporated directly into Blender, a popular open-source 3D modeling software. Our main goal with this computational tool is to overcome the bottleneck of mold design within rapid mold prototyping, significantly speeding up the process. By allowing objects modeled in Blender to be converted into laser-cuttable mold patterns within the same environment, we hope to make the design workflow smooth and intuitive; by leveraging designers' pre-existing knowledge of the Blender interface, we aim to make the learning curve for our mold-making tool far easier to manage. We discuss our requirement generation process, through which we learned how to support the overall FoldMold workflow through a combination of automation and user input. We then present our software implementation, walking through its features as we follow the molding process of an example object.

4.2 Background: FoldMold Technique and Workflow

FoldMold is a method for creating single-use molds from paper and wax, based on paper patterns which can be cut by a 2D numerically controlled cutter or even by hand. These patterns are joined with papercraft and woodworking techniques, then soaked in wax to solidify them into a smooth-surfaced mold.

4.2.1 Bending

To make flat paper sheets conform to the curves and bends of 3D shapes, we employ a set of cutting techniques. For sharp bends, the material must be folded to create a clean, sharp edge. This level of precision can be difficult to achieve with manual folding, often resulting in uneven or warped edges – even more so when the material is thick or dense, e.g., cardstock or thicker. Our approach is to cut a small amount of material away. For each fold, scoring (partially laser-cutting the fold line on the outside of the bend) strain-relieves and guides the fold. The depth of the cut line controls the degree of bending. Deep cuts reduce structural strength, requiring a balance. Through trial and error, we found that 50% is ideal for most folds.

While *folding* involves creating one straight line to allow a sharp bend, scoring involves

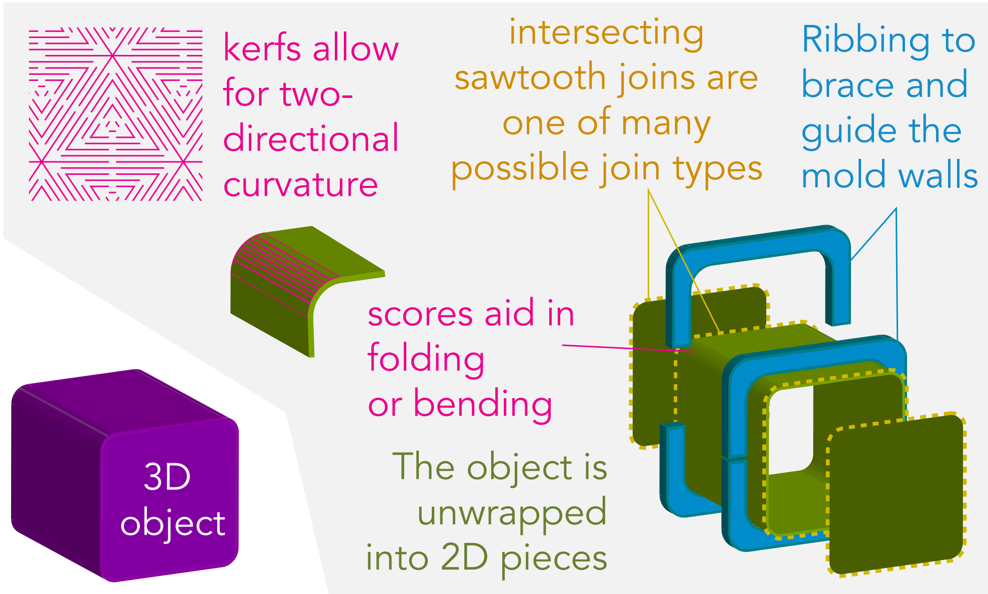

Figure 4.1: Molds can be bent in multiple ways, including simple scores (for 1D curves or sharp edges) or by using a kerfing pattern (for a specific two-directional curve). Joinery features allow for 3D reassembly and ribbing elements to reinforce the shape.

many cuts (or "scores") to create unidirectional curves (Figure 4.1; best demonstrated with a cylinder). As the number of lengthwise scores along the side of the cylinder increases, the cross section begins to approximate a circle. Here, the tradeoff is between cutting time and surface quality—as the density of the scores increases, the curve becomes smoother and less polygonal, but takes more time to cut. This is a parameter that designers may choose to adjust based on their stage in the design pipeline, with speed being critical in rapid exploration and quality becoming more important as the design reaches completion. Further, we can smooth discretized polygonization with wax.

A more complex approach to bending is *kerfing* (Figure 4.2). Like scoring, kerfing depends on removing material. Combining sets of curved or angular cuts in different configurations affords interesting curvatures, ranging from increased flexibility for unidirectional curves to bending material in two directions at once. Because these cuts are discontinuous, they can fully pierce the paper without it falling apart. Increased cut depth increases the paper flexibility.

4.2.2 Joining

To form 3D shapes from 2D patterns, pieces must be joined. Joins must: (1) fully seal the seams such that the casting material will not leak out, (2) maintain the smoothness of the interior, where the casting material will be poured, and (3) be relatively simple to manually

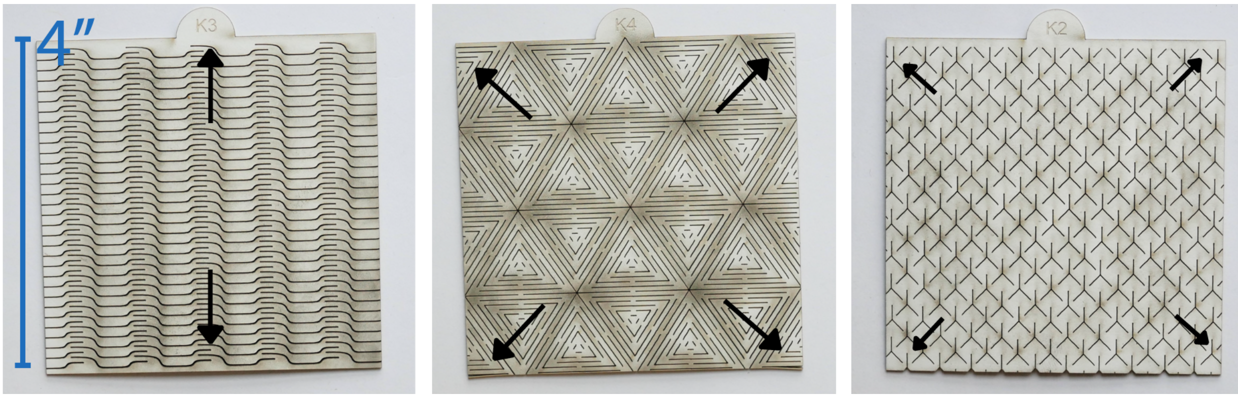

Figure 4.2: Three different kerfing styles. Each enables the material to bend in different ways (arrows). While the middle and right patterns have the same direction of bending, the middle pattern is more flexible. Patterns from [1]; laser-cut onto illustration board, a dense thick paper.

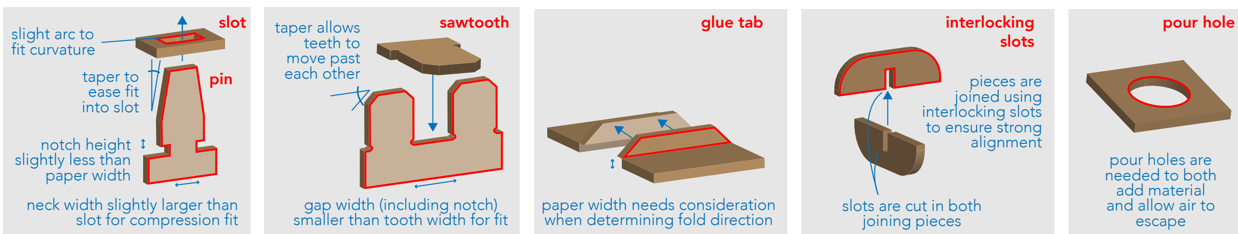

Figure 4.3: Join types. FoldMold uses four join types, and a pour-hole feature. Joins utilize pressure fitting for secure joining and to maintain alignment.

assemble. We implemented multiple joining techniques, including sawtooth joins, pins, and glue tabs (Figure 4.3).

Sawtooth joins rely on pressure fitting in order to create a tight seal. Gaps are slightly smaller than teeth – so they can be push-fit together, and held by friction, an approach made possible by paper's squishability malleability. To ease slipping the teeth into slots, we constructed gentle guiding tapers, with barb-like notches that prevent them from pulling out. This technique constructs joins that face outwards from the model, preserving the interior surface quality.

Pin joins follow a similar principle, with small tabs pushed through slightly undersized slots. Their flange is wider than the corresponding slot to ensure a pressurized, locking fit. They should be used when the adjoining piece needs to be pulled through, e.g. when extra material is advantageous for the design. As with sawteeth, we put tapers and notches into the pins to facilitate assembly.

Glue tabs require an adhesive to stick two flat surfaces together. Glue tabs are wider than both the sawtooth and pin joins, making manual cutting simpler. Keeping the interior of the seam smooth is tricky with this join; though it might be intuitive to overlap the tabs as in typical box construction, this would create a discontinuity on the model's inside surface. Instead, we bend both tabs outwards from the model and paste them together there, like the seam of an inside-out garment. Here, they can be manipulated to reduce mismatch while preserving interior surface quality (Figure 4.3).

4.2.3 Ribbing

Layering wax onto the paper makes it much stronger, but in some cases this is not enough. For example, if a mold is filled with a dense material such as plaster, or has a large volume, internal wall pressure may cause walls to deform or bow. External support can also help to maintain mold element registration (Figure 4.1).

4.2.4 Process

The steps of the FoldMold process are described here.

Step 1: Create a 3D object

First, the designer starts by modelling the 3D object positive in a 3D modeling program. Models can also be found online, imported, and edited as needed.

Step 2: Create Corresponding 2D Patterns

Next, using a CAD tool such as Autodesk Inventor, the maker must create 2D patterns that correspond to the 3D object. This can be done by using the dimensions of the 3D model to draw 2D projections of the faces, deciding where to separate faces by folds or seams.

Step 3: Design FoldMold Features

Once the designer has created a 2D design for the 3D model, they must use a CAD tool to draw joinery features along the edges of their 2D pattern. Bending elements, such as scoring or kerfing, must be drawn on as well. Any necessary ribbing elements must also be drawn.

Step 4: Cut the 2D Patterns

Once the CAD drawings are complete, they can be imported into a program like Adobe Illustrator where colours can be assigned to edges. This step is for ensuring that different cut settings can be applied to, for example, scores and full cuts with a CNC cutter. These patterns can be cut from paper using a laser cutter, vinyl cutter, an X-Acto knife, or even a pair of scissors.

Step 4: Prepare the FoldMold

The maker must now assemble the cut patterns into a 3D mold, set it in wax and verify a smooth surface finish. Parts of the mold that are very small and detailed, or with high 2D curvature, can be 3D printed out of wax then fixed to the main mold once it has been set in wax. Paper can be a very flexible and conformable material, great for molding complex shapes. These same properties make it harder to maintain those shapes once the material has been poured. The key part of the assembly process is setting the paper with wax. This has the dual benefit of increasing mold strength and stiffness, and sealing the paper and all joins. To build up the mold strength, the maker repeatedly dips it in a pool of molten wax. As the wax hardens, it stiffens the paper. This essentially "locks" in the shape. The repeated dipping builds layers of wax, successively increasing stiffness. Depending on the size of the piece, between one and ten dips might be needed, taking between 10 and 30 minutes altogether, depending on the size of the piece. Dipping a formed or partially assembled sheet of paper in molten wax saturates the porous paper, and prevents the cast material from seeping through or into the paper. It also makes the wax-soaked paper a convenient non-stick surface that is easy to remove from the cast object. Detailed surface textures can be 3D printed in wax and tacked to the wax-dipped mold.

Techniques that rely on removing material, such as scoring and kerfing, would leave the surface bumpy if left uncovered, and show up in the cast object. Wax dipping prevents this as well, filling and smoothing cuts. For very fine areas, dipping may obscure desired detail or dull sharp angles. In such cases, at mold-cleaning time we use a small brush to add wax, or a hot tool to melt away excess wax and to "burnish" the surface.

Step 5: Pour and Set

Once the mold has been assembled and set with wax, the casting material can be poured – then we wait for it to set. This can be materials that cure or set (silicone, epoxy resin, jello), or materials that dry (plaster, concrete). The mold can be taken apart by ripping the paper and peeling it away from the cast object. The wax impregnation makes this go easily. Any excess wax crumbs that stick to the object can be mechanically removed or melted away with low heat.

4.3 Computational Tool Requirements and Features

The requirement generation process for the computational tool was an iterative one. Any technical approach comes with inherent opportunities and limitations, and thus with FoldMold. We conceptualized FoldMold with the explicit objectives of supporting rapid molding for curvy objects that can be made with common industrial machining techniques. This folded/cut paper based approach to these molding challenges comes with certain strengths and limitations, which we present in Table 4.1. We used this assessment to prioritize how we would offer computational support, from the general making workflow to features supported and where the interface needs to support designer interventions.

Type	Strength	Limitation
Scale	*Large objects:* The FoldMold technique scales very well for large objects (up to a 1 meter scale) that would be difficult to rapidly create with other techniques.	*Very fine, detailed surfaces:* Areas of an object that are small yet intricate can be better molded by 3D printing and integrating it with the rest of the paper mold.
Curvature	*Cylindrical curves:* The flexibility of paper enables the molding of cylindrical (one-dimensional) curves. This is enhanced through scoring, allowing thicker, more structural papers to be used as well.	*Tight two-dimensional curves:* Although kerfing can be used to accomplish a 2D curve, this is suited to large radii. Tight 2D curves with small radii are not possible to create out of paper without removing significant amounts of material.
Fidelity	*Smooth surface finish:* The use of wax on the paper results in the cast object separating easily from the mold, creating a smooth surface finish. Any imperfections in the molding material can be smoothed with wax.	*Highly precise objects:* Our goal is to enable rapid iteration of objects rather than molding precise models. Because manual assembly involves some small level of human error, we cannot guarantee precision in each mold.

Table 4.1: Strengths and limitations of the FoldMold technique for scale, curvature, and fidelity.

4.3.1 FoldMold Feature Support

Perhaps the most obvious of these requirements is that the program must support the components of the FoldMold technique, which includes bending, joining, and ribbing. For bending, the tool must support folds and scoring (for cylindrical bends). We found that more complex kerfing was not useful to the construction of most of the objects we molded with the FoldMold technique, as it only supports slight 2D curves - highly curved 2D surfaces are better supported by 3D printing. Given the limited scope of complex kerfing, and the range of objects that scoring could accomplish, we chose to exclude it from the computational tool. For joining, the computational tool must support sawtooth joins, pin joins, and glue tabs. Each of these joins is best suited to a specific joining scenario, so it was important that all of these join types were possible to use with the computational tool. Ribbing is a component that is challenging to design manually, as support pieces must conform directly to the outside of the positive in order to register, align, and support segments of the mold. As such, ribbing needed to be included in the computational tool design.

4.3.2 Full Automation

Our goal in developing this computational tool was to better support rapid prototyping by reducing the time required to design a mold. The most impactful way to do this would be to have a system that fully automates the design process. In 4.3.3, we discuss cases where full automation is not ideal and user intervention is necessary. However, as a rapid prototyping system it is crucial that the *option* to fully automate design exists. That is, if a designer does not wish to intervene, the system must be able to automatically generate a working solution.

4.3.3 User Intervention

At first, we created a version of the FoldMold Blender Add-on that was highly automated. The angles of the face normals of the 3D model were analyzed to find curved areas for scoring, edges for seams were automatically found by the Blender UV Unwrapping feature, and join types were automatically assigned. Ribbing slices were distributed across the object at equal distances. Using this system, we soon found that as designers, we needed to have direct control of the output geometries and components. While automation made things much faster and created a sense of freedom, it was frustrating to not be able to directly manipulate the elements of the add-on output. Certain decisions that were obvious to us with our broader understanding of the prototyping context were overlooked by the automated process. An example of this is that the amount of external support needed varies based on the casting material (consider the deformations caused by foam and concrete, for example), which is not considered by the program. Another reason that a designer may wish to intervene is if their prototyping goals differ from the assumptions of the program. For example, if the goal of the prototype is to create a rough approximation of the model, the automatically applied scoring is not only unnecessary, but also adds extra cutting time.

It became clear to us that the FoldMold Blender Add-on needed to allow user intervention for major components. That is, the user should be able to specify (1) the join type to be placed on each seam (2) areas to score, and the density of those scores and (3) areas to support with ribbing. Of course, this could create an opposite, but still important problem: by requiring so much user input, the system could become tedious and time-consuming to use. This suggested that the system should have options for control and automation, requiring neither but allowing both. If a user wants to provide input, they should have a way to do so; if not, there should be some level of automation to fall back on.

4.3.4 System (Computational and Cutter) Compatibility

The FoldMold program, which takes as input a 3D model, must be compatible with pre-existing 3D modeling software. In order to be accessible to makers and DIY designers, this must be a free, widely available software that runs on most computers. For this reason, we built the FoldMold program as a Blender Add-on, as Blender is a widely used, free open-source 3D modeling tool that is broadly compatible, allowing 3D models constructed in other programs to be imported.

The FoldMold mold-making technique is designed to be compatible with a variety of cutters. These range from a laser-cutter (found in well-equipped makerspaces), to a vinyl-cutter (accessible enough for makers to have in their own workshops or homes), to a pair of scissors or an Xacto knife. Similarly, the FoldMold Blender Add-on must be compatible with the same range of cutters.

One element of cutter compatibility is for the generated 2D patterns to be vector files with the lines formatted in such a way that they can be read by the cutter software and converted to G-code instructions.

Another element of cutter compatibility is registration. Often, flat sheets of paper need to be scored from both sides, for example for sheets that have both convex and concave curves at different points. Laser-cutters and vinyl cutters can only cut one side of the material at a time, and a human with an Xacto knife has the same limitation. Because of this, it can be hard to align the 2D pattern to be cut in the correct location on both sides without registration marks of some kind.

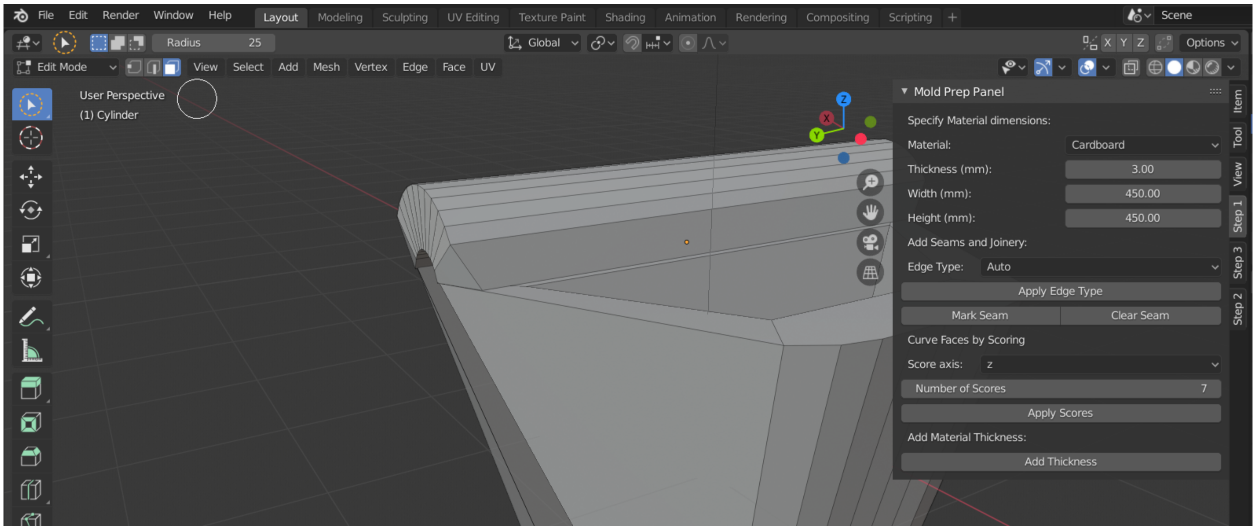

Figure 4.4: The FoldMold Blender Add-on.

4.4 Implementation

We wrote the FoldMold Blender Add-on using Blender's Python API. In this section, we dive into the details of our implementation.

4.4.1 Workflow

In Figure 4.5 we present the steps involved in using the FoldMold Blender Add-on to create a mold. The user workflow using the Add-on differs from the flow of the code, described in Sections 4.4.2 - 4.5.5. Here, we describe the user workflow and map it to the section on its technical implementation.

At the very start of the user workflow, (1), the designer must create a 3D model. This can be directly created in Blender, or alternatively created in a different 3D modeling program and imported into Blender. Next (2) is the seam and joinery creation process. This step is optional; if skipped, the FoldMold Blender add-on will automatically choose seams and apply glue tab joinery to each seam. If the designer wants to intervene, they can select seam edges and choose between pin joins, glue tabs, or sawtooth joins. We discuss the technical implementation of this in Section 4.4.2. Then, (3) if the object is intended to be curved, scoring can be applied to smooth the curves. This step is optional as well, and is simply for improving curve fidelity. The automatic process does not generate scoring. We describe the technical implementation in Section 4.4.3. The next step (4) is to generate ribbing. Automatically generated ribbing includes 3 ribs along the X axis and 3 ribs along the Y axis, with 2 Z axis ribs holding them in place for support. The designer can optionally intervene to choose the ribbing density, as well as positioning and orienting ribbing pieces to best support the given geometry. We discuss the implementation of this in Section 4.4.4. Next, (5) the 3D geometry is unfolded by the Add-on into a 2D pattern that can be cut. This is described in Section 4.4.5. Finally, the designer must manually assemble these piece and pour the casting material.

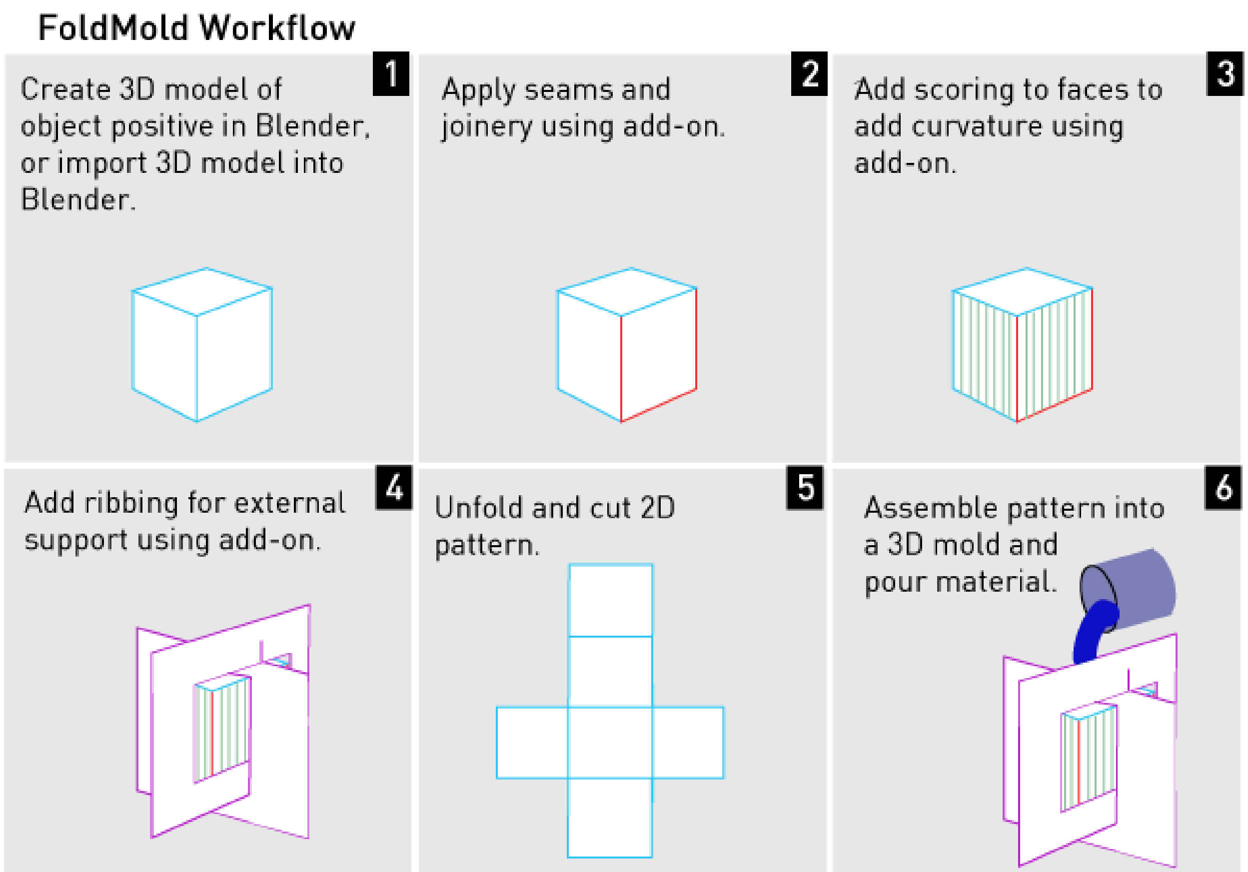

Figure 4.5: Using the FoldMold Blender Add-on to create a mold. Steps 1-4 show digital steps involving the Add-on, and steps 5-6 include physical cutting, assembly, and casting.

4.4.2 Joinery

In 4.2.2, we described the function of joinery for connecting edges to one another and sealing them. Here, we discuss how our code generates joinery patterns and places them on the edges of the mold.

Along the outside edges of each island, joinery is generated to allow the unfolded pattern to be reassembled into a 3D mold. In our program, this is done by generating joinery patterns and aligning them to the corresponding edge. These patterns include glue tabs, sawtooth joins, or pin joins. The default joinery is a glue tab, but users can assign any pattern to any edge, with the potential of having all three joinery types on a single model.

The basic components of each joinery pattern are referred to as *tiles*. Each tile comes from an SVG file, where the shape of the component is defined through a set of points. Our program parses the elements of an SVG file in order to extract the points. An advantage of using SVG images is that the system can easily be extended to include more join types simply by drawing SVG images of the tiles.

Combinations of tile components are referred to as patterns. Each type of joinery is composed of several tiles. For example, in a sawtooth join, the tiles include a tooth tile and a gap tile, which are arranged in an alternating pattern along an edge, and in an inversely alternating pattern along the matching edge such that the two edges fit together. Each pattern is generated for a specific edge. This is important for two reasons: first, the number of tiles placed must correspond to the length of the edge, and second, matching edges must fit with one another. For example, for a pin join to be assembled, one edge must have pins placed along it while the other edge must have holes for the pins to be inserted into.

$$\begin{gathered} \vec{u} = < i, j > \\ \begin{bmatrix} i, -j \\ j, i \end{bmatrix} \end{gathered} \tag{4.1}$$

Equation 4.1: The rotation matrix applied to joinery patterns, where u is the unit vector in the direction of the UVEdge

Once a pattern has been created, it must be rotated and positioned along its edge. We accomplish this by constructing a 2x2 rotation matrix (Equation 4.1) based on a unit vector in the direction of the edge (perpendicular to its normal vector). We then loop through all of the vertices in the pattern, multiplying this rotation matrix by their coordinate vectors. To position the pattern correctly along the edge, we translate the pattern vertices in between the start and end vertex of the edge.

4.4.3 Bending

In 4.2.1, we introduce the role of bending in the FoldMold technique, including folds and scoring. Here, we describe their technical implementation. In order to reconstruct a 3D mold from the flattened 2D layout, the faces must be bent and joined. Bends are possible due to folds, which are the non-seam edges in between faces - that is, the internal edges of each island. These edges are preserved throughout the process of unwrapping, and no joinery is added to them. When exporting the final PDF file to be cut, the fold lines are all exported with a different colour than the cut lines, so that the laser cutter applies different power settings when cutting. The laser cutter colour settings must be adjusted in order to only score fold lines (partial cut) rather than cutting all the way through the material.

In certain cases (such as when a curved 3D model is defined with a low polygon count), simply folding along the edges of a geometry may result in a very low-fidelity surface finish. In these cases, users may choose to adjust the score density of faces in the 3D object (see Section 4.5.2). This density must be defined before the object is exported and unwrapped. If a score density has been set, our program creates additional fold lines across those faces, resulting in a smooth curve once cut. Along with the density, the user also defines a score direction.

4.4.4 Ribbing

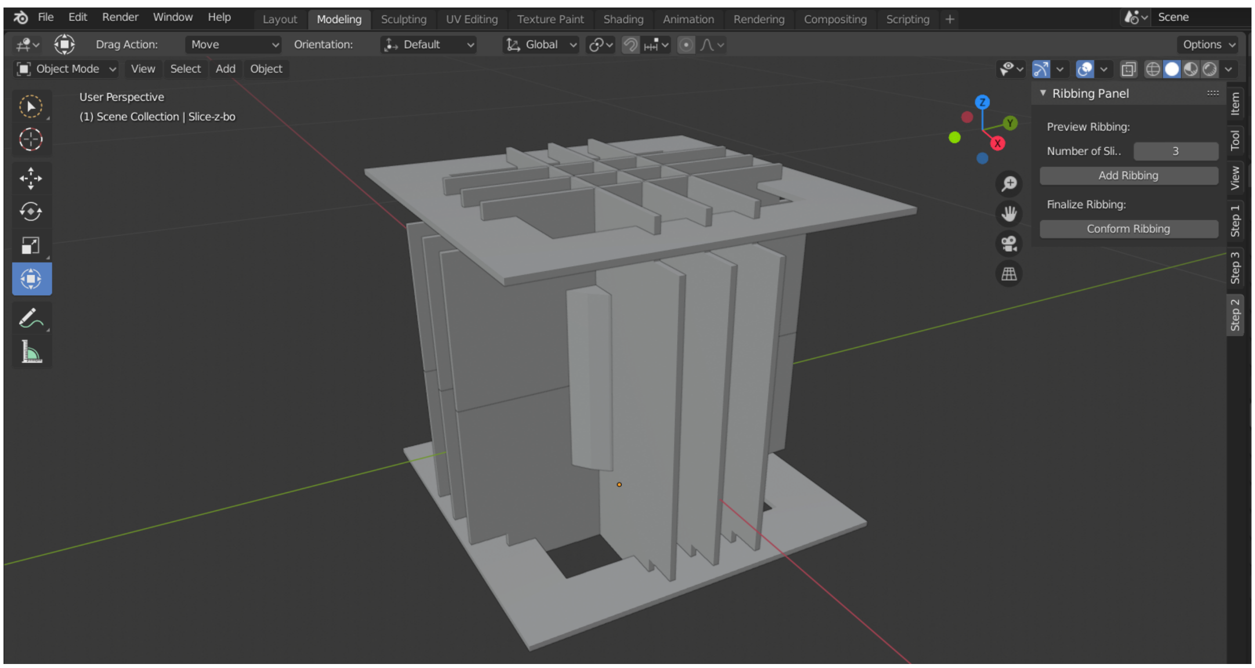

Figure 4.6: An object created in the FoldMold Blender Add-on with ribbing along the x, y, and z axes for maximal support. These ribbing sheets are designed to slot into one another for easy assembly.

In Section 4.2.3 we describe how ribbing (external support pieces) can be used to maintain a mold's structure and registration throughout the casting process. In the FoldMold Blender Add-on, ribbing must be added along three axes. The reason for this is that multi-directional ribbing adds maximal support for the mold, and the ribs slot together and hold one another strongly in place.

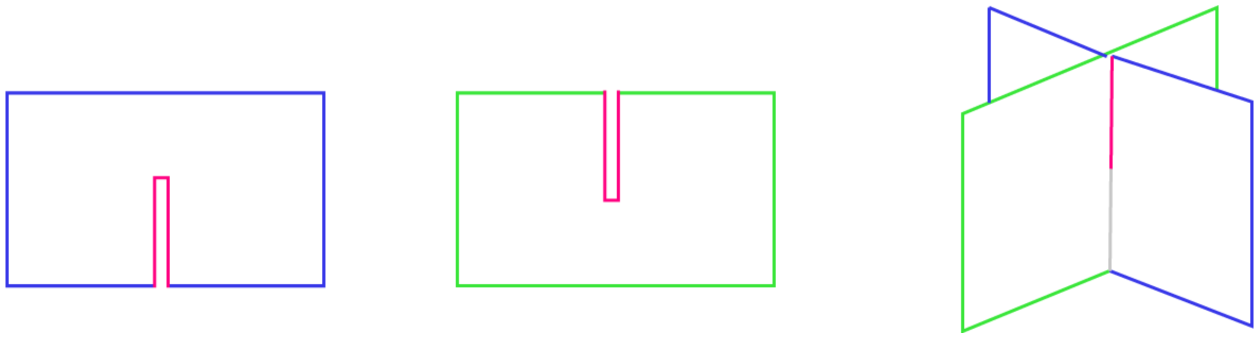

Figure 4.7: Cutting slots into ribbing slices along different axes allows them to interlock easily.

The ribbing that the FoldMold Blender Add-on generates (Figure 4.6) follows the interlocking principle shown in Figure 4.7. In the FoldMold Blender Add-on, X and Y ribs slot together, and Z ribs slot around both X and Y ribbing. While the XY ribbing fully supports the mold, adding Z axis ribbing keeps the XY ribbing sheets registered relative to one another, holding them firmly in place.

Of course, it is physically impossible to assemble ribbing that goes all the way around a model, as the model would have to pass through it. To overcome this, we split each ribbing sheet in half, allowing the halves to be joined together to surround the model. By varying the split directions, we ensure that the ribs fully interlock while still being possible to assemble.

At this stage, the ribbing slices intersect with the 3D model, going right through it. By clicking the "Conform Ribbing" button on the FoldMold interface, the user launches the process of fitting the ribbing sheets to the outside of the mold. This is done by looping through all of the ribbing sheets and performing boolean difference operations between the ribs and the mold, essentially cutting an outline of the mold into each rib.

4.4.5 UV Unwrapping

In the FoldMold technique (without computational support), a designer must mentally convert a 3D model into a flat 2D layout. Here in the computational tool this is handled by the UV Unwrapping portion of the code, which automatically converts a 3D model into an flattened 2D pattern. UV Unwrapping is an application of UV Mapping, which is the 3D modeling concept of mapping 2D patterns to 3D surfaces. "U" and "V" are the axes in the UV coordinate space, which is the coordinate space of object *surfaces*. While "X", "Y", and "Z" specify an object's position in 3D space, "U" and "V" specify a location on the surface of a 3D object. UV Unwrapping uses UV coordinate information to create a 2D pattern corresponding to a 3D surface, essentially "unwrapping" it.

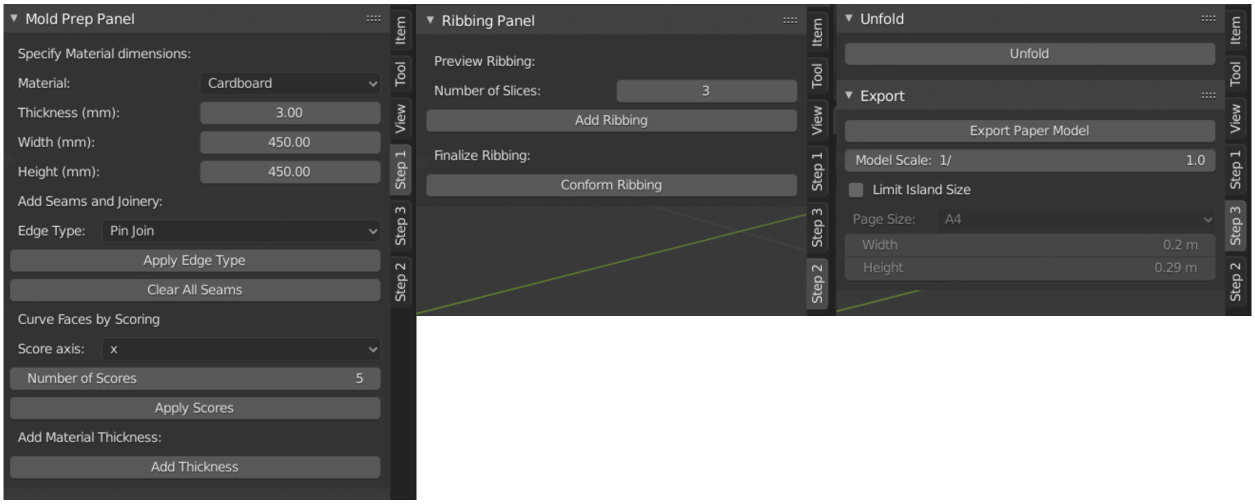

Figure 4.8: The user interface of the FoldMold Blender Add-on is based on three panels. The first panel (Mold Prep Panel) allows the user to specify a material, apply seams and joinery types, and create scores. The second panel (Ribbing Panel) supports the generation and conforming of ribbing elements, allowing the user to specify ribbing density. The third panel (Unfold) exports the mold into 2D patterns.

At this stage, the Blender mesh object (the 3D model) is converted into a 2D "unfolded" pattern. Our implementation draws from the "Export Paper Model from Blender" add-on [101] from which we use the UV unwrapping algorithm. Initially, the 3D model is processed as a set of edges, faces, and vertices. We then reorient the faces of the 3D object, as if unfolded onto a 2D plane. Next, we then alter the edges, faces, and vertices to be in a UV coordinate space.

A product of the unwrapping process is a set of islands. Islands are groups of connected faces, and are separated by seam edges, meaning that if every edge were a seam, each island would be composed of just one face. While these seams are automatically generated during unwrapping, they can optionally be user-defined. Each island has a bounding box, which is used to fit it to a page. If an island's bounding box exceeds the size of a page, it will be rotated to better fit. If that is not possible, an error will signal to the user that the object is too large to fit on the page. This can be remedied by scaling the 3D model or by defining more seams, resulting in more (but smaller) islands.

4.4.6 User Interface

We created the user interface of the FoldMold Blender Add-on in a way that reflects the step-wise process of creating a FoldMold. We divided this into three main steps, with each step having its own panel (Figure 4.8). The first panel (Mold Prep Panel) allows the user to specify a material, apply seams and joinery types, and create scores. The second panel (Ribbing Panel) supports the generation and conforming of ribbing elements, allowing the user to specify ribbing density. The third panel (Unfold) exports the mold into 2D patterns.

4.5 Demonstration: Using the FoldMold Blender Add-on for a Kitchen Grip Mold

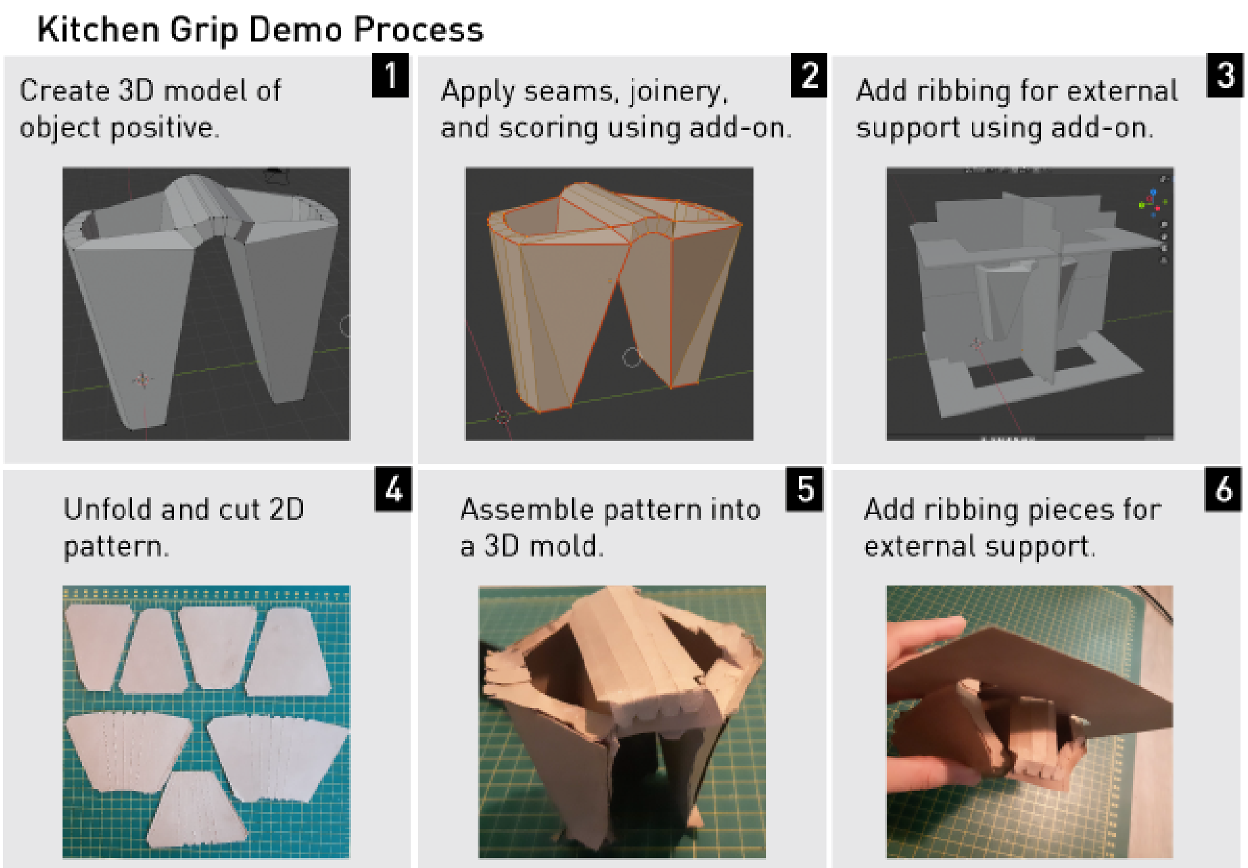

Figure 4.9: Our demonstration of the FoldMold Blender Add-on for creating a kitchen grip mold, showing the process from 3D model to a physical mold.

To demonstrate the functionality of the FoldMold computational tool, we chose to mold a simple kitchen grip. We chose this specific item due to its many different curves, showing a need for both scoring and ribbing. Here, we walk through the steps of molding a kitchen grip from 3D modeling in Blender to mold assembly.

4.5.1 Creating the 3D Model

In our process, 3D modeling relies entirely on the Blender system - at this stage, the FoldMold Add-on does not need to be used. Though a large variety of Blender operations can be used to create object meshes, we started from a cube and used a variety of mesh and vertex manipulations to create this shape. These manipulations included translation, scaling, poly-build, triangulation, mirroring, and more.

Once we completed the 3D model of the positive, it was ready to start converting into a FoldMold. The first point at which we used the FoldMold Add-on was for adding the material thickness to the model. The object that we had modelled is what we wanted as an end-result after casting, which meant that the mold had to be positioned on the outside of the object, acting as a shell. Because we wanted to create this mold out of chipboard, we selected this option in the FoldMold interface and applied the material thickness to the model.

4.5.2 Scoring

Scoring is used in the FoldMold method to add curvature and smooth curved areas. As part of our process in modeling the kitchen grip, we needed to create curves along the top as well as along the sides (see Figure 4.9). To do this, we used the FoldMold Add-on to apply scoring to the top by selecting the Y axis as the scoring direction, and setting score density to 4 (creating 4 scores) for a tight curve. For the sides, we wanted to achieve a more shallow curvature. We again created 4 scores, which were more spread out due to the larger face size. This time, we had the Z axis selected.

4.5.3 Choosing Seams and Joinery

We wanted to customize the joinery types and locations to be optimal for our model. First, we wanted to place sawtooth joins on all of the perpendicular seams, because sawtooth joins are very quick to assemble and work well at 90 degree angles. We did this by using the FoldMold “Edge Type” option (Figure 4.8). Next, we chose glue tabs for all of the flat joins on the model, as glue tabs keep the interior surface of the seam smooth and flat, creating a nice surface finish on the cast object. Along all of the edges where we selected join types, the FoldMold Add-on automatically applied seams, shown in red (Figure 4.9).

4.5.4 Ribbing

The final step before exporting the mold was adding support material to ensure that the mold would not deform when pouring and curing the casting material. We added these support pieces using the FoldMold Ribbing Panel. Due to the small size of our kitchen grip prototype, we did not need to very strongly support it, so we generated one slice of ribbing along the X and Y axes. The FoldMold system automatically generated ribbing slices along the Z-axis to hold the X and Y ribbing in place (Figure 4.9). The default positions worked well, so we chose not to manually move the slices before conforming the ribbing to the mold.

4.6 Pilot Study

We conducted a pilot study to gain insight into the following research questions:

1. How long does it take to create a mold pattern using the FoldMold Blender Add-on, and how does this compare to a user's expectations?

2. How do users feel about their ability to customize the mold to their needs? In what way could their control over the outcome be improved?

3. What obstacles currently exist in the FoldMold Blender Add-on?

This study was conducted with the approval of the UBC Behavioural Research Ethics Board (certificate number H13-01620-A021).

4.6.1 Method

We recruited one participant for this pilot study. We selected this participant for his familiarity with 3D modeling - while he had limited experience with Blender itself, he had used similar programs. The entire study was conducted remotely through a recorded Zoom call over a span of approximately 40 minutes.

We started the study by introducing the FoldMold technique and its various components, including joins, bends, and ribbing. We explained these concepts through photographs and verbal descriptions of how they work, answering any questions that came up. Once the participant felt that they understood the FoldMold technique, we introduced the FoldMold Blender Add-on. We explained the purpose of the Add-on and demonstrated how it can be used. The participant then practiced by creating joins, bends, and ribbing for a sample object (a cube) in order to gain familiarity with the controls.

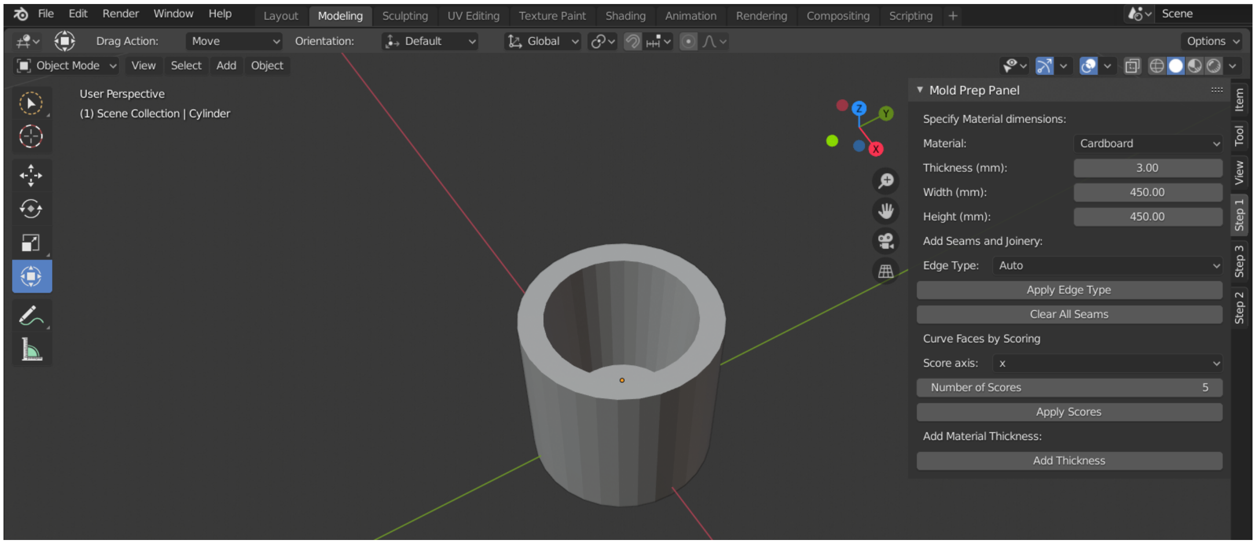

Figure 4.10: We presented this 3D model (a drinking glass) to the participant and asked them to create a mold design for casting it.

Next, we showed the participant a simple 3D model of a glass (4.10), and asked about how much time they would expect to spend on creating a FoldMold for casting this object using a 3D modeling program with and without the FoldMold Blender Add-on, in order to gain insight into how the Add-on compares to existing tools.

We then asked the participant to use the Add-on to create a mold design for this object, explaining their decisions for the mold as they made them. Once they created the mold design, we asked them to open the exported layouts and check to make sure that the layouts aligned with their expectations.

Afterwards, we asked the participant about their feelings on their ability to control the outcome and customize the mold to their needs, and asked about ways in which the Add-on could be improved.

4.6.2 Results

RQ1: How long does it take to create a mold pattern using the FoldMold Blender Add-on, and how does this compare to a user's expectations?

Upon seeing the 3D model of the glass, the participant estimated that it would take "most of a day" to create a mold pattern for it using a typical 3D modeling program, and around 2 minutes using the FoldMold Add-on. In reality, using the FoldMold Blender Add-on it took around 4 minutes to create the mold pattern, from start to export. We note that this was after the participant spent around 2-3 minutes practicing with another 3D model, with the researcher having demonstrated the functionalities prior to the practice round.

RQ2: How do users feel about their ability to customize the mold to their needs? In what way could their control over the outcome be improved?

Overall, the participant was satisfied with his ability to customize the mold. He felt that the process of choosing seams and joinery was “very intuitive and very easy to do”. On the other hand, the placement of the layouts on the paper from which they will be cut was completely out of his control, and he would have liked to manually position them on the page.

Throughout the process of building the mold, the participant was able to implement their decisions about the joinery, scoring, and ribbing. However, we observed that at certain points, system recommendations would have been helpful, especially for a novice user of the FoldMold technique. For example, the participant predicted that sawtooth joins would be the most appropriate for the rounded segments of the model. While there is no indication of this in the Add-on, when actually assembling the FoldMold, sawtooth joins can leave small gaps in a rounded mold where the material may leak out. For such cases, the system should recommend pin joins.

RQ3: What obstacles currently exist in the FoldMold Blender Add-on?

The participant pointed out that it was difficult to find the Add-on menus, especially given the very high number of controls already present in the Blender interface. He suggested that icons could help in making the menus more visible.

There were two points in the mold-creation process where warnings would have been helpful. The first point is the mode switch in between creating seams/scores and ribbing. While seams and scores are created when the model is in Edit mode, ribbing is generated in Object mode, and the user must manually switch in between these modes. However, the Add-on does not make this clear. The second point was when creating ribbing. Due to the geometry of the object and placement of the ribs, twisted geometries were created, resulting in excess pieces being including in the exported layout of the ribbing. Ideally the system would be robust to these errors and would prevent them, or provide warnings.

4.7 Discussion on Computational Tool Performance

Here, we discuss how well the FoldMold Blender Add-on met the requirements that we had initially set out to fulfill. We made these assessments based on the pilot study as well as our own experiences, which we informally gathered as a team while we were using this tool for creating molds over a span of approximately six months.

4.7.1 FoldMold Feature Support

The FoldMold Blender Add-on was successful at supporting most of the features of the FoldMold technique. The joinery features were fully implemented, and ribbing was supported very strongly as well. For bending, folds and scores were both included; of course, kerfing was excluded from our implementation. Though this was not a major hindrance for the models we were interested in creating, it did mean that we could not mold 2D curvatures.

One aspect of this that could be improved in the future is the placement of the pouring hole (the hole through which the casting material is poured into the mold). In the current FoldMold Blender Add-on, the pouring hole is placed on the top face of the model, but this does not necessarily ensure that it is accessible, as it may intersect with the ribbing.

4.7.2 Full Automation and User Intervention

We found that the balance of user control and freedom worked well for us; we were able to adjust the options as needed or allow them to default to predefined settings. Improved visualization could have increased our sense of control. Specifically, for assigning joinery types, having the joinery visualized along the edges would be helpful feedback to have as a user. Even something as simple as different colours being assigned to edges to distinguish joinery types would have helped. In the current version, all joinery types are displayed as red edges, which can lead to confusion.

As we found in the pilot study, being able to arrange the layouts would increase the sense of control. Currently, the Add-on automatically places pieces next to one another in a somewhat arbitrary way, in order to fit as many pieces as possible on each page. This is not always the best solution; for example, a maker may want to group pieces together based on their relation to one another.

4.7.3 System Computational and Cutter Compatibility

We implemented this tool as an Add-on to the Blender software, which allowed it to be accessible to everyone on our team, for both Mac and PC systems. We tested the generated patterns on two kinds of cutters, including a laser-cutter (Trotec Speedy 300) and a vinyl cutter (Silhouette Vinyl Cutter). The generated patterns were compatible with both of these systems.

4.8 Conclusion

One of the most significant challenges in rapidly prototyping molds is quickly designing the mold patterns to be cut. Design is rarely a fast process; mentally converting object positives to negatives and 3D objects to 2D layouts makes this much more tedious and time-consuming, especially for DIY makers who may not have design expertise. For this reason, we created the FoldMold Blender Add-on, a computational tool that handles the generation of laser-cuttable patterns from a 3D positive to aid in the design process, making it more possible to rapidly prototype molds. The resulting tool was very helpful within our own mold-making practice. To evaluate our progress towards the requirements we had initially established, we conducted a pilot study and reflected on our team's experiences with it. We found that FoldMold features were well supported, but could be made more robust to geometric challenges (such as the pouring hole intersecting with ribbing). We accomplished full automation while also allowing user intervention. User control over the exported layout could have been increased. Our implemented system is accessible and compatible with a variety of cutters.

Chapter 5

Discussion

At the start of this research, we set out to accomplish the following goals. Here, we revisit these goals and evaluate our progress towards them.

1. Speed and Usability: minimizing design time, making time, and amount of manual work and assembly, such that it enables rapid prototyping.
2. Accessibility for a DIY maker: using low cost, low waste materials that are easy to find, and equipment/resources that can be found in a standard workshop.
3. Production of high-performance prototypes: capturing the intended object complexity, whether in enabling interactivity or smooth curvature and surface finish. These do not have to be polished end products, but objects that meet the designer's prototyping goals.

5.1 Speed and Usability

In this thesis we showed two different methods of making the design process faster and easier. The first method is through using 2D CNC-based mold construction (Silicone I/O and FoldMold making methods), making molds quick to create.

The second method is through developing a computational tool that automatically generates mold designs from a 3D model of the positive, dramatically decreasing design time to a matter of minutes. This method is based on the FoldMold technique, which unfolds molds into flat patterns that are very quick to cut and usually easy to assemble. While manual assembly can occasionally be time-consuming depending on the geometry of the object and the design of the joinery and ribbing, these options can be adjusted to achieve faster assembly times. However, this may sometimes be at the expense of precision or surface finish.

5.1.1 Parametric Constraints Are Helpful in Rapid Exploration

In developing the Silicone I/O making method, we wanted to support exploratory, expressive design. That is, we were aiming to support designers in creating molds that were arbitrarily shaped and interesting to interact with. Taking a powerful constraint-based approach, where the designer is given access to manipulation of carefully chosen parametric constraints was a great way to accomplish this.

Though these constraints may initially appear restrictive and not suited to an exploratory system, we found them to be quite the opposite. By limiting design to a 2D layer-based format, the barrier to designing a physical object was lowered significantly, as it is challenging to design along multiple axes simultaneously. The parametric approach presented a limited set of options for the placement of each 2D primitive, and as a result was a great method of discovering geometries that a designer may not have initially thought to explore.

Our experience with the FoldMold computational tool reflected this as well. While we did not take a parametric constraint-based approach as we did with Silicone I/O, providing default starting points in the software and allowing the parameters to be adjusted was helpful in making decisions about mold design.

5.1.2 Mold Design is Highly Automatable

Though designing a 3D model of an object positive requires design expertise and human decision-making, the process of converting that object into a mold is mainly algorithmic. Given a set of user-defined vertices, edges and faces, we can compute how to place a material around the outside of the model, how to unfold that material, where to define seams, and how to create ribbing that physically fits together. These are all design tasks that would take a tremendous amount of design work and time in order to get right. By offloading this work to a computational tool, the process is made much faster and potentially more precise.

We iterated and made careful design decisions around the balance between automation and user control. While most components of generating a mold are automatable, there may be unique properties of individual geometries that the system either does not take into account, or makes non-ideal decisions for. For example, our pilot study revealed the need for control over the positioning of generated layouts. In our experience, allowing a designer to adjust and alter the automatic decisions along the way as needed allows them to create their intended mold without undergoing a long or tedious process.

In this thesis, we implemented automation for the FoldMold technique. Reflecting back on the Silicone I/O making technique, there are opportunities for automation there as well. For example, recommendations can theoretically be automatically generated for sensor pad and air channel placement based on the object geometry, reducing the designer's cognitive load.

5.2 Accessibility for a DIY Maker

5.2.1 Mold Material Choice Lowers the Barrier to Iteration

The Silicone I/O making technique uses wood (such as hardboard) which is cheap, easy to find, and easy for a maker to dispose of after use, as it is biodegradable. Beeswax and paraffin wax are similarly cheap, biodegradable, and easy to access; they are often found in grocery stores. The FoldMold making technique uses paper and wax, which are similarly biodegradable and accessible. In these regards, both techniques are very accessible to a DIY maker. Because these materials are cheap and create minimal waste, they are suitable for one-time-use molding and therefore enable designers to iterate on their mold designs.

On the other hand, laser-cutters are not always found in a DIY maker's workshop. While community studios and maker spaces usually have laser-cutters that can be booked at affordable rates, not all makers have access to these resources. By using materials that are less dense than wood (such as paper or cardstock), a vinyl cutter (an affordable machine that makers can have at home) could be used instead to cut these molds.

While the making techniques we introduced are more accessible than, for example, 3D printing, this comes with a trade-off. The precision that can be achieved with our techniques is lower than 3D printing, and the making process is more hands-on, requiring manual assembly. Of course, we recognize that our techniques are first-generation techniques that cannot easily be compared to a technology that has had decades of development, and we are hopeful that with the manual making process will become faster and easier with further development. An example of a way to do this could be to automatically optimize seams and joinery placements to reduce manual assembly work.

5.2.2 Leveraging Well-Known Software Lowers the Learning Curve

We built the FoldMold computational tool as an Add-on to Blender in order to allow the mold-making process to fit within designers' pre-existing modelling workflows. An added benefit of this is that the familiarity of the interface makes it easier to learn how to use the FoldMold Add-on, especially for designers who have no previous familiarity with the FoldMold technique. We took advantage of this by following the Blender user interface convention of panels, and allowing the mold, ribbing, and seams to be directly manipulated as Blender meshes.

While this made the learning curve much more manageable (as demonstrated by our pilot study), the challenge of learning the FoldMold process remained. We attempted to overcome this by adding stepwise instructions to the interface. In the future, this needs to be tested more extensively with designers who are new to the FoldMold technique in order to evaluate the clarity of our communicated process.

5.3 Production of High-Performance Prototypes

5.3.1 Wax Improves Surface Finish

We found that by soaking wood/paper in molten wax, the surface finish of cast objects were greatly improved. This is because the wax fills the pores of the materials, smoothing them as a result. The “filling” behaviour of wax was especially an advantage when paper was scored, as it resulted in seamless curves. While this smoothness was an aesthetic benefit, it also enabled the easy removal of cast objects from their molds, by preventing the casting material from seeping into the mold fibres.

We tested the limits of wax’s smoothing capabilities by experimenting with many different casting and molding materials. For casting materials, we tested plaster, chocolate, ice, silicone, resin, and expanding foam. For most of these materials, wax had a smoothing effect. Chocolate was an exception to this – due to the melting temperature of the chocolate, this was not a cold-casting process, and it resulted in the wax melting slightly into the cast object and causing the mold to cling to the cast object. Expanding foam also presented a challenge for surface finish; while the wax-soaked molds helped with the surface finish to some extent, the bubbly nature of the foam itself resulted in a very porous texture.

For molding materials, we experimented with hardboard, cardboard, illustration board, matboard, copy paper, and chipboard. Wax successfully acted to improve the surface finish on all of these materials, with the only exception being scored cardboard. Corrugated cardboard has empty pockets of air between its flat sides, which, when exposed (i.e. through scoring), require a lot of wax in order to be filled.

5.3.2 Layered Molds Support Placement of Internal Components

Positioning sensing components and pneumatic actuation channels within a cast structure poses a physical challenge. Because the material density of carbon-infused silicone and beeswax differ so greatly from the density of silicone, when placed within a mold these components often sink or float to a different vertical height than intended. We overcame this in the Silicone I/O making method by enforcing a layer-by-layer curing approach. Placing sensor pads and beeswax channels atop pre-cured layers made it possible to directly cure the components at the correct vertical positioning, either by curing the component layer before pouring the remainder of the silicone or by fastening them to the pre-cured layers and filling the rest of the mold.

A drawback of this approach is that it increases the overall curing time. While in some cases this may be negligible (e.g. with a fast-curing silicone and one component layer), it becomes tedious for molds with more components, detracting from the rapid-prototyping workflow. A possible solution to this could be to incorporate wax supports that connect each component to its corresponding wooden layer, allowing it to be held in place while silicone is poured and cured around it. The wax could then be melted away, and the resulting holes could be filled with silicone to create a seamless end result.

5.3.3 Papercraft and Wax Enable Curvy Mold Creation

Using the scoring feature of the FoldMold Blender Add-on, one can generate patterns for material to be strategically cut away, enhancing the flexibility of paper in a controlled way. Of course, most kinds of paper are already quite bendable, which puts the structure and precision of the intended curves at risk of deformation. However, dipping the molds in wax combats this issue, locking the curves in their intended position. Not only does the wax provide structural support - it also helps to smooth the surface finish of the curves.

While wax can help provide some structure, its strength is limited. To ensure that the curves are held in place under the weight of the casting material, the ribbing feature of the FoldMold Blender Add-on can be used to generate patterns for external support pieces to be cut out of paper.

Chapter 6

Conclusion and Future Work

To date, rapid prototyping methods for castable objects that are interactive (sensate and actuated) or curvy have been relatively unexplored, creating a bottleneck in iterative design for complex tangible objects. Being faced with this challenge ourselves, we were inspired to investigate and develop solutions that were appropriate for makers and designers with limited equipment and resources.

This work establishes two approaches for increasing speed in mold prototyping. The first method is by creating flat, laser-cuttable mold patterns, significantly speeding up the actual mold creation process due to the object discretization as well as the relative speed of a laser-cutter compared to a 3D printer. The second method is by automating mold design, off-loading much of the tricky and tedious design work to a computer software that can help a maker design a mold in a matter of minutes. However, this method still requires manual mold assembly, which adds to the overall prototyping time.

The methods we introduced here make molding more achievable for makers and designers with limited machinery, resources, and engineering expertise. The main materials we used for molding were wood, wax, and paper - all easy to find and inexpensive. Due to the low barrier to accessing these materials as well as their biodegradability, they are appropriate for one-time-use molding and therefore encourage rapid iteration.

These techniques also make it possible to cast complex objects that previous rapid prototyping techniques do not entirely enable. In Chapter 3, we demonstrated how the Silicone I/O technique can be used to create sensate and actuated devices cast entirely out of silicone, and in Chapter 4 we used the FoldMold technique to cast smooth curvy shapes by leveraging the flexibility of paper.

In this thesis, we have taken steps towards making rapid mold prototyping for complex castables possible, and we hope future work will build on our findings. Here, we outline future directions to be explored.

6.1 Performance - Rapid Lost Wax Casting

In the Silicone I/O making method, we used laser-cut beeswax sheets to cast channels for pneumatic actuation. This is a novel technique - laser-cut wax has not been explored before, and it significantly improves the speed of the rapid lost wax casting process. However, our use of it is quite limited, and could be expanded in future work for better performance in actuation.

The typical approach to lost wax casting includes creating silicone molds into which molten wax can be poured, and removed once it is solid to reveal a wax structure. Of course, this is a lengthy process, as the silicone mold itself must first be cast. A more prototyping-friendly approach is 3D printed wax. In the FoldMold making method for example, we used 3D printed wax to create fine surface details or portions of the mold that could not be created with paper. However, wax 3D printing (similar to other kinds of 3D printing) can achieve a high level of detail but is a slow process that does not scale well. Existing methods of lost wax casting are usually not amenable to a rapid-prototyping workflow.

Within the scope of this thesis, we used laser-cut wax to create simple geometries, and encountered challenges that would have been amplified in more complex cuts. Specifically, the heat of the laser cutter causes the wax to melt beneath the beam of the laser, causing a loss in fidelity. Often, the molten wax simply hardens back in the same position, filling any gap created by the laser beam and essentially sealing the cuts. For the simple geometries that we created, we were able to overcome this by increasing the speed of the laser cutter in order to minimize melting; however, some melting still occurred, preventing us from cutting very fine or very detailed shapes.

This problem could potentially be solved by designing a laser cutting algorithm that starts at an offset and gradually cuts towards the intended shape, thus allowing the wax to simply melt away at the edges rather than re-sealing cuts. Future work should explore algorithms for laser-cutting wax, as well as characterizing the use of lost wax casting in a rapid prototyping context.

6.2 Speed - Rapid 3D Design Exploration

In this thesis we presented the FoldMold Blender Add-on, which supported the creation of 2D mold patterns from a 3D object. The use of this computational tool is focused on a point in the design process at which a 3D model has already been created - that is, mid-to-late design stages. At early stages, design work is often exploratory, drawing inspiration from a wide range of geometries before constructing a 3D model. This is a slow process that can be a hindrance to rapid prototyping.

In Chapter 3, we show how the parametric design approach of Silicone I/O can be helpful in encouraging design exploration by forming relations between primitives. Applying a similar approach to a 3D, computerized context could potentially aid in the early stages of rapid mold prototyping.

For example, future work could explore the use of automatically generated recommendation "previews" of 3D shapes based on sets of 3D primitives. An alternative approach could be to produce recommendations by generating variations on created shapes, aiding in rapid design iteration.

6.3 Sensate, Actuated, *and* Curvy Castables

In this thesis, we explored rapid mold-making for interactive and curvy objects separately. While this was helpful as a way to characterize their challenges and create focused solutions, future work should investigate rapid mold-making for objects that are both interactive *and* curvy. This direction of future work combines all of our original objects, including increasing speed and performance of molding while still prioritizing accessibility.

6.3.1 Making Methods

In Chapter 3 we demonstrate a making method for creating sensate and actuated objects (Silicone I/O), and the technique we use in Chapter 4 (FoldMold) enables the creation of curvy objects. While these techniques appear to be quite distinct, they share a common element - wax. Because the FoldMold approach is based on wax-soaked paper, it can be integrated with lost-wax casting in order to create channels for actuation as well as sensing. Wax has a natural tendency to stick when melted, meaning that wax channels can be adhered to the interior of a FoldMold.

In the Silicone I/O method, we used sensor pads made with carbon fibre, which were placed inside the silicone objects as they were being cast. Ongoing work is exploring the use of carbon black (a fine carbon powder), which - though more expensive than carbon fibre - has a lower viscosity when mixed with silicone. The advantage of this is that it can theoretically be injected into the channels created by a lost-wax casting approach, making it possible to integrate into a geometrically complex object.

This poses an entirely new set of challenges. Rapidly registering and correctly angling wax components within a curvy object is non-trivial. Support structures need to be designed in order to reinforce the strength of the wax against the casting material in order to prevent breakage or deformation. Future work should investigate these challenges and establish methods of accomplishing lost-wax casting for complex castables in an accessible rapid prototyping context.

6.3.2 Computational Support For Designing Internal Components

In this work we explored the use of computational support and automation for designing molds for curvy castables. When combining curvature with sensing and actuation, the components of the mold become much more tedious and challenging for a human to rapidly design. This increases the need for a computational tool to speed up the design process.

Software for rapidly designing circuitry has been developed and researched in the past [40], but not for circuitry and actuation components that are to be created within a geometrically complex object. The unique challenge of this kind of rapid mold design is that it can be incredibly difficult to predict how sensing and actuation will be impacted by its surrounding shape. For example, a capacitive sensor will have different readings through varying material thicknesses, which a complex, curvy object is likely to have. Similarly, pneumatic actuation can look and feel drastically different through different material thicknesses.

While it is not easy for a human to design these components, their relatively deterministic nature makes them possible to mathematically model and computationally visualize. Future work should inspect the extent to which the design of curvy interactive castables can and should be computationally supported in order to enhance a designer's rapid prototyping workflow.

Bibliography

[1] T. Materials, *Laser Kerf Cut*, Trotec, 2020. [Online]. Available: https://www.trotec-materials.com/material-usage-hints/bending-trotec-materials

[2] L. Hunt, "The long history of lost wax casting," *Gold bulletin*, vol. 13, no. 2, pp. 63–79, 1980.

[3] D. Olsen. [Online]. Available: https://marketing.metaltek.com/smart-blog/history-of-metal-casting

[4] J. L. Meikle, *American plastic: a cultural history.* Rutgers University Press, 1995.

[5] "A brief history of diy, from the shed to the maker movement," Apr 2020. [Online]. Available: https://www.sciencemuseum.org.uk/objects-and-stories/everyday-wonders/brief-history-diy

[6] S. McKellar and P. Sparke, *Interior design and identity.* Manchester University Press, 2012.

[7] "Yours for the making." [Online]. Available: https://www.instructables.com/

[8] [Online]. Available: https://www.youtube.com/

[9] D. Goldberg, "History of 3d printing: It's older than you think [updated]," Dec 2018. [Online]. Available: https://www.autodesk.com/redshift/history-of-3d-printing/

[10] T. Valkeneers, D. Leen, D. Ashbrook, and R. Ramakers, "Stackmold: Rapid prototyping of functional multi-material objects with selective levels of surface details," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 687–699. [Online]. Available: https://doi.org/10.1145/3332165.3347915

[11] P. J. Bártolo, *Stereolithography: materials, processes and applications.* Springer Science & Business Media, 2011.

[12] T. D. Ngo, A. Kashani, G. Imbalzano, K. T. Nguyen, and D. Hui, "Additive manufacturing (3d printing): A review of materials, methods, applications and challenges," *Composites Part B: Engineering*, vol. 143, pp. 172 – 196, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359836817342944

[13] J. Jiang, J. Stringer, X. Xu, and R. Y. Zhong, "Investigation of printable threshold overhang angle in extrusion-based additive manufacturing for reducing support waste," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 10, pp. 961–969, 2018.

[14] D. E. Weisberg, "The engineering design revolution," *CADHistory. net*, 2008.

[15] J. Breland, "3d printing concrete structures and verifying integrity of their g-code instructions: Border wall a case study," 2017.

[16] Person. [Online]. Available: https://www.thomasnet.com/articles/custom-manufacturing-fabricating/mechanical-vs-laser-cutting/

[17] R. A. Sobieszek, "Sculpture as the sum of its profiles: François willème and photosculpture in france, 1859–1868," *The Art Bulletin*, vol. 62, no. 4, p. 617–630, 1980.

[18] J. Milewski, G. Lewis, D. Thoma, G. Keel, R. Nemec, and R. Reinert, "Directed light fabrication of a solid metal hemisphere using 5-axis powder deposition," *Journal of Materials Processing Technology*, vol. 75, no. 1, pp. 165 – 172, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092401369700321X

[19] "What is cnc plasma cutting?" [Online]. Available: https://www.manufacturingnetwork.com/knowledgebase/view/36

[20] D. Krajcarz, "Comparison metal water jet cutting with laser and plasma cutting," *Procedia Engineering*, vol. 69, pp. 838–843, 2014.

[21] T. Alderighi, L. Malomo, D. Giorgi, N. Pietroni, B. Bickel, and P. Cignoni, "Metamolds: Computational design of silicone molds," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201381

[22] C. Kriesi, Øystein Bjelland, and M. Steinert, "Fast and iterative prototyping for injection molding – a case study of rapidly prototyping," *Procedia Manufacturing*,

vol. 21, pp. 205 – 212, 2018, 15th Global Conference on Sustainable Manufacturing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2351978918301471

[23] P. Mognol, D. Lepicart, and N. Perry, "Rapid prototyping: energy and environment in the spotlight," *Rapid prototyping journal*, vol. 12, no. 1, pp. 26–34, 2006.

[24] 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*. Honolulu, Hawaii: ACM, 2014, pp. 273–280.

[25] 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*, ser. UIST '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 13–22. [Online]. Available: https://doi-org.ezproxy.library.ubc.ca/10.1145/2642918.2647388

[26] P. Cignoni, N. Pietroni, L. Malomo, and R. Scopigno, "Field-aligned mesh joinery," *ACM Trans. Graph.*, vol. 33, no. 1, Feb. 2014. [Online]. Available: https://doi-org.ezproxy.library.ubc.ca/10.1145/2537852

[27] P. Baudisch, A. Silber, Y. Kommana, M. Gruner, L. Wall, K. Reuss, L. Heilman, R. Kovacs, D. Rechlitz, and T. Roumen, "Kyub: A 3d editor for modeling sturdy laser-cut objects," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi-org.ezproxy.library.ubc.ca/10.1145/3290605.3300796

[28] T. Roumen, J. Shigeyama, J. C. R. Rudolph, F. Grzelka, and P. Baudisch, "Springfit: Joints and mounts that fabricate on any laser cutter," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 727–738. [Online]. Available: https://doi-org.ezproxy.library.ubc.ca/10.1145/3332165.3347930

[29] C. Zheng, E. Y.-L. Do, and J. Budd, "Joinery: Parametric joint generation for laser cut assemblies," in *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, ser. C and C '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 63–74. [Online]. Available: https://doi-org.ezproxy.library.ubc.ca/10.1145/3059454.3059459

[30] 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*. Seoul, Korea: ACM, 2015, pp. 1799–1806.

[31] "Basics of mold making tutorials." [Online]. Available: https://www.smooth-on.com/howto/basics-mold-making/

[32] O. D. Yirmibesoglu, J. Morrow, S. Walker, W. Gosrich, R. Cañizares, H. Kim, U. Daalkhaijav, C. Fleming, C. Branyan, and Y. Menguc, "Direct 3d printing of silicone elastomer soft robots and their performance comparison with molded counterparts," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, IEEE. Livorno, Italy: IEEE, 2018, pp. 295–302.

[33] M. Konaković, K. Crane, B. Deng, S. Bouaziz, D. Piker, and M. Pauly, "Beyond developable: Computational design and fabrication with auxetic materials," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 89:1–89:11, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925944

[34] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel, "Computational design of mechanical characters," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 83:1–83:12, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2461912.2461953

[35] V. P. C and D. Wigdor, "Foldem: Heterogeneous object fabrication via selective ablation of multi-material sheets," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 5765–5775. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858135

[36] S. Olberding, S. Soto Ortega, K. Hildebrandt, and J. Steimle, "Foldio: Digital fabrication of interactive and shape-changing objects with foldable printed electronics," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ser. UIST '15. New York, NY, USA: ACM, 2015, pp. 223–232. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807494

[37] R. Ramakers, K. Todi, and K. Luyten, "Paperpulse: An integrated approach for embedding electronics in paper designs," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 2457–2466. [Online]. Available: http://doi.acm.org/10.1145/2702123.2702487

[38] V. Savage, S. Follmer, J. Li, and B. Hartmann, "Makers' marks: Physical markup for designing and fabricating functional objects," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ser. UIST '15. New York, NY, USA: ACM, 2015, pp. 103–108. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807508

[39] V. Savage, X. Zhang, and B. Hartmann, "Midas: Fabricating custom capacitive touch sensors to prototype interactive objects," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '12. New York, NY, USA: ACM, 2012, pp. 579–588. [Online]. Available: http://doi.acm.org/10.1145/2380116.2380189

[40] D. Groeger and J. Steimle, "Lasec: Instant fabrication of stretchable circuits using a laser cutter," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3290605.3300929

[41] A. Spielberg, A. Sample, S. E. Hudson, J. Mankoff, and J. McCann, "Rapid: A framework for fabricating low-latency interactive objects with rfid tags," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 5897–5908. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858243

[42] "Software for product design: Fusion 360." [Online]. Available: https://www.autodesk.ca/en/products/fusion-360/overview

[43] X. L. Cang, P. Bucci, A. Strang, J. Allen, K. MacLean, and H. S. Liu, "Different strokes and different folks: Economical dynamic surface sensing and affect-related touch recognition," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15. New York, NY, USA: ACM, 2015, pp. 147–154. [Online]. Available: http://doi.acm.org/10.1145/2818346.2820756

[44] S. Takamatsu, T. Yamashita, T. Imai, and T. Itoh, "Fabric touch sensors using projected self-capacitive touch technique," *Sens. Mater*, vol. 25, pp. 627–634, 2013.

[45] M. Inaba, Y. Hoshino, and H. Inoue, "A full-body tactile sensor suit using electrically conductive fabric," *Journal of the Robotics Society of Japan*, vol. 16, no. 1, pp. 80–86, 1998.

[46] Y. Zhou, C. Wan, Y. Yang, H. Yang, S. Wang, Z. Dai, K. Ji, H. Jiang, X. Chen, and Y. Long, "Highly stretchable, elastic, and ionic conductive hydrogel for artificial soft electronics," *Advanced Functional Materials*, vol. 29, no. 1, p. 1806220, 2019.

[47] C.-C. Kim, H.-H. Lee, K. H. Oh, and J.-Y. Sun, "Highly stretchable, transparent ionic touch panel," *Science*, vol. 353, no. 6300, pp. 682–687, 2016.

[48] Y. Zhang, J. Ye, H. Wang, S. Huang, and H. Wu, "A flexible tactile sensor with irregular planar shape based on uniform electric field," *Sensors*, vol. 18, no. 12, p. 4445, 2018.

[49] H. Phung, P. T. Hoang, C. T. Nguyen, T. D. Nguyen, H. Jung, U. Kim, and H. R. Choi, "Interactive haptic display based on soft actuator and soft sensor," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 886–891.

[50] H. A. Sonar, M. C.-S. Yuen, R. Kramer-Bottiglio, and J. Paik, "An any-resolution distributed pressure localization scheme using a capacitive soft sensor skin," in *IEEE RAS International Conference on Soft Robotics (RoboSoft)*, no. CONF, 2018.

[51] S. C. Mannsfeld, B. C. Tee, R. M. Stoltenberg, C. V. H. Chen, S. Barman, B. V. Muir, A. N. Sokolov, C. Reese, and Z. Bao, "Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers," *Nature materials*, vol. 9, no. 10, p. 859, 2010.

[52] J. Kost, M. Narkis, and A. Foux, "Resistivity behavior of carbon-black-filled silicone rubber in cyclic loading experiments," *Journal of applied polymer science*, vol. 29, no. 12, pp. 3937–3946, 1984.

[53] H. Zhou, Z. Xia, X. Wang, Z. Li, T. Li, and F. Guo, "Effect of shape and size of nickel-coated particles fillers on conductivity of silicone rubber-based composites," *Polymer Composites*, vol. 36, no. 8, pp. 1371–1377, 2015.

[54] H.-K. Kim, S. Lee, and K.-S. Yun, "Capacitive tactile sensor array for touch screen application," *Sensors and Actuators A: Physical*, vol. 165, no. 1, pp. 2–7, 2011.

[55] T. S. Saponas, C. Harrison, and H. Benko, "Pockettouch: through-fabric capacitive touch input," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011, pp. 303–308.

[56] M. S. Sarwar, Y. Dobashi, C. Preston, J. K. M. Wyss, S. Mirabbasi, and J. D. W. Madden, "Bend, stretch, and touch: Locating a finger on an actively deformed

transparent sensor array," *Science Advances*, vol. 3, no. 3, 2017. [Online]. Available: https://advances.sciencemag.org/content/3/3/e1602200

[57] T. Wang and T. Blankenship, "Projected-capacitive touch systems from the controller point of view," *Information Display*, vol. 27, no. 3, pp. 8–11, 2011.

[58] V. Mosser, J. Suski, J. Goss, and E. Obermeier, "Piezoresistive pressure sensors based on polycrystalline silicon," *Sensors and Actuators A: Physical*, vol. 28, no. 2, pp. 113–132, 1991.

[59] C. Randell, I. Andersen, H. Moore, S. Baurley *et al.*, "Sensor sleeve: sensing affective gestures," in *Ninth International Symposium on Wearable Computers–Workshop on On-Body Sensing*, 2005, pp. 117–123.

[60] P. Parzer, F. Perteneder, K. Probst, C. Rendl, J. Leong, S. Schuetz, A. Vogl, R. Schwoediauer, M. Kaltenbrunner, S. Bauer, and M. Haller, "Resi: A highly flexible, pressure-sensitive, imperceptible textile interface based on resistive yarns," in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '18. New York, NY, USA: ACM, 2018, pp. 745–756. [Online]. Available: http://doi.acm.org/10.1145/3242587.3242664

[61] T. Giorgino, P. Tormene, and S. Quaglini, "A multivariate time-warping based classifier for gesture recognition with wearable strain sensors," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 4903–4906.

[62] F. Aezinia, Y. Wang, and B. Bahreyni, "Touchless capacitive sensor for hand gesture detection," in *SENSORS, 2011 IEEE*. IEEE, 2011, pp. 546–549.

[63] G. Singh, A. Nelson, R. Robucci, C. Patel, and N. Banerjee, "Inviz: Low-power personalized gesture recognition using wearable textile capacitive sensor arrays," in *2015 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE, 2015, pp. 198–206.

[64] A. Braun, T. Dutz, and F. Kamieth, "Capacitive sensor-based hand gesture recognition in ambient intelligence scenarios," in *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '13. New York, NY, USA: ACM, 2013, pp. 5:1–5:4. [Online]. Available: http://doi.acm.org/10.1145/2504335.2504340

[65] M. Sato, I. Poupyrev, and C. Harrison, “Touché: Enhancing touch interaction on humans, screens, liquids, and everyday objects,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’12. New York, NY, USA: ACM, 2012, pp. 483–492. [Online]. Available: http://doi.acm.org/10.1145/2207676.2207743

[66] D. Rus and M. T. Tolley, “Design, fabrication and control of soft robots,” *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[67] C. Lee, M. Kim, Y. J. Kim, N. Hong, S. Ryu, H. J. Kim, and S. Kim, “Soft robot review,” *International Journal of Control, Automation and Systems*, vol. 15, no. 1, pp. 3–15, 2017.

[68] S. Seok, C. D. Onal, K.-J. Cho, R. J. Wood, D. Rus, and S. Kim, “Meshworm: a peristaltic soft robot with antagonistic nickel titanium coil actuators,” *IEEE/ASME Transactions on mechatronics*, vol. 18, no. 5, pp. 1485–1497, 2012.

[69] H.-T. Lin, G. G. Leisk, and B. Trimmer, “GoQBot: a caterpillar-inspired soft-bodied rolling robot,” *Bioinspiration & Biomimetics*, vol. 6, no. 2, p. 026007, apr 2011. [Online]. Available: https://doi.org/10.1088%2F1748-3182%2F6%2F2%2F026007

[70] M. Calisti, M. Giorelli, G. Levy, B. Mazzolai, B. Hochner, C. Laschi, and P. Dario, “An octopus-bioinspired solution to movement and manipulation for soft robots,” *Bioinspiration & Biomimetics*, vol. 6, no. 3, p. 036002, jun 2011. [Online]. Available: https://doi.org/10.1088%2F1748-3182%2F6%2F3%2F036002

[71] M. D. Lima, N. Li, M. Jung de Andrade, S. Fang, J. Oh, G. M. Spinks, M. E. Kozlov, C. S. Haines, D. Suh, J. Foroughi, S. J. Kim, Y. Chen, T. Ware, M. K. Shin, L. D. Machado, A. F. Fonseca, J. D. W. Madden, W. E. Voit, D. S. Galvão, and R. H. Baughman, “Electrically, chemically, and photonically powered torsional and tensile actuation of hybrid carbon nanotube yarn muscles,” *Science*, vol. 338, no. 6109, pp. 928–932, 2012. [Online]. Available: https://science.sciencemag.org/content/338/6109/928

[72] B. Kim, M. G. Lee, Y. P. Lee, Y. Kim, and G. Lee, “An earthworm-like micro robot using shape memory alloy actuator,” *Sensors and Actuators A: Physical*, vol. 125, no. 2, pp. 429–437, 2006.

[73] L.-Z. Chen, C. Liu, C. Hu, and S. Fan, "Electrothermal actuation based on carbon nanotube network in silicone elastomer," *Applied Physics Letters*, vol. 92, no. 26, p. 263104, 2008.

[74] R. F. Shepherd, A. A. Stokes, J. Freake, J. Barber, P. W. Snyder, A. D. Mazzeo, L. Cademartiri, S. A. Morin, and G. M. Whitesides, "Using explosions to power a soft robot," *Angewandte Chemie International Edition*, vol. 52, no. 10, pp. 2892–2896, 2013.

[75] M. Khoo and C. Liu, "Micro magnetic silicone elastomer membrane actuator," *Sensors and Actuators A: Physical*, vol. 89, no. 3, pp. 259–266, 2001.

[76] P. Polygerinos, Z. Wang, K. C. Galloway, R. J. Wood, and C. J. Walsh, "Soft robotic glove for combined assistance and at-home rehabilitation," *Robotics and Autonomous Systems*, vol. 73, pp. 135 – 143, 2015, wearable Robotics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889014001729

[77] R. K. Katzschmann, A. D. Marchese, and D. Rus, *Hydraulic Autonomous Soft Robotic Fish for 3D Swimming*. Cham: Springer International Publishing, 2016, pp. 405–420. [Online]. Available: https://doi.org/10.1007/978-3-319-23778-7_27

[78] B. Mosadegh, P. Polygerinos, C. Keplinger, S. Wennstedt, R. F. Shepherd, U. Gupta, J. Shim, K. Bertoldi, C. J. Walsh, and G. M. Whitesides, "Pneumatic networks for soft robotics that actuate rapidly," *Advanced functional materials*, vol. 24, no. 15, pp. 2163–2170, 2014.

[79] M. T. Tolley, R. F. Shepherd, B. Mosadegh, K. C. Galloway, M. Wehner, M. Karpelson, R. J. Wood, and G. M. Whitesides, "A resilient, untethered soft robot," *Soft robotics*, vol. 1, no. 3, pp. 213–223, 2014.

[80] B. Tondu, S. Ippolito, J. Guiochet, and A. Daidie, "A seven-degrees-of-freedom robot-arm driven by pneumatic artificial muscles for humanoid robots," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 257–274, 2005.

[81] M. Wehner, R. L. Truby, D. J. Fitzgerald, B. Mosadegh, G. M. Whitesides, J. A. Lewis, and R. J. Wood, "An integrated design and fabrication strategy for entirely soft, autonomous robots," *Nature*, vol. 536, no. 7617, p. 451, 2016.

[82] L. Yao, R. Niiyama, J. Ou, S. Follmer, C. Della Silva, and H. Ishii, "Pneui: pneumatically actuated soft composite materials for shape changing interfaces," in *Proceedings*

of the 26th annual ACM symposium on User interface software and Technology. ACM, 2013, pp. 13–22.

[83] Y. Zhang and A. Nishi, "Low-pressure air motor for wall-climbing robot actuation," *Mechatronics*, vol. 13, no. 4, pp. 377–392, 2003.

[84] R. K. Katzschmann, "Building and controlling fluidically actuated soft robots: from open loop to model-based control," Ph.D. dissertation, Massachusetts Institute of Technology, 2018.

[85] T. Castle, Y. Cho, X. Gong, E. Jung, D. M. Sussman, S. Yang, and R. D. Kamien, "Making the cut: Lattice kirigami rules," *Physical review letters*, vol. 113, no. 24, p. 245502, 2014.

[86] A. Rafsanjani and K. Bertoldi, "Buckling-induced kirigami," *Physical review letters*, vol. 118, no. 8, p. 084301, 2017.

[87] S. Mueller, B. Kruck, and P. Baudisch, "Laser origami: Laser-cutting 3d objects," *Interactions*, vol. 21, no. 2, p. 36–41, 2014. [Online]. Available: https://doi-org.ezproxy.library.ubc.ca/10.1145/2567782

[88] Z. Taylor, *Wood benders handbook.* New York: Sterling, 2008.

[89] D. Mitov, B. Tepavčević, V. Stojaković, and I. Bajšanski, "Kerf bending strategy for thick planar sheet materials," *Nexus Network Journal*, vol. 21, no. 1, pp. 149–160, 2019.

[90] O. Z. Güzelci, S. Alaçam, Z. Bacınoğlu *et al.*, "Enhancing flexibility of 2d planar materials by applying cut patterns for hands on study models," *Congress of the Iberoamerican Society of Digital Graphics*, vol. XX, no. 1, pp. 1–7, 2016.

[91] S. Zarrinmehr, E. Akleman, M. Ettehad, N. Kalantar, and A. Borhani, "Kerfing with generalized 2d meander-patterns: conversion of planar rigid panels into locally-flexible panels with stiffness control," in *Future Trajectories of Computation in Design - 17th International Conference.* Istanbul, Turkey: CUMINCAD, 2017, pp. 276–293.

[92] M. Capone and E. Lanzara, "Kerf bending: ruled double curved surfaces manufacturing," *XXII CONGRESSO INTERNACIONAL DA SOCIEDADE IBEROAMERICANA DE GRÁFICA DIGITAL*, vol. XXII, no. 1, pp. 1–8, 2018.

[93] M. Wei and K. Singh, "Bend-a-rule: A fabrication-based workflow for 3d planar contour acquisition," in *Proceedings of the 1st Annual ACM Symposium on Computational Fabrication*, ser. SCF '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3083157.3083164

[94] B. Taherkhani, M. B. Azizkhani, J. Kadkhodapour, A. P. Anaraki, and S. Rastgordani, "Highly sensitive, piezoresistive, silicone/carbon fiber-based auxetic sensor for low strain values," *Sensors and Actuators A: Physical*, p. 111939, 2020.

[95] *It's Alive! Exploring the Concept of a Living Phone.* Regina CAN: CRC Press, 2013.

[96] "50000 makeblock co. ltd air pump motor." [Online]. Available: https://www.digikey.com/product-detail/en/makeblock-co-ltd/50000/1890-1035-ND/7916853

[97] Blorgggg and Instructables, "Silc circuits: High performance conductive silicone," Apr 2018. [Online]. Available: https://www.instructables.com/id/Silc-Circuits-High-Performance-Conductive-Silicone/

[98] "Ecoflex™ series, super-soft, addition cure silicone rubbers." [Online]. Available: https://www.smooth-on.com/product-line/ecoflex/

[99] "Capacitivesensor libaray." [Online]. Available: https://playground.arduino.cc/Main/CapacitiveSensor/

[100] "Super nintendo entertainment system controller." [Online]. Available: https://store.nintendo.com/super-nintendo-entertainment-system-controller.html

[101] A. Dominec, *Export Paper Model from Blender*, GitHub, 2020. [Online]. Available: http://addam.github.io/Export-Paper-Model-from-Blender/

[102] T. T. Read, "The early casting of iron: A stage in iron age civilization," 1934.

[103] D. Miller, "Smelter and smith: Iron age metal fabrication technology in southern africa," *Journal of Archaeological Science*, vol. 29, no. 10, pp. 1083 – 1131, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305440301907587

[104] B. Foundation, "Home of the blender project - free and open 3d creation software." [Online]. Available: https://www.blender.org/