

October 4, 2017
DRAFT

Making Fabrication Real

Fabrication for Real Usage, with Real Objects, by Real People

Xiang ‘Anthony’ Chen

August 2016

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Scott Hudson, Co-Chair
Stelian Coros, Co-Chair

Jodi Forlizzi

Tovi Grossman

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Copyright © 2016 Xiang ‘Anthony’ Chen

October 4, 2017
DRAFT

Keywords: Fabrication, 3D printing, real world, design tools

October 4, 2017
DRAFT

October 4, 2017
DRAFT

Abstract

The increasingly personal and ubiquitous capabilities of computing—everything from smartphones to virtual reality—are enabling us to build a brave new world in the digital realm. Despite these advances in the virtual world, our ability as end-users to transform the physical world still remains limited.

The emergence of low-cost fabrication technology (most notably 3D printing) has brought us a dawn of making, promising to empower everyday users with the ability to fabricate physical objects of their own design. However, the technology itself is oblivious of the physical world—things are, in most cases, assumed to be printed from scratch in isolation from the real world objects they will be attached to and work with.

To bridge this ‘gulf of fabrication’, my thesis research focuses on developing fabrication techniques with design tool integration to enable users to expressively create designs that can be attached to and function with existing real-world objects. Specifically, my work explores techniques that leverage the 3D printing process to create attachments directly over, onto and around existing objects; a design tool further enables people to specify and generate adaptations that can be attached to and mechanically transform existing objects in user-customized ways; a user-driven approach allows people to express and iterate structures that are optimized to support existing objects; finally, a library of ‘embeddables’ demonstrate that existing objects can also augment 3D printed designs by embedding a large variety of material to realize different properties and functionalities.

Overall my thesis aspires to *make fabrication real*—enabling people to express, iterate and fabricate their designs that closely work with real-world objects to augment one another.

October 4, 2017
DRAFT

October 4, 2017
DRAFT

Acknowledgments

to be finished ...

October 4, 2017
DRAFT

Contents

1	Introduction and Overview	1
1.1	The Disparity of Creativity	1
1.2	The Dawn of Making	1
1.3	The Oblivion of the Real World	2
1.4	Research Goals	3
2	Background and Literature	5
2.1	3D Printing: History, Principles and Recent Development	5
2.1.1	Major Categories of 3D Printers	5
2.1.2	Issues and Limitations	6
2.1.3	Printing with Multiple Material	7
2.1.4	Cost and Customizability	8
2.2	Current Research Trends in 3D Printing	9
2.2.1	Optimizing the 3D Printing Process	9
2.2.2	Making 3D Printed Objects with Interactive Properties	10
2.2.3	Innovating Tools and Devices for Fabrication	11
2.3	3D Printing for Real-World Objects and Usages	12
2.3.1	Combining 3D Printing and Real World Objects	12
2.3.2	Specifying Real World Usage and Functional Requirement	14
2.4	Enabling People to Create Custom Designs	15
2.4.1	User-Driven Suggestive Interfaces	15
2.4.2	Supporting Hands-On Making Process	16
2.5	Summary of Literature Review	16
3	Techniques for Making Attachments to Real World Objects	17
3.1	Existing Attachment Solutions	19
3.2	Techniques For 3D Printed Attachment	19
3.2.1	Technique #1: Print-Over	20
3.2.2	Technique #2: Print-to-Affix	21
3.2.3	Technique #3: Print-through	21
3.3	Geometric Analysis: Viability	22
3.3.1	Understanding Viability: Collision Detection	22
3.3.2	Attaining Viability: Resolving Collision	24
3.4	Geometric Analysis: Durability	25

3.5	Geometric Analysis: Usability & Semantics	26
3.6	A Pipeline For Printed Attachments	27
3.6.1	Interactive Exploration	27
3.6.2	Model Generation, Printing and Post-processing	28
3.6.3	Software and Hardware Implementation	30
3.7	Evaluation	30
3.7.1	Evaluating Time and Material Cost	31
3.7.2	Evaluating Strength of Printed Attachments	32
3.8	Summary of Attachment Techniques	35
4	Adapting Real World Objects for Custom Users and Use Cases	37
4.1	Scenario Walkthrough: Adapting a Cutter	39
4.1.1	Iteration I: Generating Wrappers for a Better Grip	39
4.1.2	Iteration II: Making Levers to Assist with Clutching	40
4.1.3	Iteration III: Anchoring the Cutter for Situated Use	40
4.2	A Design Space of Adaptation Strategies	40
4.3	Reprise: Computational Design of Adaptations	43
4.3.1	Techniques for Specifying Types of Action with an Object	43
4.3.2	Computationally Generating Adaptations	44
4.3.3	Adjusting Design Parameters for Customization	45
4.3.4	Attaching Adaptations onto Real World Objects	46
4.3.5	Implementation, Measurement and Fabrication	48
4.4	Results and Validation Through Replication	50
4.5	Discussion and Summary of Designing Adaptations	52
5	User-Driven Structure Design to Support Real World Objects	57
5.1	Forte: User-Driven Generative Design	59
5.1.1	Basic Concepts and Terminologies	59
5.1.2	Sketching Designs and Loading Scenarios	60
5.1.3	Design Optimization with Global Similarity Control	61
5.1.4	Refining Optimization with Local Suggestive Edits	63
5.1.5	Informing Performance Trade-offs with Visualization	64
5.2	Implementation	64
5.2.1	Sketching Designs and Loading Scenarios	64
5.2.2	Design Optimizations with Global Similarity Control	64
5.2.3	Refining Optimization with Local Suggestive Edits	66
5.2.4	Informing Performance Trade-offs with Visualization	66
5.2.5	Software and Hardware	67
5.3	Design Sessions	67
5.3.1	Participants	67
5.3.2	Tasks & Procedure	67
5.3.3	Feedback and Discussion	68
5.3.4	Designs Created by Participants	69
5.4	Post-processing and Fabricated Results	69

5.5	Limitations, Trade-offs and Future Work	72
6	Embedding Everyday Objects to Augment 3D Prints	75
6.0.1	Contribution	77
6.1	Design Space	77
6.2	A Library of Embeddables → Rich Materiality	79
6.3	Overview of Design and Fabrication Process	83
6.4	Specifying the Geometry of Embeddables	84
6.5	Finding Optimal Paths to Insert Embeddables	87
6.5.1	Deformational Insertion of DOW=1 Embeddables	87
6.5.2	Non-deformational Insertion of DOW=0, 2, 3 Embeddables	88
6.6	Generating Guides for Crafting Embeddables	90
7	Summary and Timeline	93
	Bibliography	95

October 4, 2017
DRAFT

List of Figures

3.1	Examples of how existing objects can be augmented using our techniques: a) turning a battery into a LED torch; b) making a magnet from a Teddy bear; c) adding a stand to a glue gun and d) attaching a name tag to a pair of scissors.	18
3.2	A computational pipeline for designing and fabricating attachments to augment everyday objects.	18
3.3	a) The magnet holder printed directly on a teddy bear that was scaffolded on support structures; b) a handle added to an espresso cup; c) strapping to make a reusable 4 pack handle; and d) a bracelet printed through a heartshaped charm.	20
3.4	Example of a print-through process: the printer pauses at a point where the scissors can be dropped to interlock with the name tag, after which the print job resumes.	22
3.5	Print-through provides visual feedback (highlighting intersecting faces) to inform users of object-object collision that might not look obvious from certain viewing angle.	23
3.6	Interlocking objects are not always viable for printing: a) the torus cannot be inserted while the structure is being printed due to collision; b) a different orientation makes the print viable. (arrows indicate printing directions).	24
3.7	A heat map is an effective way to visualize the analysis results for adding a handle to a wrench: a) red areas indicate a handle cannot be printed over (viability); b) emphasizing durability shows preference for areas with small curvature; and c) emphasizing balance (usability) shows preference for areas near the center of mass (This assumes the forces applied have the same direction as the surface normals).	28
3.8	Directly drawing on the object is a simple way to specify the attachment area, such as drawing a stroke around a bottle (a, b) to indicate where to attachment a strap. (c) a heat map visualization around the selected cross section.	29
3.9	In the Teddy bear magnet example, a) a model is generated with a magnet holder, a connector, and the support structure; b) Tool path view in Repetier-Host: the extruder pauses and moves away for inserting the teddy bear (c).	29
3.10	To test print-over in comparison to printing in one piece, we used both techniques to print attachments on surfaces with varied curvatures and then exerting tangential pulling force.	33
3.11	Testing the strength of a 3D printed handle attachment (b) by clamping it on a table (c) and pulling using a scale to measure force (a).	33

3.12 Strength test results show print-over is strong enough to sustain stress as if they had been printed in one piece. Both our techniques suffer from an increase of surface curvature (a) and roughness (b); (c) shows that affixing with straps (150mm zip tie) is able to sustain over 2.5 kg of pulling, and reacts differently to curvature.	34
3.13 To test strapping in print-to-affix, we increased the curvature between the attachment and the attaching surface and exerted tangential pulling force.	35
 4.1 Our main contribution is the tool integration and a formalized design workflow for making 3D printable adaptations onto everyday objects. For example, an occupational therapist can use our tool to explore different strategies of adapting a wire cutter (a), such as creating a wrapper to soften the grip (b-e), adding two levers to assist with clutching (f-h), or replacing one lever with an anchor to situate the cutter on the work surface (i-m).	38
4.2 Existing examples of adapting everyday objects: a faucet extension (a), a guide for sharpening drill bit (b), and turning a spray-can to a spray-gun (c).	39
4.3 A design space of adaptations summarized from over 3000 existing examples: five major adaptation strategies support various types of actions, from static grasp and hold, to linear and rotational motion that involves one or multiple parts of the objects.	41
4.4 The design workflow of Reprise. Start with specifying the types of <i>action</i> applied on an object, which serves as input for generating user-selected <i>adaptations</i> . The initial design can be further customized by <i>adjusting</i> a set of parameters. Finally, extra fasteners can be added to make adaptations more <i>attachable</i> onto the object.	42
4.5 Reprise shows a virtual hand to let the user specify how an object is grasped, such as forming a cylindrical grasp on a knife (a), or a spherical grasp on a bottle (b).	43
4.6 Reprise uses a spherical control for specifying pushing/pulling an object, such as pressing a button on a remote control (a), or pulling a zipper (b)	44
4.7 To specify rotating an object, Reprise lets a user select where the object is held (a), on which plane it is rotated (b), and the fulcrum of rotation (c). The red arrow shows the rotation arm along which a lever can be generated (d).	44
4.8 To specify clutching, the user selects the two parts of the object that are being clutched and then the fulcrum. The red arrows indicate the directions of the clutching arms (a). The user can generate a lever for clutching the spray bottle, while holding its body in hand (b).	45
4.9 Reprise lets the user specify moving one object towards or away from another, such as putting a key into a lock hole.	45
4.10 Reprise can generate a wrapper—a cylindrical structure bounding part of an object (a), or an extension—an extrusion from a selected area of the object (b).	46
4.11 Besides using sliders, users can also click a button to generate a large set of design variations (ab), from which they can select one that best matches the design they have in mind, and continue to develop it from that model (c).	47
4.12 Reprise provides a simple technique to split an adaptation in halves so an object, such as this fork, can be put inside a wrapper (Figure 4.17a).	47

4.13 A pipe clamp, provided by Reprise, attaches the generated stand (a) to the object. Clicking on the object places the pipe (b). Another stroke cuts it open and adds a pair of clamps (cd).	48
4.14 Beams can further support the object for an adaptation. To generate one, simply select a point on the object and another point on the adaptation.	48
4.15 As the phone's screen is marked as inaccessible (red) by an adaptation, the generated handle avoids covering that area.	49
4.16 Reprise also provides an SVG file for making a measuring tool for adaptations that require fitting users' hand or fingers. This one shown above was laser cut.	49
4.17 Replicated wrapper/extension examples. Ninjaflex was used for the fork wrapper (a), zipper handle extension (b), and the bottle lid wrapper (c). Nylon-based soft material was used for the cutter.	51
4.18 Replicated handle examples.	51
4.19 Replicated lever examples.	52
4.20 Replicated anchor/stand examples.	53
4.21 Replicated guide examples.	53
4.22 Combining multiple adaptations: a handle (a), a guide (b) and an anchor (c) can be combined into one (super) adaptation (d).	54
5.1 Forte is a 2D design tool that takes a user-driven approach of generative design: a user sketches a reading chair inspired by the painting ‘The Great Wave off Kanagawa’ (a) with specified loading scenario (red arrow forces, blue ground); Forte then generates structures (with real-time feedback) to support the loads while resembling the user’s sketch (b), which can then be post-processed to create a 3D fabrication-ready model (cd).	57
5.2 Overview of Forte’s default tool bar: adjusting the amount of material for generating the structures (a); different layers for sketching and specifying the loading scenario (b); drawing and erasing tools (c); controlling how the generated results are similar to the original sketch (d); and selecting different design optimization methods (e).	58
5.3 From users’ sketch, Forte (right) generates lightweight structures to support loads, such as a robot leg (left) that needs to support the robot’s weight using a specified amount of material.	59
5.4 Forte’s three types of design optimization iteratively generate structures based on user’s sketch while providing them with real-time visual feedback akin to an animation (a-e), the result of which can be controlled by a global ‘similarity’ value (f-h).	60
5.5 Interaction to specify a load.	61
5.6 Lasso-selecting an area of clearance around the tip of the hook (no structure generated here).	61
5.7 Stress visualization informs performance trade-off between different similarity values (left), and the amount of material (right).	62

5.8	Users can suggestively erase part of a generated result (ab), which prompts the system to update the optimization addressing the intent of removing material; with this tool, users can also refine or smoothen a generated result (c).	62
5.9	Users can suggestively add material to a generated result (circled strokes), which prompts the system to re-run the optimization and attempt to add structures accordingly.	63
5.10	Designs created by 10 participants using Forte.	70
5.11	Fabricated examples designed from Forte’s generated structures: legs for a quadruped robot (a); a bike seat (b); an S-shaped chair (c); high heels (d); pet jumping platforms (e); a reading chair (f); and a tea table (g).	71
5.12	All our fabricated examples from the sketches in Forte, to generated 2D designs, and to post-processed 3D models; we performed a Finite Element Analysis on each 3D model based on the input loading scenario, which shows consistent stress distribution across a range of increasing loads.	73
6.1	We present a library of <i>embeddables</i> —everyday objects that can be cut, worked and embedded into 3D printable designs to augment their material properties. For example, inserting a florists’ wire makes elastic TPU-based objects stay in shape (plastic deformation) (a); a screw is used to reinforce a pegboard hook (b); a piece of sand paper becomes the surface of a printed sander (c); the malleability of wax makes it possible to further personalize assistive devices even after they are printed (d).	75
6.2	Everyday objects are made of various materials, e.g., for gloves, we have leather (a), silicone (b), nitrile (c), rubber (d) and even infused copper (e).	76
6.3	A design space characterizes the material properties of embeddables, as well as their <i>degree of workability</i> (DOW)—whether they are embedded as fixed-shape objects ($DOW=0$), or can be cut to length (1), to a fixed-thickness profile (2), or to a customized 3D form (3).	78
6.4	Inserting wire makes adjustable hinges for this pair of Harry Potter glasses.	80
6.5	Embedding higher density nuts adds weight to stabilize this printed stand for large smart phones.	80
6.6	Embedding sponge softens and tightens the grip of this printed wrench.	80
6.7	Embedding a piece rubber to make a functional door stop.	81
6.8	Embedding wire straightened from a paper clip into a spoon avoids snapping, which could be dangerous to the user during breakage. Please also see our accompanying video figure.	81
6.9	Embedding clear marbles into a lamp shade makes it more translucent, and can also refract light to brighten the upper part.	82
6.10	Embedding coins to laptop stands creates a heat sink.	82
6.11	Embedding scraps cut from a silicone oven mitt into a pair of tongs makes it possible to be used with hot objects.	82

6.12 Overview of Medley, the computational tool that supports the library of embeddables, from searching an embeddable by material properties (a), to interactively specifying its geometry and placement (b), to finding an optimal insertion path (c), and to generating fabrication-ready model (d) and instructions for the crafting process (e).	83
6.13 Specifying the geometry of a DOW=0 embeddable (e.g., a nut) by ‘drawing’ it on a design object (b), which aligns its pre-computed principal axis (a) with that of the sketched points’ (bc).	84
6.14 Top view of ‘pushing’ an embeddable to a design object: after sketching a to create the geometry (a), each sketch point raycasts along the reversed direction of its normal to find another control point (b), and by interpolating pairs of these points we can ‘push’ in an embeddable, which is driven by a depth value d between 0 and 1.	85
6.15 Specifying the geometry of a DOW=1 embeddable (e.g., a wire) by drawing a continuous line (ab) or clicking at discrete points (cd).	86
6.16 Specifying the geometry of a DOW=2, 3 embeddable by drawing an open freeform line (a), which is then expanded into a 3D shape with controllable thickness and width (b→e).	87
6.17 Specifying the geometry of a DOW=2, 3 embeddable by drawing an (automatically) closed line (a), which is then retessellated to a surface extrudable into a 3D shape (b→e).b	87
6.18 A 2x2 space of inserting embeddables that considers (i) whether insertion occurs during or after printing the design object; and (ii) whether the embeddables undergo deformation during insertion.	88
6.19 Finding an optimal insertion path for DOW=1 by extending each of the end points along its tangent vector, and along a circular path of radius equal to the bend radius of the material. We compute and search for the shortest amongst all possible paths.	89
6.20 Finding an optimal insertion path for DOW=2, 3: for a given insertion direction θ , we compute the minimum volume that needs to be removed in order to insert an embeddable; we then search step-wise across possible insertion directions to find one with the minimum to-remove volume.	89
6.21 Generating a cap to fill up empty space created by removing volume for the insertion path	90
6.22 Demonstration of inserting a cap during the printing process: an embeddable (a nut) is inserted (ab), and then a cap (c), after which the printing continues (d) and finishes (e).	90
6.23 For freeform custom embeddable geometry, we generate printable templates to assist the crafting process: in the case of making the soft wrench handle, we can use the printed templates to trace the embeddable’s shapes on a sponge before carving them out.	91

October 4, 2017
DRAFT

List of Tables

3.1	Prediction models for printing time of attachment techniques and data from a exemplar object+attachment.	31
3.2	Prediction models for material cost of attachment techniques and data from a exemplar object+attachment.	31

October 4, 2017
DRAFT

Chapter 1

Introduction and Overview

1.1 The Disparity of Creativity

As computers and the Internet become increasingly ubiquitous, people are now able to create or curate a variety of digital content. Today it only takes a few minutes to build and customize a personal website. Anyone can write a beautifully typeset blog without knowing anything about layout, font, or kerning. Writing and publishing apps has never been easier. You can even build your own virtual reality without making a step out of your dorm room¹, or catch an army of Pokemons as you go². The digital realm is such a wonderland where anyone, with a mouse, a keyboard or even just a touch screen, can become the greatest architect of their own virtual world.

In contrast to the prosperity of the cyberspace, people's ability to transform the real physical world has always been quite limited. In fact, since the second industrial revolution, people have been increasingly reliant on manufacturing and retail to procure what they want—the making and craftsmanship of individuals gradually became obsolete and increasingly irrelevant. More recently, the convenience of online shopping and low-cost, fast shipping further disincentivizes people from make things on their own.

Thus we are faced with a disparity of creativity: we are so freely and capable of building almost anything in the digital realm; we are so limited and much less capable of making things in the physical world.

1.2 The Dawn of Making

Fortunately, the advent of low-cost fabrication technology (most notably 3D printing) has given rise to a dawn of making. With 3D printing—a simple layer-by-layer additive manufacturing approach—people can simply import a 3D model, let the printer take care of the fabrication process and produce a physical artifact of their own design. Even without professional 3D design skills, people can go to online communities such as Thingiverse³ to find and share 3D mod-

¹Minecraft. <https://minecraft.net/>

²Pokemon Go. <http://www.pokemon.com/us/pokemon-video-games/pokemon-go/>

³Thingiverse. <http://www.thingiverse.com/>

els with one another. These communities have curated a large set of things a person would want to make—from reconstructed model of Roman sculpture, to a fully functional plastic bottle shredder. Meanwhile, despite the current material limitation in desktop printers, we are moving towards a future where the availability of various materials further broadens the scope of making. People can now print things that are soft⁴, sculptable⁵, chemical-resistant⁶, structurally strong⁷, magnetic⁸, metallic⁹, edible¹⁰, or self-actuating [48].

3D printing¹¹ has served as a platform that aggregates research and development from various domains to collectively enable people to transform their imagination and creativity into real objects that could impact their real lives. Indeed, such real world impact, whether it is small life hacking or a world-wide collaborative project [76], is what makes 3D printing stand out from other consumer software and electronics.

1.3 The Oblivion of the Real World

Given such fabrication capability, one natural approach to impact the real world is to use the printers to constantly produce new things—bespoke objects or people’s own creative designs. While this has been an important aspect of 3D printing, the sole focus of making new things is oblivious of a whole world of existing objects that we are already interacting with at a daily basis. Rather than always producing things from scratch, perhaps another important mission of 3D printing is to closely couple with and build upon real world objects. In this way I believe we can maximize our ‘degree of freedom’ in making things: people can choose to fabricate something entirely new, or to base their design on—and at times in combination with—existing things. As a result, we should expect a reciprocal relationship: 3D printable designs can augment existing manufactured things; and vice versa.

Besides existing objects, another important aspect that is often ignored in fabrication is people. It is true that 3D printing is bringing people back to making things. The kind of making it affords, however, is quite different from traditional practices. Traditional fabrication—everything from jewelry making to wood working—is fairly hands-on. People express and realize their design ideas through the manual making process. In contrast, CAD and digital fabrication post a fairly high bar to achieve such freedom of design; at times, an automatic process (e.g., topology optimization) would take over the task, leaving users almost as by-standers with limited control of how a design should evolve.

⁴NinjaFlex. <https://nunjatek.com/>

⁵Sculptable material by Adam Beane Industries. <http://www.adambeaneindustries.com/>

⁶FilaOne GRAY by Avante Technology. <http://www.avante-technology.com/>

⁷PC-Max by Polymaker. <http://www.polymaker.com/shop/polymaker-pc-max/>

⁸Polymagnets. <http://www.polymagnet.com/polymagnets/>

⁹Yibo3D. <http://www.yibo3d.com/>

¹⁰Edible six-pack holder by We Believers <http://www.webelievers.com/>

¹¹As I focus on 3D printing as the main fabrication technique, hereon in this thesis I used the terms ‘3D printing’ and ‘fabrication’ interchangeably

1.4 Research Goals

The goal of my thesis research, in brief, is to *Make Fabrication Real*—the primary goal is to enable a making process (executed by 3D printers) that is closely integrated with real world objects and usages, whereby 3D printing can be used to extend, adapt, and support existing objects and vice versa; the secondary goal is to enable designs to be expressed and driven by ‘real people’, as opposed to being constrained by traditional CAD tools’s limited expressiveness, or dominated by design automation that minimizes the role of users to merely pressing a few buttons to advance to process.

These goals are manifested in the following four projects, as described in the remainder of this thesis.

- **Chapter 2** sets the scene with a brief review on the background of 3D printing—its core technical components and principles, as well as some recent developments in both research and development.
- **Chapter 3** addresses the attachment problem, which is a fundamental requirement for using 3D printing to augment real world objects. Specifically, I present Encore—a design tool with a suite of techniques that allow a user, with a unmodified consumer grade 3D printer, to directly print attachments over, around or through existing objects.
- **Chapter 4** builds on the attachment techniques, and focuses on enabling the design tasks of making such attachments that work with existing objects. Specifically, I describe Reprise—a design tool that provides simple interaction techniques and computational geometry for specifying, generating, customizing and fitting adaptations onto existing objects.
- **Chapter 5:** While previous two chapters focus on extending or adapting real world objects, This project—Forte—contributes a design tool that takes a user-driven approach to guide users to compose and fabricate objects that support existing things. Specifically, users start with sketching a design from their intuition; the system in the background optimizes their initial design based on the structural requirements; then users can further tweak the design while the system provides on-demand feedback and suggestion for keeping it functional.
- **Chapter 6:** So far the thesis has demonstrated that 3D printing can extend, adapt and support existing objects. The last project explores the opposite of this relationship: can existing objects augment 3D printed designs as well? Specifically, I describe Medley—a computational library of ‘embeddables’ where the rich material variety of everyday objects can be used to augment 3D printed designs, overcoming the currently limited multi-material capability on most desktop printers.
- **Chapter 7** concludes this thesis by reflecting on my exploration of these projects, discussing issues and questions that arise, and suggesting new opportunities for future work.

October 4, 2017
DRAFT

Chapter 2

Background and Literature

To set the scene, I first present a brief review of 3D printing’s history, its working principles and some recent development of the technology. Next I shift the focus to academic research, where several trends of research have been capitalizing on the democratization of 3D printing, tackling problems related to optimizing the 3D printing process, making 3D printed objects more interactive, and innovating new types of fabrication tools or devices. As fabrication is a relatively new topic in Human-Computer Interaction (HCI), my hope is to draw together all this recent work to build up the backdrop of my thesis research. Last but not least, I focus on reviewing several papers that are specifically related to my research goals, which also prelude the following chapters where I describe my main results.¹

2.1 3D Printing: History, Principles and Recent Development

The first 3D printing concept was published in 1981 by Hideo Kodama² from Nagoya Municipal Industrial Research Institute in Japan. He proposed a type of photopolymer rapid prototyping system that could build up layer after layer, each of which represented a slice of a 3D model. However, Kodama’s idea did not seem to take off, as the first commercially available 3D printer only debuted 5 years later, which was a stereolithography printer by 3D Systems³.

2.1.1 Major Categories of 3D Printers

Today, the term ‘3D printing’ loosely refers to a class of *additive manufacturing* techniques[27]. In general, the concept of additive manufacturing encompasses two key aspects: (*i*) the deposition of material through an *extruder* (sometimes also called a print head); and (*ii*) the positioning of the extruder in 3D space. The combination of these two allow an object to be created following a bottom-up, layer by layer manner.

¹There is some other closely related work that will be reviewed inline with specific projects, as I feel the discussion of them is more appropriate within those contexts rather than in a standalone chapter.

²<http://www.avplastics.co.uk/3d-printing-history>

³<http://www.3dsystems.com/30-years-innovation>

In an **Fused Deposition Modeling** (FDM) printer, an extruder deposits material (often called ‘filament’) by melting and pushing it through a hot end. Each extrusion produces a thin line of material, which is then laid down on a print bed or some printed layers of an object. To position the deposition point, several motors drive the extruder and the print bed to move relatively to one another, allowing material to be laid down in pre-computed locations, forming (2D) geometric patterns. The printer will create a 2D layer at a given height, which represents a cross section of an object; then it moves vertically up to create the next layer above the current one. This process is repeated until a full 3D object is created from a stack of the 2D layers.

The **photopolymerization** approach, such as stereolithography (SLA) [105], uses UV-curable photopolymer resin as material. UV light selectively cures a layer to create the cross-section of the object, and then the platform is lowered and the next layer is laid down; this process is repeated until all layers of an object is produced.

Sintering-based techniques, such as selective laser sintering (SLS) [106], first lays down a layer of powder (e.g., metal) and then uses laser to selectively sinter part of the powder layer based on the geometric pattern of the cross-section . Similarly, next the platform is lowered and the next powder layer is laid down and sintered to continually build up an object.

Laminated object manufacturing (LOM) [104] is an approach that successively glues together layers of material (paper, plastic or metal laminates), each of which, similarly represents a cross-section of a 3D object. Once printed, the object might be ‘trapped’ inside the stacked layers (as volumetrically the object is only a subset of the collection of layers). Post processing might be needed to remove unwanted material and carve out the target object. Several recent projects have extended this approach, cutting and stacking fabric to create soft objects [32, 58].

PolyJet⁴—a patented technology by Stratasys—is similar to inkjet printing except that liquid polymer is jetted, which is then cured, usually using UV light. The process is similar to a combination of FDM and material jetting (and curing).

Note that the family of 3D printer is still growing and lots of innovation is taking place. For example, recent work by Harvard researchers can print metal structures in mid-air by annealing nanoparticles with laser [81]. Perhaps the most innovative part of this work is it no longer constrains the printing process to layer by layer; rather, wire structure is created following its geometry, position and orientation in 3D space.

2.1.2 Issues and Limitations

3D printers’ additive approach simplifies the making of objects with arbitrary shape by almost brute-forcedly treating them unexceptionally as a stack of layers. Such simplification also gives rise to several issues and limitations.

Speed is a universal issue amongst all types of 3D printers. Although the printing time varies from object to object, from printer to printer and from one setting to another, it is typical for a print job to take hours to finish. Fundamentally, the layerization of objects inevitably ignores their inherent geometric characteristics. Everything is sliced into a stack of layers, which at times is a suboptimal approach in terms of speed. Consider a long and thin vertical wire structure that contains very little material. A typical 3D printing process will treat the structure as a stack

⁴<http://www.stratasys.com/3d-printers/technologies/polyjet-technology>

of its small cross sections, rather than optimally printing along its length (or height), where a strand of material can continuously make up a large portion of the wire's volume and in total less extruder movements are required to produce material for the object. Another reason for limited speed has to do with the deposition of material. For example, SLA needs to wait for each layer to cure in order to continue to the next one, making the curing time of material a bottleneck for speed. However, this is also an opportunity for breakthrough—recently Carbon⁵, Prodways⁶, SLASH⁷ all claimed to have achieved much faster speed thanks to breakthrough in the chemistry of material that allows for shorter curing time.

Overhang Despite the universal simplicity of FDM, its layer-by-layer approach has a inherent limitation: each layer needs to ‘stand’ on the previous ones. One can imagine that if a layer is much larger than the ones below it, part of this layer might have to stand on thin air, which is called an ‘overhang’ situation. Overhang is not a universal issue—some 3D printing technology does not have this problem. For example, for SLA, SLS and LOM, the overhang structure is naturally supported by material that does not belong to the object but has not yet been removed during the printing time. FDM and PolyJet, however, does not have such ‘convenience’ and will require a type of ‘support structure’—sacrificial structures that serve to support overhanging parts, and then removed from the final print. Essentially, support structures are put in place, and fabricated simultaneously with the target object to artificially create no-overhang situations. What is problematic about adding and printing these support structure is that it costs extra time and material to ensure stable support, exacerbating the speed problem. Further, as they become integral parts of the original object, it is generally difficult to remove them post hoc, unless the printer uses a separate, easily removable material specifically for support structure.

Structural strength affects the quality and usage of the 3D printed objects. Material is perhaps the primary factor for structural quality. Beyond traditional plastic-based filament, innovation in material, such as carbon nanotube⁸ and polycarbonate-based filament⁹ has proven to provide, amongst other improvements, stronger structural properties. However, material is not the only variable in deciding a printed object’s strength. The layer by layer approach of FDM printers has caused a side effect of anisotropic strength—meaning the structural property of an object is dependent on the direction in which it is printed. Specifically, a printed object is generally much stronger within layers than between them, as adjacent layers are anisotropically adhered rather than isotropically integrated.

Speed, overhang and structural strength are the three major issues/limitations for many of today’s 3D printers. They also have aggregated a large body of research that seeks to solve or mitigate these problems, which I will review in the next section on 3D printing research.

2.1.3 Printing with Multiple Material

As many existing objects are made of more than one kind of material, it is a natural ‘push’ for 3D printers to provide multi-material printing capability. Some types of printers are amenable

⁵<http://carbon3d.com/>

⁶<http://www.prodways.com/en/>

⁷<https://www.kickstarter.com/projects/644653534/slash-the-next-level-of-affordable-prof>

⁸<http://www.3dxtech.com/>

⁹<http://www.polymaker.com/shop/polymaker-pc-max/>

to switching between different kinds of material. PolyJet, like inkjet printers, can select which color of material to use during the printing process. FDM, on the other hand, is not so flexible: to use the same extruder, the switching of material has to be done manually; alternatively, adding more extruders will inevitably increase the complexity of the mechanical design, which is not a scalable option. For LOM, although different layers can be arranged a priori, it only manages to change material from layer to layer, not within each layer. For SLS and SLA, even manually switching to a different kind of material is a cumbersome process, as it would require cleaning up the whole volume of current material and replenish it with new material.

2.1.4 Cost and Customizability

Even with simplified layer-wise working principles, the making of 3D printers was never a trivial task. Primarily, there have been two challenges: (*i*) precision—the thinner each layer could be the higher resolution the final object; (*ii*) material—the more material can be fed into a printer the more variety of objects users can make. The pursuit of high precision and special material has always been pushing the development of 3D printers.

Perhaps due to such a high technical bar (in both manufacturing and usage), 3D printers in their earlier years were never considered or envisioned as a piece of consumer-grade equipment. The expensiveness was another important factor. Indeed, according to 3D Printing Industry, not until 2007 did the market see the first 3D printer under \$10,000¹⁰. Even this price point was still way too expensive for consumer use. (In comparison, the first iPhone, which also debuted in 2007, was under \$600.)

Thus the dawn of making never took place in traditional 3D printing industry; in fact, many have attributed the democratization of 3D printing to some early do-it-yourself (DIY) projects that made the first steps of making affordable, (relatively) easy-to-build fabrication machines. For example, Fab@Home from Connell University is a kit available for general public to produce and assemble a multi-material 3D printer [102]. The RepRap project¹¹ was engendered by Adrian Bowyer from the University of Bath in early 2004. The RepRap printer, foremost, is a fully functional FDM printer; further, the name stands for **R**eplicating **R**apid-prototyper, meaning that this printer can be used to replicate new ones of its kind by producing constituent parts for assembly. This was a significant mission of this project, as it would provide a solution for making 3D printers widespread to a large group of people.

Later in around 2009, Makerbot became one of the earliest consumer 3D printer manufacturers. In just a few years' time, numerous small or large companies emerged and joined the wave of making 3D printing affordable and usable to non-professional, non-expert users. To date, the cost of a consumer 3D printer has dropped to less than \$300. These low-cost 3D printers can be found in people's garages and basements, in educational and research institutes, and in hardware and facility stores. In 2013, Gartner projected a growth of 95.4% in consumer and enterprise 3D printer shipments worldwide through 2017¹².

¹⁰<http://3dprintingindustry.com/3d-printing-basics-free-beginners-guide/history/>

¹¹<http://reprap.org/>

¹²<http://www.gartner.com/newsroom/id/3139118>

In academic research community, the availability of 3D printers not only makes it possible to bring 3D designs to reality, but also continuously inspires new ideas and problems, which I review in the following section.

2.2 Current Research Trends in 3D Printing

As mentioned above, 3D printing is a very specific type of fabrication technique with relatively simplified principles following a bottom-up, layer by layer approach. It currently stands at the intersection of several research areas, including mechanical engineering, material science, computational geometry, computer graphics, and HCI. My thesis research is primarily situated in the area of HCI, and partially related to computational geometry and computer graphics. The literature review in this section spans these three areas (primarily in HCI) with an attempt to weave prior work into a backdrop of my own research.

Within this scope, research on 3D printing by and large converges to the following three trends: *(i)* optimizing the 3D printing process, *(ii)* making 3D printed objects with custom properties, and *(iii)* innovating tools and devices for fabrication. While such categorization is by no means complete (nor are the trends mutually exclusive to one another), it serves well to organize a massively growing body of work that involves 3D printing.

2.2.1 Optimizing the 3D Printing Process

Although 3D printing can expressively fabricate objects of almost any shape, the process itself has some inherent limitations.

Speed is perhaps the most noticeable problem. To solve the speed problem, researchers have taken an approximation based approach—instead of following the default slow printing process, they try to modify and accelerate it at the cost of lowering the fidelity of the printed result.

For example, faBrickation uses Lego bricks to replace parts and components that would otherwise need to be 3D printed [54]. In this way, assembling existing Lego bricks makes it faster to approximate the target object than printing it from scratch. The accompanying design tool converts an existing 3D model into a ‘low-fi’ version consisting of Lego bricks and outputs instructions for assembly. As this approach sacrifices the fine resolution of the original model, it also allows users to specify parts or components that need to be precisely 3D printed: a combination of both yield a flexible way to accelerate the making of an object. Similarly, Platener introduces laser cutting—a much faster fabrication process—into 3D printing. The design tool looks for parts of an object that resemble planar structures, and uses laser cut ‘plates’ to replace them, thus reducing the extra time if otherwise using 3D printing [10].

Besides approximation, it is also possible to print objects incrementally. For objects with large volume, researchers have come up with techniques that build these objects around a (large) existing component. For example, RevoMaker is a modified 3D printer wherein an existing block of material can be clamped on, rotated, and new parts can be printed on each side of it to finish a complete object [26]. CofiFab presents a coarse to fine approach, which first produces a coarse approximation of an object as the base (often using faster methods such as laser cutting), and subsequently fabricates parts with finer-grained details onto it [83].

A third approach of acceleration is to redesign the printing process. WirePrint, for example, converts a 3D model into a ‘low-fi’ custom wire mesh structure amenable to be printed fast [53], especially when using printers with high Z-axis speed (e.g., the Delta printer¹³). Similar to how ‘low-poly’ tessellation saves space for storing 3D models, ‘low-fi’ printing saves time for making 3D objects. This approach was further developed in On-the-Fly Printing—a custom 3D printer that uses cooling to further accelerate the WirePrint approach and adds a rotation mechanism so that new ‘wires’ can be added or removed at arbitrary points [60]. This enables incremental on-the-fly printing that happens as users are virtually building or editing a 3D model, thus accelerating the process with parallelization.

The requirement of **support structure** is another concerning problem of 3D printing. In the research community, there have been several approaches focusing on solving the first problem of support structure. Specifically, the goal is to reduce the cost of extra time and material without making these structures less stable (i.e., the overhanging layers can still stand through the 3D printing process). Schmidt and Umetani proposed a top-down procedural approach to generate support structures, which results in a significant reduction in wasted time and material compared to using manufacturer-provided method [74]. The tree structure in Schmidt and Umetani’s work is further improved by Dumas et al., who propose a bridge structure to provide stronger and more stable support [23]. Bridging is a specific FDM technique wherein the printer extrudes material horizontally trying to connect two end points at the same level. Although this is also an overhanging situation, the bridge is usually able to stand because of the tension of the extruded material as well as the support at the two ends (similar to how a hammock can be tied to and hung between two structures and even support human weight). Dumas et al.’s work exploits this bridging capability to hierarchically build up steady scaffolding in a cost-efficient way.

Structural quality This unique problem of 3D printing has motivated several interesting projects, such as Umetani and Schmidt’s cross-sectional analysis [94] and Ulu et al.’s build orientation optimization [93], both of which attempt to find an optimal direction to print an object so as to maximize its strength (or minimize the effect of cross-layer anisotropy).

All this work above focuses on understanding the limitations, optimizing the printing process and has solved a number of interesting problems related to speed, overhang and structural quality. Next I move on from the printing process and look at the design of objects that gives them custom properties once printed.

2.2.2 Making 3D Printed Objects with Interactive Properties

One compelling power of 3D printing is the customization of objects—the ability to transform them into something with unique properties customized by the users.

One well-explored approach to achieve this is by embedding custom structure or components into a printed object. This approach can often result in more interactive 3D prints, as the embedding enables the objects to transmit various signals with input and output capabilities. For example, Printed Optics is a general technique that embeds optical elements into objects to serve as display or optoelectronic sensors that react to how people manipulate the objects [14, 107].

¹³Instead of moving the print head orthogonally, Delta printers apply the delta robot principle, which uses three arms to position the print head in 3D space.

InfraStruct expands this idea to the terahertz (or sub-millimeter) spectrum: through terahertz imaging information can be encoded into or read from the internal design of 3D printable objects [108]. Accoustumrments further pushes this approach with transmission of a near-ultrasound swept frequency signal through a custom-built embedded tube. Upon reception of the signal, it can characteristically describe users' interaction with a printed object. In particular, this approach seems more deployable to smart devices, as audio signals can be received by their microphones, and 3D printed enclosures can be easily attached to the devices [44]. Similarly, Lamello 3D prints objects with structures that produce distinct acoustic signals when interacted with [73], such as a slider whose thumb brushes through a comb-like structure and produces distinguishable sound corresponding to its position.

It is also possible to embed and analyze other types of signal. Vazquez et al. manipulate air pressure inside a printed object with custom material and internal structure, which can also afford a similar class of interaction [97]. Similarly, Schmitz et al. embed liquids into 3D Printed objects to sense tilting and motion [75]. Vasilevitsky et al. embed structures within 3D printed mechanical components that enable the sensing of their mechanical states through electronic signal, such as a voltage divider in a gear and a variable capacitor in a hinger [96].

All this work above has pursued a close coupling of an object's intrinsic geometry and the added structures or components that give rise to new—mostly interactive—properties. In Graphics, researchers also have explored other physical properties of printed objects, such as making imbalanced objects stand [64], or making static objects easily spinnable [5], both of which by optimizing their internal material distribution.

Other work also investigates a ‘looser’ way of making 3D printed objects interactive via instrumenting extra components, often as a post processing step. For example, Sauron instruments mirrors inside a 3D printed object, such as a game controller, and uses a light source and a camera to detect user interaction via light redirection, such as sensing the button presses on different locations of the controller [70]. Makers’ Marks further facilitates the prototyping process using sculptable material (e.g., clay) to create physical models with add-on mechanical or electronic components. A vision-based approach then extracts the location information of these add-ons, which subsequently inform their final installation on 3D printed objects [72]. Jones et al. explored a similar approach, but instead building the fiducial marks as part the sculpted model, which then allow the scanning software to identify widget types and locations, and create structure and space for mounting them [40].

2.2.3 Innovating Tools and Devices for Fabrication

The expressiveness (as well as limitations) of 3D printing has cultivated research on improving its process and diversifying the printed object’s properties and behaviors. Further, it also inspired innovation on new fabrication tools and devices that are based on, or at times go beyond the traditional FDM approach.

One fairly customizable component is the extruder. For example, the ‘Teddy bear printer’ is able to 3D print soft object with needle-felted yarn. While the architecture of the printer is largely based on a conventional FDM machine, the key innovation of a custom print head contributes to a vastly different material usage and property. Specifically, the custom print head is able to handle the extrusion of and the piercing and entangling with soft fibre, while leveraging the along-

axis movement to formulate the geometry of print [32]. The aforementioned xPrint project also innovates the extruder of a conventional FDM printer, allowing it to handle solution material, while also able to be plugged in with other modules, such as ventilation and mechanical stirs [98].

It is also possible to combine both subtractive and additive manufacturing into one single printer. Scotty is a realization of this concept: as an object is milled away, each layer is scanned; the digitalized model can then be encrypted, sent to another printer, and printed using the additive approach [55].

Another direction is to extend the degree of freedom (DOF) of a conventional printer. One example of this approach is Peng et al.'s On-the-Fly printing—by adding two rotational DOF to a printer, it is now possible to print an object incrementally from various points and orientations, rather than always follow a fixed direction [60].

Other researchers also attain to augment the ‘vision’ of a 3D printer. Since there is no feedback loop in a conventional printer, the machine has no knowledge or awareness of the object that is being printed, thus no ability to prevent errors or improve upon its current state. To overcome this weakness, MixFab provides a mixed reality environment for users to model their design while juxtaposing it with real objects [99]. To truly integrate ‘vision’ into the printing process, MultiFab employs cameras and machine vision techniques to enable the printer to self-calibrate, scan object or partial prints, and perform self-corrections [80].

2.3 3D Printing for Real-World Objects and Usages

Amongst all this plethora of recent work, there is a strong focus on innovating the printing process, material and machine design, which enables more efficient printing, a larger variety of printable objects, and different fabrication techniques to empower and realize a user’s creativity. While all this advancement supports an end-to-end pipeline of fabricating new things, it limitedly assumes that objects always have to be made from scratch. What about a world of things that already exist? Should they be discarded and replaced, or rather augmented and built upon? What usage and functional requirements should be considered when augmenting or in general dealing with real world objects? How to specify these requirements in the design process? Further, lots of existing work also worships a complete automation of the fabrication process where sometimes the user’s act of making is reduced to a few button presses. Does this limit people’s innate creativity by keeping them ‘out of the loop’? In this section I review several key existing projects that investigate these questions - specifically, how should 3D printing involve real objects and real people.

2.3.1 Combining 3D Printing and Real World Objects

By and large, existing work has explored two ways of combining 3D printing and real world objects: (*i*) making objects that are composed of both 3D printed parts and existing things as components, (*ii*) embedding real-world objects into 3D prints; and (*iii*) using 3D printing to augment real world objects.

Composing target objects with real world objects. Research that follows the first approach is perhaps motivated by the need to overcome the limited speed of 3D printing. If it is possible to substitute 3D printed parts with something that already exists, less of the object needs to be printed hence smaller amount of overall time.

The aforementioned faBrickation project takes a quite radical approach substituting the majority of an object with lego bricks while only keeping precision-sensitive components to be 3D printed [54]. For example, to make a headmounted phone case (for virtual reality applications), everything except the lens holder can be assembled using lego bricks. The design tool performs the conversion from the original geometry to lego-based composition and outputs instructions for assembly. It is obvious that lego bricks only provide fairly limited resolution; however, it manages to instantiate a rough object of a user’s design, which is amenable to the iterative design process. Another project by Luo et al. goes beyond prototyping and attains to make functional object with lego bricks. In particular, they apply a force-based analysis to achieve a layout of lego bricks that makes the target object stable even at a fairly large scale [47].

Laser cut pieces can also be used to substitute 3D printed parts, which is demonstrated in the aforementioned Platner project [10]. It is worth noticing that this method is not limited to pure planar geometry: it is possible to laser cut curvy structure by, for example, using a special ‘S’ cutting patterns, or making the piece bendable with post hoc heating.

To generalize all this work, the essential idea is anything can become part of something we wish to fabricate, at least geometrically. 3D Collage demonstrates this idea by allowing artist to compose a target geometry with a collection of other objects (which are often semantically related to the target) [25]. RealFusion provides a 3D reconstructing environment wherein a user can simply bring in ready-made objects, digitalize and use them to model a target object [62]. To push this essential idea further towards fabrication, BottlePrint offers a design tool for using empty plastic bottles to approximate 3D objects, which allows making things at much larger scale beyond desktop printers. Virtually, they employ a tetrahedron-based tesselation scheme making target objects amenable to be made of by putting together and connecting bottles. Physically, a custom bottle connector is 3D printed, which is also numbered to ease the assembly process [42].

Embedding Elements into 3D Printed Objects Various elements of different material properties have been embedded into fabricated objects. Printed Optics embeds elements to customize the optical properties of an object [107]. Acoustroments embeds tunnels in printed device casings to enhance interactivity by sensing acoustic signals from user input [44]. InfraStructs uses multiple materials and AirCode uses air pockets to embed encoded information in a printed object [45, 108]. PipeDream embeds conductive material routed through the interior of an object to enable various kinds of interactivity over the surface [71]. Liquido can capacitively sense tilting and motion of printed objects, which is enabled by a fabrication platform that can embed liquids during the printing process [75]. Rather than static material, HotFlex proposes an approach to dynamically modify the state (shape) of 3D printed objects after they are printed, which is achieved by embedding low melting point material whose stiffness can be controlled by computationally generating heat from embedded resistors [30].

Using 3D printing to augment real world objects. The previous group of work tends to treat everyday objects as building blocks deprived of their existing functions but serving to be components of something else. Another possibility to deal with these objects is to build upon their existing functionality and to augment or customize it with 3D printed parts. In fact, the need

to augment everyday objects predated the dawn of personal fabrication. For example, Davidoff et al. propose ‘mechanical hijacking’—using motors that are designed to actuate existing controls or physical interface in specific ways [21]. However, these ‘hijacking’ devices have to be manually designed. RetroFab, on the other hand, provides a design environment for end users to scan existing physical interfaces, design new add-ons to modify their controls, and fabricate and install them so as to automate or optimize the usage with these devices or appliances [66]. RetroFab’s augmentation is generally heavyweight and aim for automating physical interfaces; in contrast, Project Reprise (detailed in Chapter 4) focuses on exploring a wide range of lightweight adaptations onto everyday things and hand tools to support custom usage or for users with special needs [18]. What Reprise represents is a class of design tools that go beyond the specification of geometry alone to provide very application domain specific knowledge and features related to real world objects. As a result, the printed adaptation serves for specific real world functions in tandem with existing objects.

One fundamental issue of 3D printed augmentations is how to attach them to existing objects. A standard approach is to use fasteners or adhesives - using fasteners would require attachments to be designed with fasteners-specific structures, such as bolt holes; using adhesives is more universal and accessible but also takes non-trivial effort and expertise to attain a strong adhesion. With 3D printing, we can explore other attachment techniques by leveraging the customizable geometry and printing process. Project Encore (detailed in Chapter 3) was conducted to experiment with such 3D printing based attachment techniques. Around the same time, Teibrich et al. explored a similar idea of ‘patching physical object’. Their printer—augmented with milling bit and 5-dof movement—is able to mill out parts of an existing object so that new components can be readily printed on it [87].

2.3.2 Specifying Real World Usage and Functional Requirement

Functional specification is a very essential step in professional industrial design and manufacturing.

For non-expert users, however, it could be difficult to understand or incorporate such information into the modeling process. To overcome this difficulty, prior work has explored guided design of functional objects. SketchChair allows a user to sketch a chair by drawing its cross section on a 2D canvas then ‘extruding’ it to a full 3D model [69]. Further, their tool can test the chair using a virtual human model and physical simulation, so that users are aware of any potential problem of their design [69]. However, there is little support for informing the users how they could correct or improve their design. To bridge this gap, Umetani et al. propose a design environment that, during the geometric editing process, also continuously visualizes the valid range of the design parameters [95]. Specifically, *feedback* is provided to the users once a constraint is violated, while *suggestions* guide them to transform the problematic design to a valid one. Although the system provides useful and executable guidelines, the design tasks seem to be fairly limited, primarily focusing on arranging a set of rectangular planks. Martinez et al. enables more freedom of expressing a visual pattern by providing a user-defined template [49]. By feeding this template into an optimization pipeline, the system can achieve aesthetically pleasing design while staying close to a strong enough structure. Although these patterns define a user’s desired appearance of the object, they remain as microscopic features; there is

little support for users to design the macro geometry of the object other than indirectly providing functional constraints to the optimization pipeline.

Outside of academic research, there is also a growing interest in industries to support end-user friendly functional optimization of their design. For example, Project DreamCatcher¹⁴ from Autodesk aims to generate a large number of design options based on users' high-level functional goals. It applies a generative design method to create design driven by specific "functional requirement, material type, manufacturability, performance criteria, and cost restrictions". In general, however, functional aspects of 3D design is yet to be supported in existing end-user oriented modeling tools.

2.4 Enabling People to Create Custom Designs

We have shown that researchers are trying to change the fact that 3D printing by default is a process detached to a real world of objects. However, few have realized that even more detached is the involvement of real people. Although it looks like a dawn of fabrication is bringing 'making' back to people, there still exists huge gaps between people's ideas and design tools' ability to let them express and iterate such ideas.

Prior work has demonstrated two approaches that empower people to create custom designs: the first approach is to have the system provide design suggestions based on preliminary user input, thus advancing users' design ideas without forcing them to take care of each small design step; the second approach is to support manual making process integrated into the digital fabrication pipeline whereby users can customize and realize their design ideas by hand.

2.4.1 User-Driven Suggestive Interfaces

In a broader context, user-driven suggestive interfaces date back to research in mixed-initiative interaction—an approach of designing intelligent systems that allows users and systems (e.g., an interface agent) to complementarily take on different roles in a task, enabling an optimal collaboration with each being able to "do what they do best" [24]. For example, TRAINS-95 is a dialog-based planning system where a railway manager describes the high-level routing goals while the system detects potential problems (e.g., traffic, weather) [24]. Horvitz summarizes principles of building such mixed-initiative user interfaces, as illustrated in the LookOut system that suggestively prompts and helps users to schedule events from their emails [31]. More recently, mixed-initiative interaction has been demonstrated in tools that help analysts 'wrangle' data into processable format by proactively suggesting plausible transform operation based on user input [95].

In the domain of design tools, user-driven suggestive interfaces allow users to take the lead and drive the design task, while the system observes, analyzes and suggests problems or improvements to enhance the design. For example, Chateau is a 3D sketching tool that predicts a user's next drawings or provide suggestions to complete drawings by observing user input [35]. Tsang et al. use images similar to a target design to guide the creation of 3D curves, and suggest

¹⁴<https://autodeskresearch.com/projects/dreamcatcher>

relevant geometry based on users' input strokes [92]. Umetani et al. present a tool for guided furniture design: as users edit their design, the tool indicates structures that are non-durable and/or non-stable, and further offers suggestions for solving these problem [95].

2.4.2 Supporting Hands-On Making Process

Fortunately, we have seen a few projects attempting to give people more (and easier) control of the design process. For example, WrapIt enhance hand-made wrapped jewelry by providing 3D printed jigs based on digital design, which then guide designers to wrap wires into custom shapes to realize the design [34]. ProxyPrint further explores how 3D printed artifacts can serve not as the target objects themselves, but intermediate tools for users to handmake those objects [91].

Researchers also developed novel hardware for promoting users' participation into the making process. D-coil provides an actuatable workbench as well as a hand-held wax extruder, together leading the user to extrude wax that formulates predefined geometry [59]. Protopiper is hand-held device that extrudes plastic pipes as building blocks for users to prototype objects at room-scale [3]. Reform integrates many of the aforementioned elements—it combines both additive and subtractive manufacturing with built-in scanning capabilities. This allows a collaborative relationship between humans and machines: users initiate a design with intuitive hand-modeling of soft material, the printer then digitizes it and performs machine-based modification as the user iterates on the digital model [100].

2.5 Summary of Literature Review

In this chapter I lay out the historical context of 3D printing, briefly introduce its working principles and recent development into consumer-grade low-cost devices. This has fostered a new area of fabrication research spanning HCI, graphics, mechanical engineering and a few other domains. As the focus of my thesis is mostly in HCI (with some overlaps with geometry and graphics), I then sample a range of recent work in these areas as a way to elicit my particular interest in this thesis, which is to integrate 3D printing with real world objects and to enable people to express their custom designs. The next few chapters unfold a presentation of four projects that encompasses these two themes, starting with techniques for making 3D printed attachment directly onto, around and through real world objects.

Chapter 3

Techniques for Making Attachments to Real World Objects

Current 3D printing process has the problem of being detached from real world objects, primarily because existing use of 3D printing technology primarily focuses on creating new objects from scratch. This oversimplified assumption precludes the possibility that existing objects can also be extended, repaired or more generally modified to fulfill additional purposes. Being able to augment these existing objects rather than remaking them presents a more sustainable option for fabrication and, further, avoids unnecessarily replacing objects that might be of personal value to the users.

To explore the possibility of augmenting real world objects, one fundamental ‘prerequisite’ is the ability to attach new 3D printed parts or components to existing objects. Fabricating parts to augment existing objects would require amongst other things an understanding of what attachment techniques can be used to join one object to another. There are a myriad of ways to fasten or attach two objects together from anchor bolts to zippers (see for example: [8, 103]). In our everyday life, this is often achieved using adhesives or fasteners. Adhesives are easily available and easy to use, however to achieve certain reliability one needs to know what kind of adhesive to choose based on the surfaces it is applied to, and further know and apply proper procedures (such as waiting an appropriate amount of time for full curing). Fasteners are mostly mechanical and often require special structures (e.g., bolt holes, flat surface to be clamped on) within an object to achieve robust fastening. Thus both options are can provide general solutions, but may be difficult to adapt to individual objects’ specific geometric and material properties. In particular, objects’ geometry might affect the viability (is a certain area attachable?), durability (is the attachment durable?), and usability/semantics (how would an attachment support or hinder the usage of the object?). Overall, how can we leverage the power of 3D printing to address these issues of attaching augmentations to existing objects?

In this chapter, I explore three specific attachment techniques enabled by a unified computational framework: *(i)* Print-Over prints attachments directly onto the surface of an object, such as printing a magnet holder to a Teddy bear (Figure 3.1b); *(ii)* Print-to-Affix fabricates parts which are separately attached using straps or adhesive, such as adding a structure to make a glue gun stand (Figure 3.1c); *(iii)* Print-Through prints an attachment through and around a hole of an existing object to interlock with it, such as printing a label into a the handle of a pair of scissors

(Figure 3.1d).

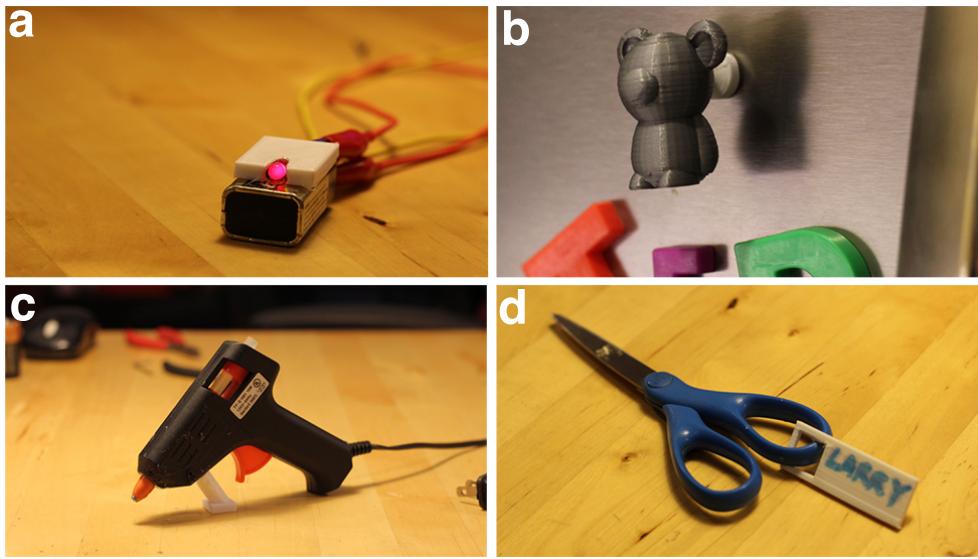


Figure 3.1: Examples of how existing objects can be augmented using our techniques: a) turning a battery into a LED torch; b) making a magnet from a Teddy bear; c) adding a stand to a glue gun and d) attaching a name tag to a pair of scissors.

These attachment techniques were integrated into Encore—a tool that can import a 3D model of an existing object, perform geometric analyses for the attachment techniques, and allow user exploration through visualization and direct manipulation. Our analysis metrics are designed to provide up-front information about viability, durability and usability of the attachments, where the user can explore tradeoffs among these metrics. Finally, Encore will generate a production-ready model of the parts to be printed, including information necessary for the custom printing process. Figure 3.1 shows an overview of the underlying framework.

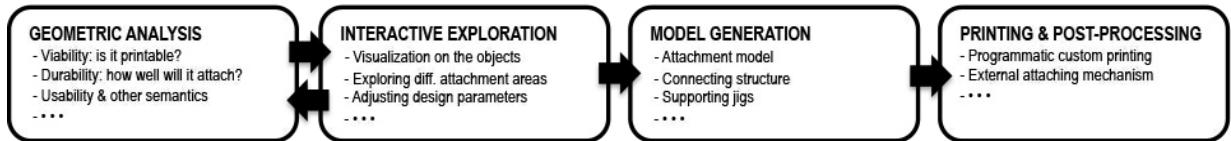


Figure 3.2: A computational pipeline for designing and fabricating attachments to augment everyday objects.

In the remainder of the chapter, I first review existing attachment solutions and how they might afford end-user customization. Then I present the attachment techniques, describe several exemplar analysis metrics for evaluating the goodness of attachment, and present Encore’s visualization and direct manipulation interface. Finally I report an evaluation of printing cost and strength.

3.1 Existing Attachment Solutions

Attachment—fastening or otherwise binding two objects together—is a task that commonly arises in everyday life, and one that is not entirely straightforward. Material properties, strength, usability, and aesthetics all need to be considered when attaching objects together. To address this, websites like ThisToThat¹ let a person enter two materials and suggests the best option for gluing them together. Other approaches leverage objects’ mechanical properties, such as the widely used joineries and fasteners commonly used in various manufacture industries (e.g., [8]). Attachment can also be made complex by the specific constraints of the objects being connected. For example, the free universal construction kit² offers adaptors between 10 otherwise incompatible children’s construction kits from Lego to Tinker toys.

In this work, we are primarily interested in attaching 3D printed components to existing objects. We assume that the existing object is not designed with attachment in mind, and is not to be directly modified (e.g., by drilling holes in order to use a bolt). Specifically, given an existing object and a new part we would like to fabricate and attach, we can either create a binding force between the new part and the old (such as using adhesives, or printing directly over a material that the filament will easily adhere to it) or in some cases we can loosely interlock the new part and the old (such as a buckling a strap through a handle or adding a charm to a charm bracelet or a key to a ring).

Choosing among these forms of attachment is a matter of understanding the task to be accomplished and the constraints that come with it. For example, if we wish to add a doggie bag holder to a dog leash, we may want it to be removable, but not moveable once attached (so it doesn’t flap around too much). It must be sturdy enough to survive many walks, dropped leashes, and so on, but does not need to carry much weight (just a roll of plastic bags). It may need to attach just below the handle of a cloth leash, or perhaps we are designing one that can attach to the plastic handle of a retractable leash. As another example, if we wish to add a handle to an espresso cup, it should withstand sheer forces based on the typical weight of a cup. These examples clearly demonstrate the wide range of issues that must be considered, and the interaction of properties of the existing object, task, and object to be attached.

To summarize, to determine the goodness of an attachment technique, viability, durability and usability are all important, although their relative importance may vary. Keeping these issues in mind, our focus in this project is on attachments to, around, or through an existing object without modification by leveraging the fabrication process and customizability of 3D printing. Below I showcase three exemplary attachment techniques.

3.2 Techniques For 3D Printed Attachment

Our techniques allow attachments to be printed directly onto an existing object (print-over), or separately and then adhered or strapped to it (print-to-affix), or through and around the object’s holes (print-through). Here we assume that the models of the existing and the new objects have

¹<http://www.thistothat.com/>

²<http://fffff.at/free-universal-construction-kit>

been acquired using 3D scanning, or created from scratch, and focus on methods for attaching the two together.

3.2.1 Technique #1: Print-Over

The first technique, print-over, is able to print an attachment directly onto the existing object. Once the attachment location is specified, the existing object is oriented and scaffolded with support structures so that it will not move while the attachment is printed on it. It is also important to ensure that the existing object will not impede the motion of the print head while the attachment is being printed.

Figure 3.1b shows a magnet holder directly printed over a Teddy bear toy (in this case also 3D printed) to make it a fridge magnet. As shown in Figure 3.3a, this was done by scaffolding the Teddy bear to the print bed so that the attachment area is facing upward and is accessible by the extruder, which then prints the magnet holder.

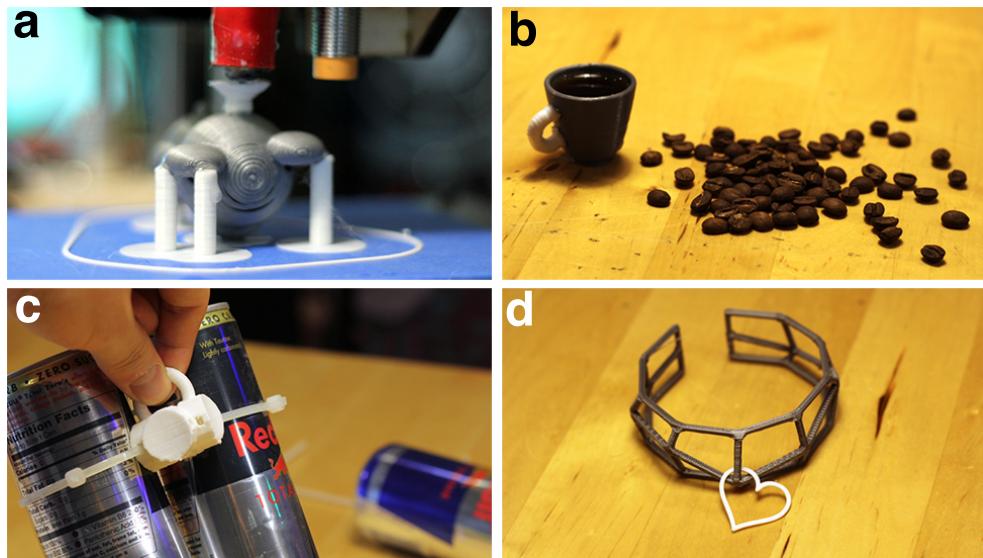


Figure 3.3: a) The magnet holder printed directly on a teddy bear that was scaffolded on support structures; b) a handle added to an espresso cup; c) strapping to make a reusable 4 pack handle; and d) a bracelet printed through a heartshaped charm.

Print-over works well when the attachment and the existing object are made of the same material, or materials that share similar thermodynamic properties. As detailed later, our evaluation shows that print-over is strong enough to sustain stress as if they had been printed in one piece. When objects are made of less compatible materials, we employ a work-around to perform print-over. Figure 3.1a shows an LED casing printed directly over a 9V battery to make a simple torch. This was done by adding a thin layer of glue on the battery prior to printing the casing. The glue simply creates a plastic-like layer that allows the printer's material (PLA) to stick to the battery while being printed.

3.2.2 Technique #2: Print-to-Affix

The second technique, print-to-affix, makes use of the concept of a connector that conforms to the surface geometry of the existing object and is snug-fit to the new object. This connector can be printed separately and then attached using external mechanisms (e.g., glue, or straps). This leverages the customizability of 3D printing to bridge objects and attachments that by default are likely to have unmatching surfacing. Figure 3.3c shows connectors used for the handle attachment to hold it tight with cans. Print-to-affix is a very general technique and can encompass for example the use of adhesive, straps, or even a custom printed part that snaps into place in some way. It also has the advantage of not preferring the existing object to be flat, as there is no concern about the print head colliding with the existing object.

Figure 1c shows a structure added to a glue gun to make it stand. This geometry was made to precisely match the attaching part on the glue gun, then 3D printed, and attached using adhesive. However, one disadvantage of using adhesive (or the aforementioned print-over) is the difficulty in detaching the attachment for redesign or reuse. Affixing with straps solves this problem. For example, using straps for the ‘4 pack’ holder in Figure 3.3c makes this handle reusable without either breaking it or the cans it holds.

Print-to-affix includes a wide variety of options. Details such as whether adhesives or straps are used significantly change the properties of the resulting attachment. For example, in one experiment, a 150mm zip tie could sustain up to a 4kg pulling force before the test object slipped, while the particular adhesive we applied tended to break at a less than 1kg force. On the other hand, using adhesives is less noticeable from an aesthetic standpoint.

3.2.3 Technique #3: Print-through

The third technique, print through, leverages the structural holes in some existing objects (e.g., keys and rings) to print the attachment through and around it. To accomplish this, the attachment is partially printed, the existing object is placed, and then printing continues until the two objects are interlocked and the print is complete. This requires determining whether there is a viable point at which to stop the print, and whether once the existing object is placed it will interfere with the print head.

Figure 3.1d shows a name tag printed through the handle of a pair of scissors. As shown in Figure 3.4, this was done by programmatically pausing the printer at a point where the scissors can be placed down and out of the way of the print head. Print-through has aesthetic qualities that distinguish it from print-to-affix and print-over: it typically creates a loose but permanent connection between two objects. Figure 3.3d is another such example whereby a bracelet was printed through a charm.

As highlighted in the techniques just presented, a range of tradeoffs must be considered when creating 3D printed attachments to existing objects. To address this, we present a series of analysis, based on the geometric properties calculated over the triangular meshes representing the existing object and the attachment. These analyses represent a sample of metrics that pertain to viability, durability, and usability. However, it is our intent and expectation that this set would be expanded over time and as new attachment techniques are explored.

- *Viability* indicates whether it is possible for a new part to be fabricated and attached at

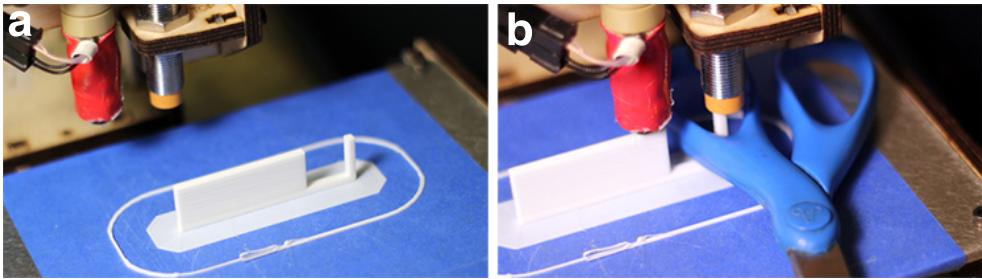


Figure 3.4: Example of a print-through process: the printer pauses at a point where the scissors can be dropped to interlock with the name tag, after which the print job resumes.

a given location on an existing object. For example, collision with the extruder during a direct print over process would violate viability;

- *Durability* examines how the geometric properties of the contact area between the attachment and the object could potentially strengthen or weaken the bond between them;
- *Usability and other semantics* considers various issues related to the actual usage of the attachment, such as the forces we expect to be applied to it, its balance, and some technique-specific issues (such as the length of strap required for fastening an attachment).

The next section will describe the details of these analyses, which serve as the computational backend of Encore—an interactive tool that informs people how attachments can be added to existing objects and how well they will perform in terms of viability, durability, usability and other semantics.

3.3 Geometric Analysis: Viability

Viability can encompass issues such as appropriateness for a specific printer with regard to issues such as support or size, as well as feasibility of printing without collision when an existing object is present during printing. Our work focuses on this last issue, which arises during print-over and print-through. More specifically, when printing involves two objects, it is only viable if (*i*) the attachment can be printed without the extruder’s tool paths colliding with the existing object (extruder-object collision); (*ii*) the attachment and the existing objects also do not intersect and collide with each other for a given spatial configuration (object-object collision). As shown in this section, both types of collisions are affected by the position and orientation of the existing object and the attachment. Thus, we must determine a position and orientation of both objects where the attachment can be printed collision-free.

3.3.1 Understanding Viability: Collision Detection

The first step of viability analysis is to detect collisions that will prevent the attachment from being printed.

Detecting Extruder-Object Collision

Extruder-object collision occurs when the placement of the existing object impedes the printer’s extruder movement while the attachment is being printed. This type of collision problem is very common in subtractive manufacturing and is often solved by careful tool path planning [19, 46], such as controlling the path of the cutting bit on a CNC router to avoid running into parts that are not to be removed. WirePrint has shown that collision can be avoided in non-layered additive manufacturing by modeling and considering the geometry of the extruder and the object [53]. In print-over, given an attachment Λ to an existing object Ω , we scan all Ω ’s vertices above the attaching point to detect whether they are within the range of the extruder’s movement, which can be obtained by computing the bounding radii of both the extruder and Λ .

Detecting Object-Object Collision

We consider and address two forms of object-object collision between an attachment and an existing object.

Direct Collision. In an interface for attachment placement, the mapping of the 3D scene onto a 2D screen could cause user errors in the placement of Λ with respect to Ω . A user might perceive Λ as interlocking with Ω , and yet the two in fact slightly intersect with each other, such that Λ is unable to be fabricated (Figure 3.5).

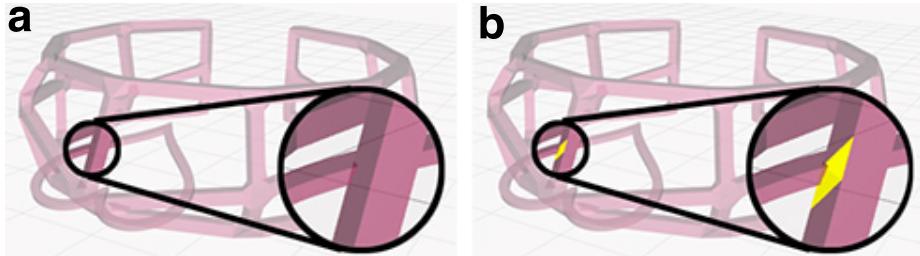


Figure 3.5: Print-through provides visual feedback (highlighting intersecting faces) to inform users of object-object collision that might not look obvious from certain viewing angle.

Such direct object-object collision is commonly dealt with using approximate models (e.g., checking axis-aligned bounding boxes for collision [9]). Such models could be problematic for attachments: for example, in print-through, Λ and Ω can be interlocking and collision free even though their bounding boxes do intersect with one another.

To address this, we compute whether two objects truly intersect by analyzing their meshes. To narrow down the detection scope, we compute both Λ and Ω ’s bounding spheres and locate a set of mutually bounded’ faces from Ω , denoted as F^Ω . Essentially, Λ and Ω are intersection free if and only if none of F^Ω ’s faces intersect with Λ ’s. By walking through a pre-computed octree of Λ [51], we can determine if a given face of F^Ω intersects with Λ . Further, as detailed later, we can also visualize the intersecting areas to inform the users how to reposition the attachment to an occlusion-free location (Figure 3.5).

Indirect Collision. When objects need to be placed around or through another object (e.g., a partial print), indirect collisions can occur. For example, viability is violated in print-through

if one object cannot be moved to its intended interlocked position without passing through' (i.e., intersecting) the body of the other object.

Figure 3.6 shows an artificial print-through example to explain this problem. In Figure 3.6a, the existing object (the torus) cannot be inserted into the larger structure (building layer by layer in the direction of the arrow) no matter when the printer is paused. Note that as illustrated in Figure 3.6b, indirect collision is orientation dependent there might exist an orientation that does provide a viable print for some pause point, different than the one specified by the user for analysis.

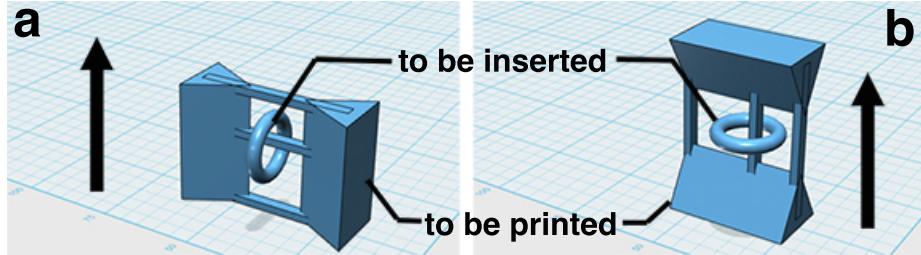


Figure 3.6: Interlocking objects are not always viable for printing: a) the torus cannot be inserted while the structure is being printed due to collision; b) a different orientation makes the print viable. (arrows indicate printing directions).

Inspired by Zhou et al.'s use of physics to unfold an object so as to test whether it can be folded [111], we make use of a reverse physics simulation. Specifically, we test whether there is a viable path for moving the existing object to its desired location (e.g., the scissors in Figure 3.4) partway through the print of the new object. We test such viability by reversing the insertion process. It starts with an object already inserted into the partial print, which is obtained by slicing the whole print so that its top layer is just above the inserted object. Then gravity is reversed. If the object 'escapes' under the reversed gravity, there exists a path for it to be inserted back from above and end up at the same placement where the test started.

In cases where this test fails (i.e., the object is trapped in the partial print), we can also perform a binary search to find a viable printing orientation, which we discuss next.

3.3.2 Attaining Viability: Resolving Collision

Once collision is detected, our analysis also searches for a solution by exploring alternate positions and orientations.

Resolving Extruder-Object Collision

Extruder-object collision is only an issue when printing an attachment Λ after an existing object Ω has been placed. Given a candidate surface area S of Ω on which Λ is to be printed, the first step is to rotate Ω so that S is relatively level and facing upward for extruder to print Λ on (e.g., the Teddy bear in Figure 3.3a). Generally S is not a perfectly flat surface to print on, so our next step raises Λ 's first printing layer P_0 to be above the entire S . However, after these two

operations, printing Λ might still run into object-extruder collision if parts of Ω are also above P_0 . To address this issue, we first print a connector on P_0 —a cylindrical structure that is small enough to be printed without collision. These connectors serve to continue raising the starting layer until Λ can be printed collision-free. One potential issue of this approach is the strength of these connectors in supporting Λ , which we also discuss later in the durability analysis section. Even with these solutions, some surface areas, such as those with a high convexity, might still be unprintable. As detailed later, our exploration phase (Figure 3.2) visualizes this information to guide the user in selecting alternate viable attachment areas.

Resolving Object-Object Collision

For object-object collision, a search for the position and orientation of Λ can determine whether there is a viable solution (in which the reversed gravity’ test succeeds). A simplified example of this situation is shown in Figure 3.6, where rotating Λ by 90 degrees resolves the issue.

Formally stated, the goal is to find a pair of rotations (α, γ) such that there exists a viable pause point for $R_x(\alpha)R_y(\gamma)\Omega$. (R_x and R_y are rotation matrices around x and y axis, respectively; rotating around z axis would not change the viability). The solution space is naturally continuous: given an pair of viable $(\hat{\alpha}, \hat{\gamma})$, there must also exist intervals $\alpha_l < \hat{\alpha} < \alpha_u$ and $\gamma_l < \hat{\gamma} < \gamma_u$ such that any printing direction in $\{(\alpha, \gamma) | \alpha_l < \alpha < \alpha_u, \gamma_l < \gamma < \gamma_u\}$ is also viable. Our search process is akin to a binary search of such intervals.

To sum up, the viability analyses described above detect collision issues that arise while fabricating the attachment either with the extruder or the existing objects. To prevent collision, we can orient the object or raise the starting layer. We also show that a collision-free printing orientation belongs to a continuous solution space and can be found via a binary search.

3.4 Geometric Analysis: Durability

Once a new part is viable for fabrication, the next question is its durability: how well it will attach to the existing object. One simple way to measure durability is to consider the contact area S between the existing object and the attachment; however our framework is extensible to other models and methods such as the cross-sectional analysis [94] and Finite Element Method [86]. Specifically, given a candidate contact area S , we consider the following metrics.

Size

Size can be approximated by summing up the area of each triangle in or intersecting with S . To generalize to multiple attachment sizes, we normalize the measurement by the volume V of the attachment. This metric serves as a simple way to capture an approximation of the structural strength of the connection point with respect to the attachment.

Flatness

Flatness can be computed by summing up the vertex-wise distance between S on the existing object and the printing layer at the bottom of the attachment. The flatness score is highest when

S is perfectly flat (coplanar with P); and lower as S becomes more irregular, meaning the triangles in S have highly variable heights and orientations, or S has an overall higher curvature.

Direction and Area of Force

The relationship between force and strength is affected by the type of attachment being used. For example, techniques such as print-over and using adhesives use interfacial bonds, with adherence to both the existing object and the attachment. In contrast, attachment techniques such as strapping create a force that holds the attachment onto the existing object, which can be measured as follows.

Consider an attachment surface S on the existing object for a strap, we compute S 's convex hull $\text{ConvHull}(S)$ using a Graham scan [28]. For each point p in S , we compute its shortest distance to $\text{ConvHull}(S)$: if the distance is smaller than a threshold we consider p to be making contact with a strap around S . The object is more strappable at S if most of S 's points make contact with the strap. A further consideration is contact point distribution: more evenly distributed contacts suggest a more balanced strapping force.

To sum up, durability examines how the geometric properties of the contact area between the attachment and the object could potentially strengthen or weaken the bond between them. Our analysis shows how contact area and size can be used to compute durability (most usefully for adhesion-based attachments) and explores the metric of strappability, which captures the contact area and force associated with a strap.

3.5 Geometric Analysis: Usability & Semantics

Even for a new part that is viable and durable, there are often further considerations related to the actual usage of the attachment. We provide two exemplar analysis of usability and semantics: (*i*) for handle-like attachments, we consider whether its attachment point creates a balance of the entire object when being gripped or held; and to illustrate a metric which is highly specific to one attachment technique (*ii*) an aesthetic strap length metric.

It is only natural that the analysis of usability and semantics would be highly specific to the type of attachment being used. For example, Prévost et al. discuss geometric modifications that can achieve balance such that an object will stand without falling [64]. Thus, the metrics we present are by no means exhaustive for the usability and semantic aspects of attachments; rather they suggest exemplar analysis that goes beyond the stage of fabrication and considers the effects and tradeoffs for an attachment in use.

Balance When Holding an Object By the Attachment

Balance when holding an object by a given attachment can be measured by moment, which indicates the tendency of the held object to rotate under its own weight. For example, as shown in the 4 pack' holder in Figure 3.3b, if the handle is rotated 90 degrees, it would be more difficult to hold the cans. When there are multiple contact points selected (such as the handle on an espresso cup), we simply sum up the moments of these points before normalization.

Technique Dependent Usability Analysis: Strap Length

Some attachment techniques may make use of very specific usability analysis. For example, the length of the strap is directly related to fastener cost and object appearance. In particular, we consider an analysis of strap length for a candidate attachment configuration normalized to a baseline (e.g., the bounding sphere of the existing object). The strap length can be computed from the aforementioned convex hull of the strapping area.

3.6 A Pipeline For Printed Attachments

We have presented a series of geometric analysis that can be used to quantify the goodness of various potential attachment options from the perspective of viability, durability, and usability. As shown in Figure 3.2, analysis results can be integrated into a pipeline for supporting iteration, interactive exploration, model generation, printing and post-processing. We illustrate these phases of the pipeline with our implemented tool, Encore.

3.6.1 Interactive Exploration

The interactive exploration phase creates a feedback loop wherein a user can explore different design parameters. Encore provides visualization and direct manipulation techniques to facilitate this process.

Visualization Techniques

Visualization provides effective feedback to the users to inform them about properties of their own design. When making attachments, one type of visualization is to compute an attachment score for each point on an existing object and overlay the results as we render the 3D model. Specifically, for each face on the object’s mesh, we locate its neighborhood area S , and pre-compute the viability, durability and usability analysis for this area. The results from different analyses can be weighted and combined based on user input and the purpose of the attachment.

Figure 3.7 shows Encore’s heat map visualization of these computed values, rendered on a wrench where the user would like to use print-over to add a handle (e.g., for hanging the wrench from a machine that needs frequent maintenance). The red areas indicate the parts of on the wrench where a handle cannot be printed over due to unavoidable occlusion (Figure 3.7a). The user can adjust the weights given to the metrics, such as choosing to emphasize durability. The visualization is interactively updated and shows preference (green) for areas with small curvature (Figure 3.7b). Alternatively, the user can emphasize balance, which narrows down the preferred areas to those near the center of mass (Figure 3.7c).

Direct Manipulation Techniques

Some attachment techniques, such as strapping, might require more user direction. For example, Figure 3.5 shows in the print-through technique, how the user can position the existing object to interlock with the attachment. When the two meshes intersect with one another, the intersecting

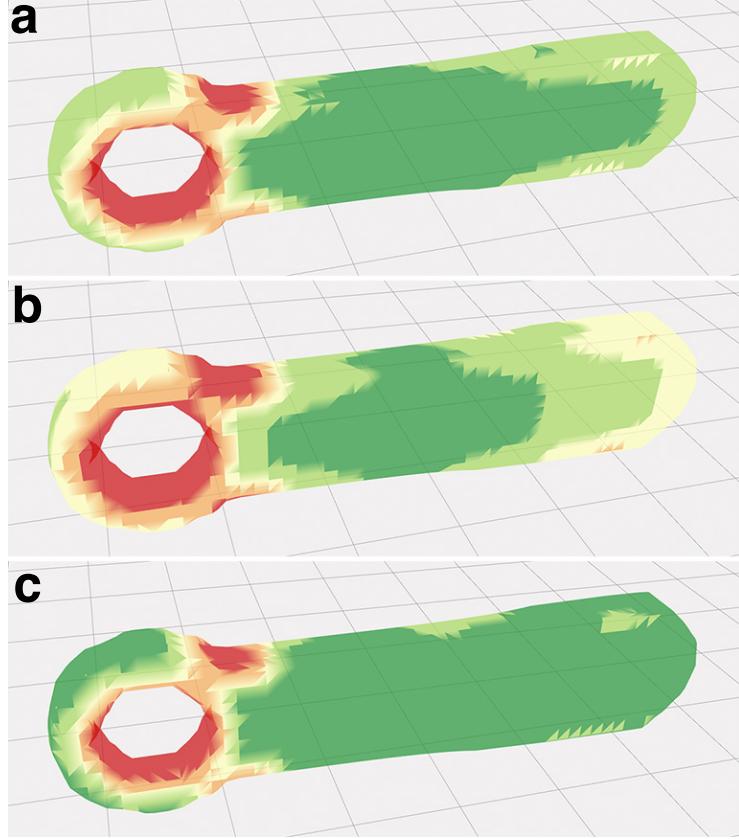


Figure 3.7: A heat map is an effective way to visualize the analysis results for adding a handle to a wrench: a) red areas indicate a handle cannot be printed over (viability); b) emphasizing durability shows preference for areas with small curvature; and c) emphasizing balance (usability) shows preference for areas near the center of mass (This assumes the forces applied have the same direction as the surface normals).

area is highlighted, which prompts the user to reposition the object to an intersection free location. Another useful technique is to allow the user to draw on the existing object to specify the attachment area. For example, for affixing using straps, the user simply draws a stroke around part of the object to indicate a strap. Based on this partial strap, we can find the corresponding cross section by first finding a plane that best fits the stroke points, and then intersecting the object with the plane. Figure 3.8 shows an example of drawing to specify where to strap an attachment on a bottle.

3.6.2 Model Generation, Printing and Post-processing

Once the user is satisfied with the attachment location, the pipeline continues to the last two phases: model generation, and printing and post-processing.

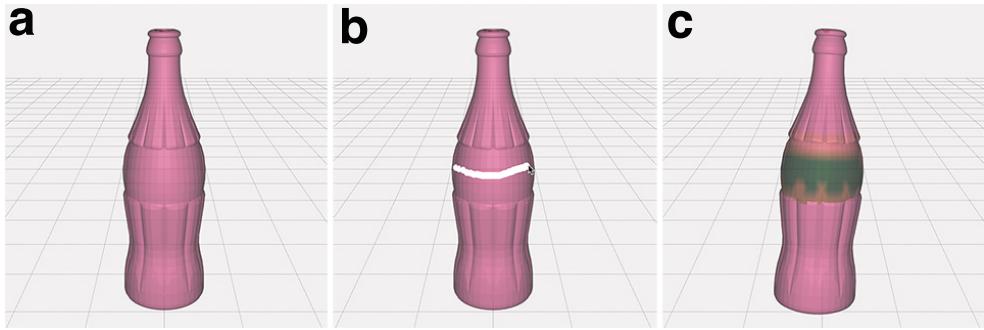


Figure 3.8: Directly drawing on the object is a simple way to specify the attachment area, such as drawing a stroke around a bottle (a, b) to indicate where to attachment a strap. (c) a heat map visualization around the selected cross section.

Model Generation

Model generation outputs a single file that includes the attachment, connector(s) and support structure. In particular, as discussed earlier, the connectors can raise the starting layer for printing the attachment to avoid potential occlusion issues; or be designed to snug-fit the existing objects (thus enabling affixing using adhesives or straps).

For some techniques, adding support can also solve occlusion problems. For example, in the print-through technique, when the attachment is much smaller than the existing object, it needs to be raised by a support structure so that the object can go through it while staying below the extruder. Other techniques use support to hold an object in place, such as direct print-over.

Figure 3.9a shows a connector and the support structure for printing over a magnet holder on a Teddy bear (the same one shown in Figure 3.3a). In this case, the five support cylinders at the bottom hold the bear in place and at the right orientation and have been prepared to exactly conform to the surface of the bear.

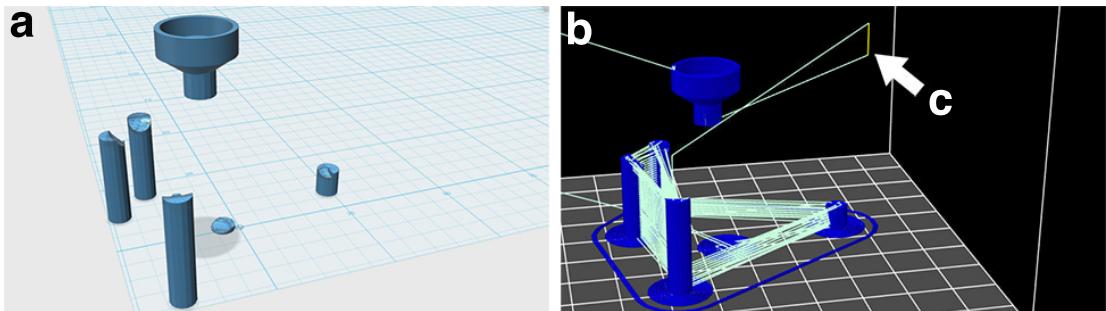


Figure 3.9: In the Teddy bear magnet example, a) a model is generated with a magnet holder, a connector, and the support structure; b) Tool path view in Repetier-Host: the extruder pauses and moves away for inserting the teddy bear (c).

Printing and Post-Processing

In the last phase, the user imports the generated models into the 3D printing tool chain for fabrication. Some attachment techniques require a simple customization of the printing process, which can be programmatically achieved using 3D printing software such as Repetier-Host³ (Figure 3.9b). For example, both print-over and print-through require pausing the printer, inserting the existing object, and then resuming the print job. Having computed the pause point in the analysis, we can generate commands to perform these operations, such as:

```
G1 X0 Y0 ; move the extruder away
G1 Z40 ; raise the extruder
@pause ; pause the print for insertion
; print is manually resumed here after insertion
G1 Z31.550 ; restore extruder's height
```

Figure 3.9c shows the extruder’s tool path as a result of these few extra commands. The extruder makes room for the Teddy bear to be positioned onto the support structures. It then returns and finishes printing the magnet holder.

Finally, some techniques require a few post-processing steps after the attachment is fabricated, such as applying adhesives or straps to affix a new part that is printed separately from the existing object.

3.6.3 Software and Hardware Implementation

We have applied a computational framework—geometric analysis, interactive exploration, model generation, printing and post-processing supporting all three attachment techniques in Encore. Encore was implemented in JavaScript primarily using three.js⁴—a library for programming with WebGL. All the attachment examples were fabricated by an FDM printer made from Printrbot’s Simple Maker’s Kit⁵. We used Slic3r⁶ for G-code generation, Repetier-Host for communicating with the printer, and 1.75mm PLA as printing material in all our examples.

3.7 Evaluation

Below we report on an evaluation of our attachment approach from two perspectives: *(i)* to evaluate the printing process, we compare the cost of time and material for each technique; *(ii)* to evaluate the reliability of the printing results, we tested print-over and print-to-affix techniques, which rely on extrinsic adhesion or fastening. Specifically, for these techniques, we investigate how well a fabricated handle can sustain forces that typically occur when holding, carrying and controlling an object.

³<http://www.repetier.com/>

⁴<http://threejs.org/>

⁵<http://printrbot.com/shop/simple-makers-kit-2/>

⁶<http://slic3r.org/>

3.7.1 Evaluating Time and Material Cost

One potential benefit of printing to augment existing objects rather than creating new ones is to spend less time and material printing. To verify this, we compare the printing plus processing time and material cost between our attachment techniques (print-over, print-to-affix, print-through) and a baseline approach, which prints a brand new object that has the attachment as an integral part of it.

Cost Prediction Model

As shown in Table 1 and Table 2, by the nature of different attachment techniques, we can predict how they differ in printing time and material cost.

Printing time is broken down to time spent on actual 3D printing (T) and time on handling or post-processing (t). For example, print-over requires the printing of the attachment, connector(s), and support structure; it also takes time to insert the existing object to be printed over, as well as to apply adhesives to increase the firmness of the scaffolding.

Material cost is split between printing the attachment and other structures. For example, print-through might also require support structures to keep the interlocking object collision free, even if the attachment itself does not have overhang problems. For the baseline technique, printing time and material is spent on a new object made of the original object combined with the added attachment.

Table 3.1: Prediction models for printing time of attachment techniques and data from a exemplar object+attachment.

Technique	Prediction model for time		Case study: Utah teapot
Print-over	$T_{attachment} + T_{connector} + T_{support} + t_{insert} + t_{apply.adhesive}$		31:41
Print-to-Affix	Adhesive	$T_{attachment} + t_{apply.adhesive}$	13:56
	Strap	$T_{attachment} + t_{strap}$	11:27
Print-through	$T_{attachment} + t_{insert}$		19:43
Print as one piece	$T_{attachment+object}$		1:13:13

Table 3.2: Prediction models for material cost of attachment techniques and data from a exemplar object+attachment.

Technique	Prediction model for time		Case study: Utah teapot
Print-over	$M_{attachment} + M_{connector} + M_{support}$		916mm
Print-to-Affix	Adhesive	$M_{attachment}$	402mm
	Strap	$M_{attachment}$	396mm
Print-through	$M_{attachment} + M_{support}$		582mm
Print as one piece	$M_{attachment+object}$		3054mm

A Case Study

We verify these intuitions in a concrete attachment case: adding a torus-shaped handle to the Utah teapot. We chose the Utah teapot because it is a classic 3D model and because it is also one of the few widely used 3D models where all of our techniques are applicable. The results are shown in Table 3.1 and Table 3.2.

The goal of this evaluation is to reveal how time and cost differ between various techniques, rather than to obtain a ‘true’ value through repetitive trials. As such, we used the slicer program (Slic3r) to estimate printing time and material cost. This program calculates time and cost as it analyzes the input models and generates the corresponding G-code. For techniques that require handling or post-processing, such as applying adhesive, we empirically estimate the time. The particular adhesive we chose (Loctite Professional Heavy Duty Epoxy Loctite #1172794 requires 5 minutes of ‘setting’ time, while strapping usually takes no longer than 3 minutes.

Both the prediction model and case study results support our intuition—when support is needed, time and materials go up; and the baseline condition is dominated by the time needed to print the existing object.

To better understand the tradeoffs, we conducted another evaluation to test the general strength of our print-over and print-to-affix, also compared to the same baseline (an integral print of existing object and attachment). Print-through is often done loosely without binding between surfaces. Thus strength issues for print-through are directly related to the objects themselves, rather than the attachment technique. Similarly, print-to-affix using adhesives depends heavily on properties of the adhesive used. As such, no testing was performed for these two techniques. We report on print-over and print-to-affix using strapping.

3.7.2 Evaluating Strength of Printed Attachments

The goal of this evaluation is to understand the tradeoffs and limits of our techniques in comparison to each other. We tested a type of attachment handles, as they are likely to experience external forces, such as holding, gripping or pulling. For each attachment technique tested, we fabricated handles using the same printer and material as described earlier. Similar to [94]’s test of printed objects’ cross sectional strength, we exerted a bending force tangential to each handle’s attachment point (pulling the handle sideways at 90° with respect to its main axis, as illustrated in Figure 3.11). We then gradually increased the force and measured the point of fracture [89].

Evaluating Print-Over Attachment

For testing print-over attachment, we designed a torus-shaped handle with fixed outer radius of 7.5 mm and inner radius of 2.0 mm. We empirically set these values to match the size of human fingers. Larger scales of handles are much more time-consuming to fabricate and might go beyond the dimension limit of our printer. Thus we leave them for future work.

We then computationally generated a series of 20mm by 20mm by 5mm platforms whereon to fabricate the torus handles, connected via a cylindrical connector (radii: 1.5mm, height: 3mm).

Next we modified the top surfaces of these platforms: the independent variables are the Curvature and Roughness of these top surfaces.

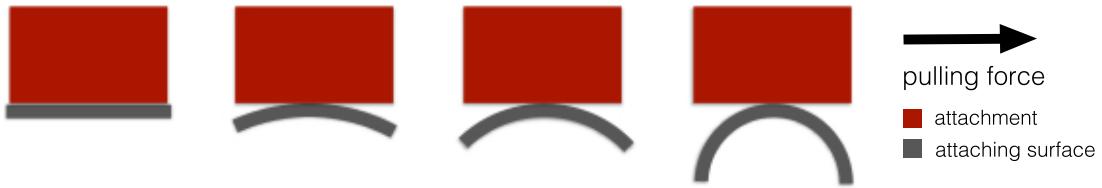


Figure 3.10: To test print-over in comparison to printing in one piece, we used both techniques to print attachments on surfaces with varied curvatures and then exerting tangential pulling force.

- Curvature. Assume R is the radius of a platform's top spherical surface and r is the radius of the cylindrical connector of the handle. We modeled the curvature by $C = r/R$. Specifically, we tested on $C = 0$ (a flat surface), 0.25, 0.5, and 1.
- Roughness. We performed Boolean subtraction operation between the platform's top surface and a pre-set number N of small spheres whose centroids are on the surface, positions randomized, and radii randomly distributed in [0.25mm, 1mm]. This made the surface porous, thus creating roughness. We tested on $N = 0$ (smooth surface), and 10 (rough surface).

We fabricated the platforms and handles under these conditions using print-over and the baseline print-in-one-piece approach.

As shown in Figure 3.11, we clamped the platform firmly onto the top of a table so that the handle was pointing upward. We then used a handheld scale to hook and pull the handle horizontally. We used a slow uniform motion for increasing the force. The process was videotaped to capture the scale's reading at the moment when the handle fractured or detached. We performed three trials for each condition.

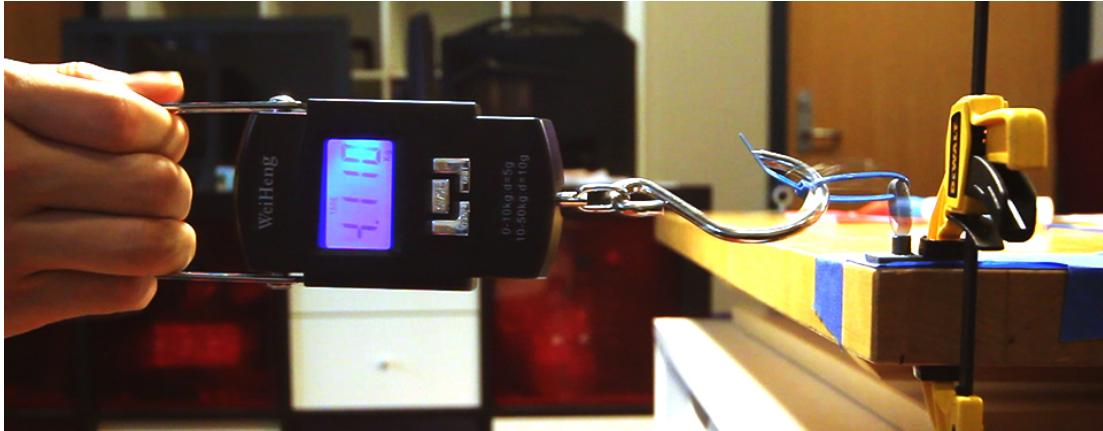


Figure 3.11: Testing the strength of a 3D printed handle attachment (b) by clamping it on a table (c) and pulling using a scale to measure force (a).

Figure 3.12 shows the results of the strength test: overall print-over's best performance was close to that of the baseline. However, it suffered from an increase of surface curvature (Fig-

ure 3.12a) and roughness (Figure 3.12b).

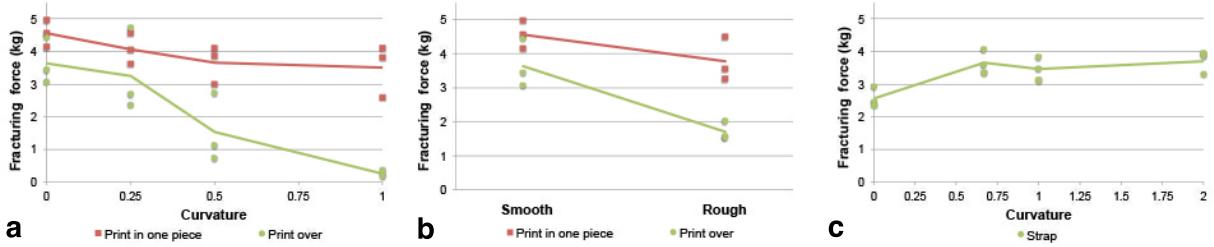


Figure 3.12: Strength test results show print-over is strong enough to sustain stress as if they had been printed in one piece. Both our techniques suffer from an increase of surface curvature (a) and roughness (b); (c) shows that affixing with straps (150mm zip tie) is able to sustain over 2.5 kg of pulling, and reacts differently to curvature.

Affixing using adhesion is also related to this evaluation. However, its performance is hard to test in a general way, as it strongly depends on the particular types of adhesives used, as well as the user's experience and expertise of handling it. Therefore we leave it as future work.

Evaluating Strapping-based Attachment Techniques

Using straps requires a different analysis than print-over. The key question for strapping is how likely the object is to slip' through the strap when pulling forces are exerted. This is a potential issue that does not occur with the baseline and adhesive attachment mechanisms. Thus we tested this affixing technique independently of the other attachment options.

Specifically, given the same cross section S (we use a circular S), we are interested in whether and how the curvature perpendicular to S affects 'strappability'. To test this, we computationally generated a series of olive-shaped objects created by revolving a circular segment (with radius R) around its secant. We computed the secant so that the revolved shape had a cross section of radius $L/2$. We then symmetrically cut off the two ends of so that the remaining pieces have a uniform length of L .

We quantified curvature as $C = L/R$. We tested $C = 0$ (a cylindrical lateral area), 0.67, 1, and 2. For each object, we generated a handle attachment using Encore, and then fabricated it and strapped it around the center of the object using a 150mm zip tie. We then firmly clamped the handle to a table and then pulled on the strapped object, measuring the minimal force required to pull the test object out of the strap using a similar setup as the previous test (Figure 3.13).

Figure 3.12c shows the test results. Depending on the curvature, a 150mm zip tie attached using our approach can sustain a pulling force of from 2.5 to 4 kg without letting the object slip. Interestingly, strapping strength improved with increased curvature (Figure 3.12c), in contrast to other attachment methods (Figure 3.12a).



Figure 3.13: To test strapping in print-to-affix, we increased the curvature between the attachment and the attaching surface and exerted tangential pulling force.

3.8 Summary of Attachment Techniques

Although the tested attachment techniques were developed using the same framework, our performance testing demonstrates that they have their own unique advantages and weaknesses. A limitation of our evaluation is its focus on 3D printed existing objects. In this discussion we provide some anecdotal intuition about other factors that might affect the reliability of attachments fabricated by our techniques. Further, to better understand these techniques as a whole, we also discuss some practical issues in comparison with one another.

Print-over, as shown in our evaluation, creates strong adhesion between attachments and objects that are made of the same or very similar material. It also requires no post-processing. However, it has weak adhesion on some materials. For example, when making the LED torch (Figure 3.1a), our initial attempt to directly print on the metal shell of the battery was not successful. The problem was eventually solved by the simple addition of a thin layer of glue on top of the battery shell. We expect that the use of a glue layer will show promise in other situations as well, but substantially more experimentation and testing is required before its range and properties can be precisely understood.

Affixing with adhesives is widely applicable to a range of geometry and material. But it requires good selection of adhesives, and in some cases careful handling of the amount and mixing of ingredients in order to achieve the best adhesion. Compared with the other techniques (e.g., print-over and print-through), affixing with adhesives relies perhaps the most on the expertise of the users.

Affixing with straps is easily adjustable and also makes an attachment reusable. However, it could be less aesthetically desirable for some use cases. Surprisingly, within the range of our tests, as curvature increased strapping attachment became stronger, whereas all other techniques became weaker. We hypothesize that when the interface between the existing object and the compliant strap has higher curvature, it requires more deformation of the strap in order for the object to slip. Thus the strap can sustain a higher pull force when the curvature increases. However, this hypothesis requires more investigation.

Print-through is a natural attachment mechanism that requires no adhesion or fastening. However, it has limited scope of use as only some objects have appropriate holes or loops that are required to perform this technique.

Despite its increasing popularity, 3D printing has been used almost exclusively to create objects ‘from scratch’, rather than leveraging things that already exist in our day-to-day life. In this project, we present a framework for using 3D printing to augment everyday objects, and

provide specific analytical and computational support for realizing one approach of augmentation adding functional attachments to existing objects. Our framework encompasses key aspects of attachment including geometric analysis, interactive exploration, model generation, and printing and post-processing support. In particular, we present an extensible set of analytical techniques, all based on the geometric form of the existing object and the proposed attachment, and range from very general print viability issues, to durability issues, to very specific usability issues.

There are several remaining questions that require further investigation. Foremost, are there any other attachment techniques that can leverage the use of a 3D printer? Are there any new perspectives that could be introduced into the analyses? Are there any specific use of adding attachments to existing objects that might require further computational support? In my next project, I focus on exploring the third question: I go from developing attachment techniques to enabling the design of functional attachments that can adapt real world objects for custom use.

Another question is given these attachment techniques, how we can design add-on components so that they can serve to augment an object in user-customized ways. My next chapter answers this question through the exploration of designing 3D printed adaptations—components that mechanically enhance or repurpose an existing object.

Chapter 4

Adapting Real World Objects for Custom Users and Use Cases

In our everyday life, we often find a need to adapt objects or tools, as the way they were originally designed might not work for certain users or use cases. For example, a faucet might be out of reach for a child, thus it needs to be extended¹ (Figure 4.2a); it is hard to hold a drill bit straight while sharpening it, thus a guide comes in handy to keep it in place² (Figure 4.2b); painting a big surface with a spray-can could be fatiguing for the fingers, thus a mechanism can be used to turn it into a spray-gun³ (Figure 4.2c). These add-on components are called *adaptations*. Adaptations change the mechanical properties of existing objects to make them more accessible or to customize them for specific use cases.

The advent of low-cost 3D printing offers the possibility to rapidly construct a wide range of adaptations. However, designing or re-purposing adaptations is hard with general-purpose 3D modeling software, as it requires a certain level of expertise from users [33]. In general, 3D modeling tools do not take into account how an object is used, what adaptation strategy is available, or how to rapidly generate the corresponding geometry or further customize it.

Prior work, such as Patching [87], Encore [17] and AutoConnect [43], mostly focuses on attaching or directly fabricating new components onto existing objects. However, it remains unclear how to design these components so that they can serve to adapt the objects in user-customized ways. P.Pod [21] and RetroFab [65] explored the mechanical problem of hijacking and automating controls on existing physical interfaces. However, such mechanisms are usually too heavyweight for adapting common household items and hand tools.

In this chapter, I build on previously explored attachment techniques, and develop a design tool for specifying, generating, customizing and fitting 3D printable adaptations onto everyday objects. My particular focus is on empowering end users to leverage new fabrication technologies to create adaptations by providing high-level techniques for specification and computational support for geometry generation.

Reprise allows a user to specify how an object is used and with what types of action, such as using a virtual hand (Figure 4.1b) to indicate how a person would hold a wire cutter. As 3D

¹<http://peachyco.com/handleextender.html>

²<http://www.thingiverse.com/thing:1024834>

³<http://www.thingiverse.com/thing:51966>

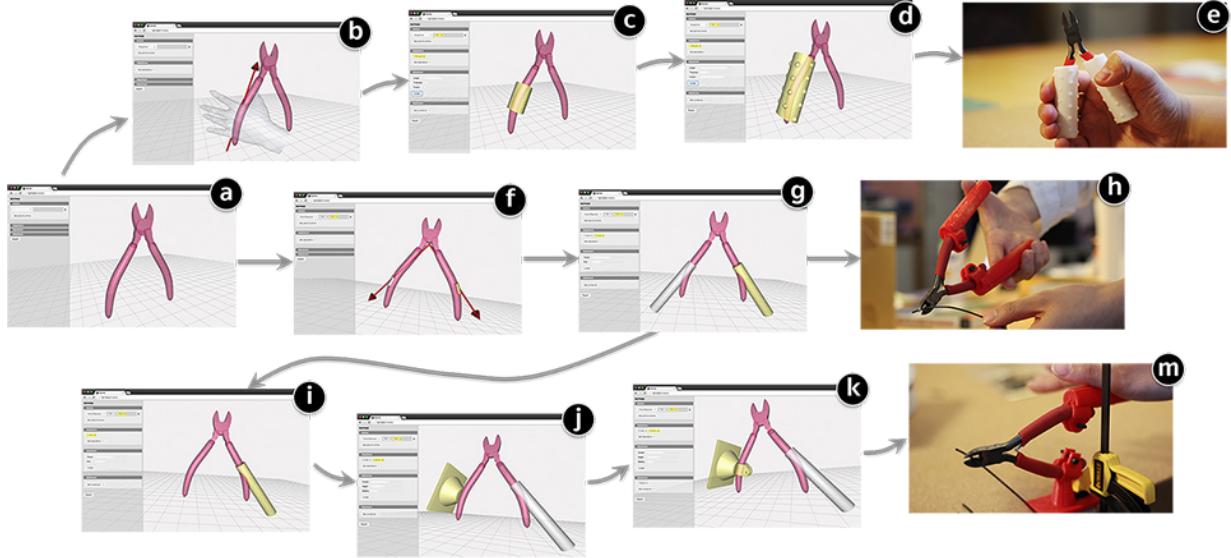


Figure 4.1: Our main contribution is the tool integration and a formalized design workflow for making 3D printable adaptations onto everyday objects. For example, an occupational therapist can use our tool to explore different strategies of adapting a wire cutter (a), such as creating a wrapper to soften the grip (b-e), adding two levers to assist with clutching (f-h), or replacing one lever with an anchor to situate the cutter on the work surface (i-m).

geometry itself does not encode how an object is used in the real world, Reprise’ techniques enable a user to interactively describe this information *in situ* on models of the objects.

Once the actions are specified, this information is fed into a library of design strategies—Wrapper/Extension, Handle, Lever, Anchor/Stand, and Guide (Figure 4.3). These adaptation strategies were derived from an analysis of over 3000 lifehacking and assistive technology examples found in three books [63, 68, 109] and two online communities [61, 90]. A user can select one or more strategies, which automatically generate the initial design of an adaptation, such as a wrapper for a cutter’s handle (Figure 4.1c), levers for easing clutching operation (Figure 4.1g), or a structure to anchor the cutter for situated use (Figure 4.1i).

To further iterate on the design, a user can manipulate simple slider widgets to customize these adaptations to better suit the person’s needs and preferences, such as making the cutter’s wrapper longer and thicker for easier gripping (Figure 4.1d), increasing the levers’ torque for clutching (Figure 4.1g), or making a larger base for stability (Figure 4.1k). Finally, Reprise also provides a simple toolkit for making the adaptations more attachable onto the objects, such as generating a pipe clamp that connects the cutter’s handle to the anchor (Figure 4.1k).

To demonstrate the expressiveness as well as understanding the limitations of Reprise, we replicated existing adaptation examples found during our design space exploration (Figure 4.3). We identified a real world exemplar for each of 18 cells of the two-dimensional design space. Among the 18 examples, Reprise was able to replicate 15 of them; the other 3 cases suggest new features to be added to the tool and fertile opportunities for future work.

The main contribution of the Reprise tool lies in its ability to generate and iterate across a



Figure 4.2: Existing examples of adapting everyday objects: a faucet extension (a), a guide for sharpening drill bit (b), and turning a spray-can to a spray-gun (c).

range of likely useful adaptations. However, it can also be seen as an early exemplar in a class of design tools which can make 3D printing more accessible and useful for ordinary people. In particular, it goes beyond the specification of geometry alone to provide very application domain specific knowledge and features. This helps to substantially reduce the gulf of execution [56] that users must bridge as they cross from function to geometry, e.g., from geometric form to the implications of that geometry on the desired end result.

4.1 Scenario Walkthrough: Adapting a Cutter

Reprise lets a user explore a range of design strategies for adapting everyday objects. To set the scene, we first walk through the flow of Reprise in an exemplar scenario where Larry—an occupational therapist—is trying to adapt a wire cutter for an electrician who recently suffered from a hand injury.

4.1.1 Iteration I: Generating Wrappers for a Better Grip

To start, Larry imports a 3D model of the cutter (from a provided repository or through scanning). He selects ‘Grasp/Hold’ as the type of action the patient is having difficulty with. He then clicks on the handle of the cutter to specify the part that is being grasped. Reprise shows a virtual hand model, which Larry can rotate to indicate the relative orientation between the gripping hand and cutter (Figure 4.1b). Next, Larry selects ‘Wrapper/Extension’ as the adaptation strategy. Based on the previously specified grasping action and its hand orientation, the system generates an initial design of the wrapper (Figure 4.1c). Larry wants to further tweak the design using simple slider controls. He adjusts the ‘Length’ slider to make sure the wrapper covers an area large enough for three fingers. He then increases the thickness to make a rounder grip. To further tighten the grip, he adds small bumps to increase the friction (Figure 4.1d). Finally, Larry exports the generated geometry as an STL file and prints the adaptation using a soft material (such as NinjaFlex⁴), which provides a soft grip and makes the wrapper snugly fit with the cutter’s handle (Figure 4.1e).

⁴<http://www.ninjaflex3d.com/products/ninjaflex-filaments/>

4.1.2 Iteration II: Making Levers to Assist with Clutching

Larry's patient responds positively to the soft wrapper. However, while he can now comfortably hold the cutter, his hand is still too weak to perform the necessary clutching action, especially when cutting hard wires. Thus Larry explores an alternate design strategy. In Reprise, he first selects 'Clutch/Squeeze' as the target operation. He selects the two handles as the components on which clutching is applied, and then moves the cursor to indicate the fulcrum of clutching. In these few steps, he specifies the two arms for clutching (Figure 4.1f). With this information, Larry proceeds to select 'Lever' as the adaptation strategy, which automatically generates two levers from the handles (Figure 4.1g). Larry then tweaks the design, playing with the torque of the levers, as well their size for grasping.

4.1.3 Iteration III: Anchoring the Cutter for Situated Use

Larry's patient does not like the lever design, as it makes the cutter somewhat too big to hold in one hand (Figure 4.1h). To solve the problem, Larry uses Reprise to replace one lever with an anchoring structure with which he can affix one of the cutter's handles on the work surface (Figure 4.1ij). The patient then simply uses his palm and body weight to press down and cut wires. The key here is to firmly attach the anchor to the cutter's handle. Reprise's attachment toolkit allows Larry to generate a pipe clamp for holding the cutter in place on the anchor (Figure 4.1k). Once printed, Larry bolts the cutter on the anchor, which is then clamped onto the patient's work surface (Figure 4.1m).

To summarize, Reprise allows Larry to generate adaptations with just a few clicks, to tweak the design using simple slider controls, and to explore different design strategies while iterating and customizing the adaptation to suit the patient's particular needs and preferences. Below we describe a design space that informs the design and implementation of Reprise.

4.2 A Design Space of Adaptation Strategies

For years, well before digital fabrication was widespread, people have been taking a 'bricolage' approach for making adaptations on objects that would otherwise be difficult to use. In particular, many innovative solutions came from the need to create assistive technologies (AT) for people with special needs. For example, Werner documented his effort in making assistive technology for village children with very limited resources [101]. Books like [63, 84, 109] have compiled simple recipes for individuals to adapt household items and tools with materials that can easily be found at home. Although started with a small and specific audience, these AT design and making solutions usually have a broader impact and implication on universal design [22].

Another source of inspiration is online communities such as Thingiverse [90] and Pinterest [61], which curate a repertoire of lifehacking solutions people have explored and shared with one another. Recent work by Buehler et al. [16] has summarized a plethora of assistive technology from Thingiverse, which suggests promising opportunities of learning from these resources.

To inform the computational design of adaptation, we collected and reviewed over 3000 lifehacking and assistive technology examples from the aforementioned literature (85 from [63],

Types of Action Adaptation strategies	Static	Linear Motion		Rotational Motion	
		1-part	2-part	1-part	2-part
		Grasp/Hold	Press/Pull/Slide/...	Join & Separate	Rotate
Wrapper/ Extension	A1	B1		D1	E1
Handle	A2	B2		D2	E2
Lever	A3		C3	D3	E3
Anchor/ Stand	A4	B4		D4	E4
Guide		B5	C5		

#	Adaptation	Source	Link*
A1	Fork gripper	Adapt My World	N/A
A2	Phone holder	AT in Minutes	N/A
A3	Reacher	Pinterest	https://www.theindependencecenter.org
A4	Cup stabilizer	Pinterest	https://www.especialneeds.com/
B1	Zipper puller	Pinterest	https://www.pinterest.com/pin/439593613602539707/
B2	Door handle	Pinterest	http://www.aidsfordailyliving.com.au/daily-living/arthritis-handle.htm
B4	Pill popper	Pinterest	https://www.pinterest.com/pin/336995984589953720/
B5	Remote quide	AT in Minutes	N/A
C3	Bottle opener	Thingiverse	http://www.thingiverse.com/thing:1599
C5	Drilling quide	Thingiverse	http://www.thingiverse.com/thing:267196
D1	Tennis ball jar opener	Pinterest	http://www.instructables.com/id/Tennis-Ball-Jar-Opener/
D2	Key handle	Pinterest	http://www.essentialaids.com
D3	Faucet turner	AT in Minutes	N/A
D4	Bellclamp jar & bottle opener	Pinterest	http://www.statina.com.au/bellclamp-jar-bottle-holder.html
E1	Rounded safety scissors	Pinterest	https://www.teaching.com.au/
E2	Handles for a nail cutter	Pinterest	http://www.aidsfordailyliving.com.au/daily-living/nail-clippers.html
E3	Spray bottle lever	Pinterest	https://www.pinterest.com/pin/90142430014475114/
E4	Anchored scissors	Pinterest	https://www.pinterest.com/pin/74802043786592112/

* For Pinterest examples, we provide the direct link if there is one

Figure 4.3: A design space of adaptations summarized from over 3000 existing examples: five major adaptation strategies support various types of actions, from static grasp and hold, to linear and rotational motion that involves one or multiple parts of the objects.

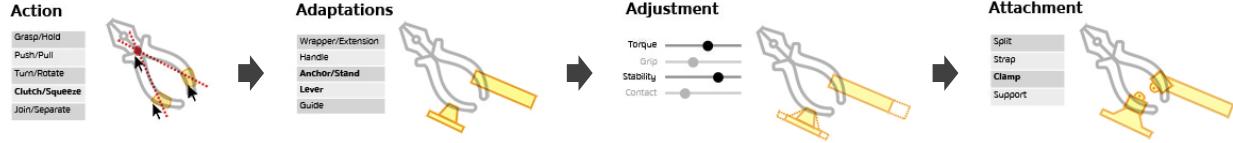


Figure 4.4: The design workflow of Reprise. Start with specifying the types of *action* applied on an object, which serves as input for generating user-selected *adaptations*. The initial design can be further customized by *adjusting* a set of parameters. Finally, extra fasteners can be added to make adaptations more *attachable* onto the object.

56 from [109], and 33 from [68]) and online communities (950 from Pinterest and 2028 from Thingiverse using queries ‘assistive’, ‘technology’ and ‘adaptation’). Amongst these examples, we identified adaptation designs, and organized them into themes in a bottom-up fashion. This sampling process continued until we felt that we had reached saturation.

We present our findings as a design space of adaptation strategies. As shown in Figure 4.3, we identified five major categories of adaptation designs, as well as the types of actions they support to hold, control or operate the objects. While some examples might fit in multiple categories, this design space captures the range of ideas we found in our review and suggests opportunities for novel adaptations (which we discuss in the Results section).

#1 Wrapper/Extension is something wrapped around an existing object or an extension that lengthens or expands it. Wrapper and extension support various actions with objects. A wrapper can be used to soften the grip (A1), or to assist with grabbing and rotating an object with less force or less fine motor control (D1). An extension can help with pulling an object (B1) or—with a specific design—to enhance safety for cutting tools (E1).

#2 Handle is an add-on grippable part attached to an existing object, which makes the object easier to hold (A2), pull (B2), or rotate (D2). Adding handles also helps with clutching, such as using a nail cutter (E2).

#3 Lever is a structure added to an existing object, usually to make it easier to be turned or rotated, such as the lever used in D3 for turning a faucet, or in E3 for squeezing a spray bottle. In other cases, a lever also affords actions other than rotation, such as grabbing objects (A3), or opening a beer bottle (C3).

#4 Anchor/Stand is a structure with which an existing object can be stably situated or affixed to the environment. Figure 4.3 shows a stand to hold the cup stable (A4), a pill popper for holding a pack of pills and popping them out (B4), a clamp for holding a jar or bottle for easier opening (D4), and an anchor for a pair of scissors for situated use (E4).

#5 Guide is a structure that guides the movement of, or against, certain objects or their components. B5 shows a guide for locating and pressing buttons on remote controls. C5 is a guide for drilling a screw vertically towards and into a surface.

Having summarized these five major adaptation strategies, below we describe the technical details of Reprise—a tool that integrates and implements these strategies for generating 3D printable adaptations.

4.3 Reprise: Computational Design of Adaptations

Figure 4.4 shows the workflow of Reprise. Users start with importing a 3D model of an object they want to adapt, specify what types of action is applied, and on which parts of the object. Next they select an adaptation strategy from a library provided by Reprise, which automatically generates the geometry. This initial design can then be customized by adjusting a set of parameters. Finally, extra fasteners can be added to better attach the adaptations onto the object. Reprise then exports all the generated geometry as 3D printable STL files.

4.3.1 Techniques for Specifying Types of Action with an Object

As a user selects a type of action, Reprise provides interaction techniques for specifying where and how the action is applied on the object. As 3D geometry itself does not encode how an object is used in the real world, Reprise’s techniques enable the users to specify this information *in situ*, which later serves as input parameters for generating adaptations. Most of these techniques—as described below—start with the user selecting one or more *points of action*—the locations on the object where it will be held, controlled, or manipulated.

Grasp/Hold As shown in Figure 4.5, Reprise displays a virtual hand, which can be rotated by moving the mouse cursor around the point of action. This allows users to describe the relative orientation between the grasping hand and the object, such as forming a cylindrical grasp [20] on a fork (Figure 4.5a), or a spherical grasp [20] on the cap of a water bottle (Figure 4.5b). Reprise also provides a simple jig (Figure 4.16) for measuring the target user’s hand size, which can then be entered into the system.



Figure 4.5: Reprise shows a virtual hand to let the user specify how an object is grasped, such as forming a cylindrical grasp on a knife (a), or a spherical grasp on a bottle (b).

Push/Pull A spherical control is used to let the user specify the direction of the push/pull, such as pressing a small power button on a remote (Figure 4.6a), or pulling a zipper pull horizontally (Figure 4.6b).

Rotate The user selects the plane (XY , YZ or ZX) on which the object is rotated, such as the vertical plane on which to turn a door handle (Figure 4.7b). Next the user selects the fulcrum (Figure 4.7c) within that plane. Reprise dynamically displays an arrow pointing from the fulcrum to the point of action to help the user specify the rotating arm (shown as the arrow).

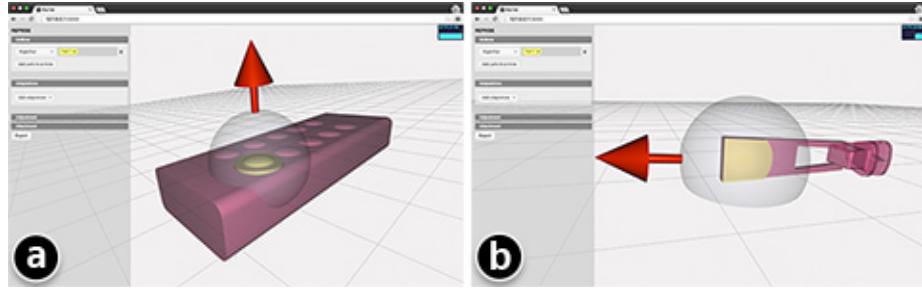


Figure 4.6: Reprise uses a spherical control for specifying pushing/pulling an object, such as pressing a button on a remote control (a), or pulling a zipper (b)

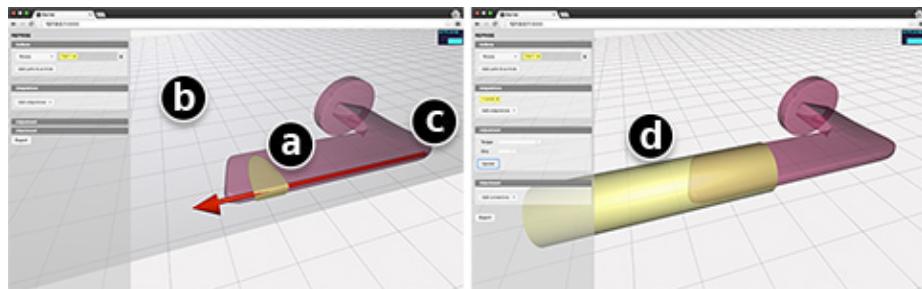


Figure 4.7: To specify rotating an object, Reprise lets a user select where the object is held (a), on which plane it is rotated (b), and the fulcrum of rotation (c). The red arrow shows the rotation arm along which a lever can be generated (d).

Clutch usually involves two components, such as clutching the two handles of a cutter. Thus the user will select two points of action on the object, such as the handle and the neck of a spray bottle. The plane of clutching is then computed from the normals of the selected points as well as the line segment formed between them. Similar to Rotate, the user then selects a fulcrum on that plane where two arrows are shown indicating the directions of the clutching arms ((Figure 4.8a)).

Join/Separate involves two objects moving towards or away from each other. To simplify the problem, Reprise uses their bounding boxes to describe how the objects will be joined or separated. Specifically, the user will select two faces, respectively, from the two bounding boxes, which then shows the direction along which the two objects will move towards or away from one another. For example, as shown in Figure 4.9a, the key would go into the lock hole along its teeth as indicated by the arrows.

4.3.2 Computationally Generating Adaptations

Taking the user-specified actions as input parameters, Reprise then provides a list of strategies for rapidly generating the initial design of adaptations. The specific methods for generating these models are based on the aforementioned survey of existing adaptation examples (Figure 4.3). In generating these adaptations, Reprise uses variations of a cylindrical geometry as the primary

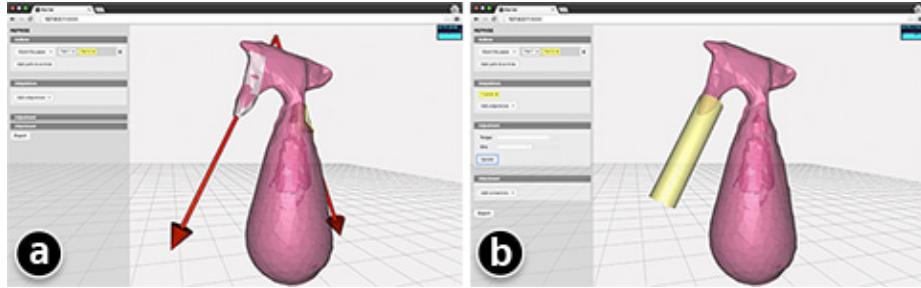


Figure 4.8: To specify clutching, the user selects the two parts of the object that are being clutched and then the fulcrum. The red arrows indicate the directions of the clutching arms (a). The user can generate a lever for clutching the spray bottle, while holding its body in hand (b).

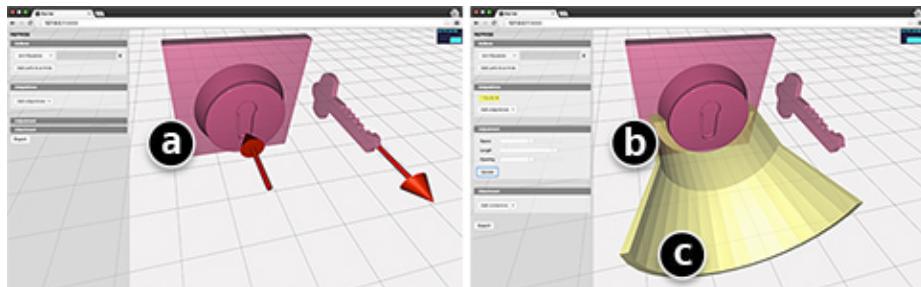


Figure 4.9: Reprise lets the user specify moving one object towards or away from another, such as putting a key into a lock hole.

building blocks, and a series of Constructive Solid Geometry (CSG) operations for creating the specific adaptations.

For example, a cylinder wrapping around part of an object can make a wrapper (Figure 4.10a) or a lever (Figure 4.7b and Figure 4.8b). Extruding a cylindrical structure from an object’s surface creates an extension, such as enlarging a button on a remote control (Figure 4.10b). To make an anchor, Reprise creates a cylindrical pillar that connects to a rectangular base at the bottom, and to the object at the top (Figure 4.13a). When creating a guide, it is also possible to use an object’s bounding cylinder to represent its path of movement, which creates a ‘tunnel’ (Figure 4.9b) with a wide ‘opening’ at the entrance (Figure 4.9c). By default Reprise displays the models in 1:1 scale to let the user get a concrete sense of the adaptations’ size and dimensions when adjusting these parameters.

4.3.3 Adjusting Design Parameters for Customization

With the initial design of the adaptations, Reprise provides users with simple slider controls to adjust the design parameters for further customization. For example, for gripping, they can lengthen a wrapper so that it can be held by multiple fingers (the sizes of which can be measured

using a tool described later in Figure 4.16); they can also adjust the radii of the cylindrical components, or add small bumps on the surface to tighten the grip (Figure 4.10a). To enhance strength and support, they can increase the torque of the levers, or give the anchor or stand a larger base. For easier hand coordination, they can widen the opening in a guide, making it easier to move one object into the ‘tunnel’ and towards the other object (Figure 4.9c). To adjust the overall size, they can make a handle more elliptic or vary the length of its arc (Figure 4.11c).

To help users navigate the rich parameter space, Reprise also allows them, with one click, to generate a large set of design variations (Figure 4.11a→b), similar to the technique in Side Views [88]. This provides them with a more intuitive way of browsing and comparing different designs, select one that best matches what they have in mind (Figure 4.11b→c), and from which they can continue to customize the model (Figure 4.11c). For example, as shown in Figure 4.11, to make an adaptation for grasping a fork, the user can select wrappers of different length, thickness, and tightness, as well as handles of different sizes and styles.

4.3.4 Attaching Adaptations onto Real World Objects

As a last step, the user normally wants to generate some attaching mechanism(s) for installing the adaptations onto the object. Prior work, such as Encore [17] and AutoConnect [43], has explored a range of attachment techniques, which can be used as a post-processing step to install the adaptations.

In addition, Reprise also offers several simple built-in techniques for making the adaptation more attachable. For example, ‘Split’ lets the user draw a stroke on the model of the adaptation (Figure 4.12a) to split it into halves (Figure 4.12b), such as splitting a wrapper so that a fork can be placed in it (Figure 4.17a). ‘Clamp’ lets users create a simple pipe clamp: first positioning the pipe (Figure 4.13b) and then stroking a line to cut the pipe open and generate clamps (Figure 4.13cd) where a bolt can be used to fasten the structure. ‘Beams’ can be added to the adaptation by selecting a point on the object and another point on the adaptation (Figure 4.14ab). This will generate structures to further support the object, such as to further stabilize a mug (Figure 4.14c).

In some cases, part of an object is not accessible or should not be occluded by the adap-

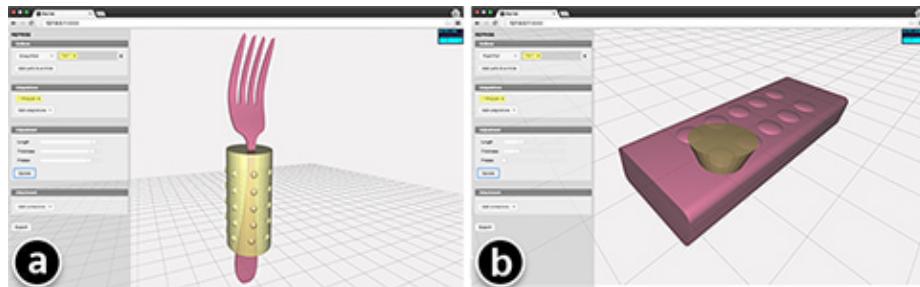


Figure 4.10: Reprise can generate a wrapper—a cylindrical structure bounding part of an object (a), or an extension—an extrusion from a selected area of the object (b).

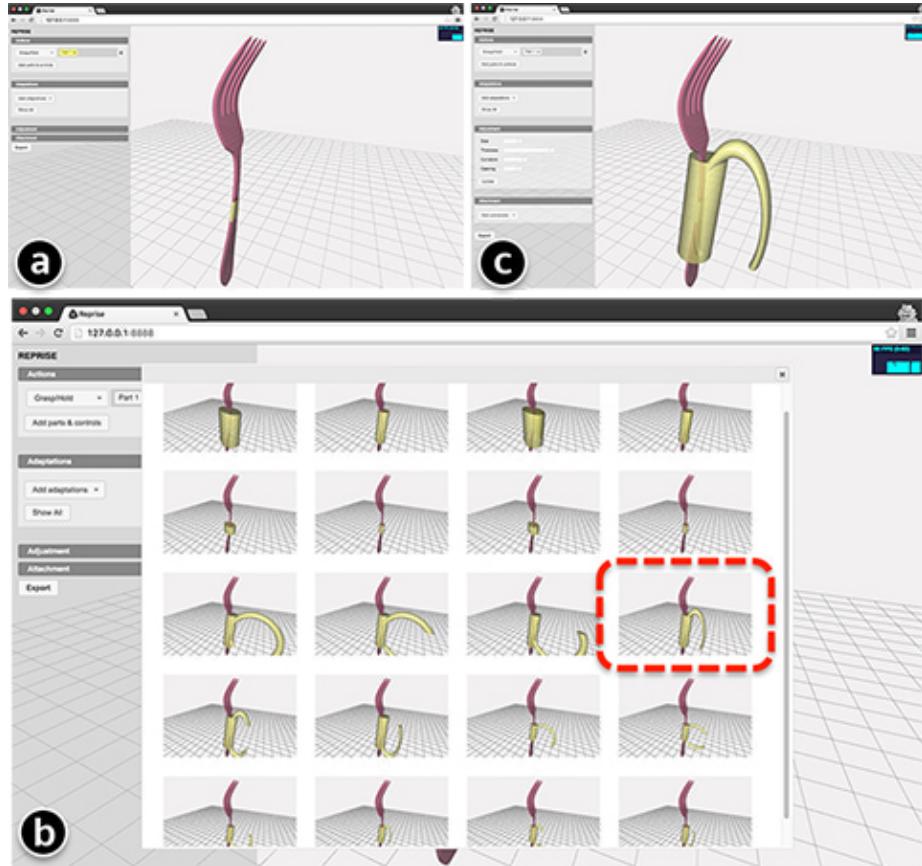


Figure 4.11: Besides using sliders, users can also click a button to generate a large set of design variations (ab), from which they can select one that best matches the design they have in mind, and continue to develop it from that model (c).

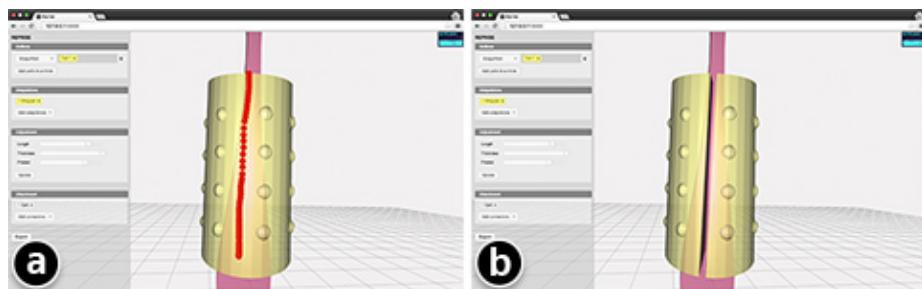


Figure 4.12: Reprise provides a simple technique to split an adaptation in halves so an object, such as this fork, can be put inside a wrapper (Figure 4.17a).

tations, such as a phone's screen. Reprise allows a user to specify which part of the object should be excluded when generating adaptations. As shown in Figure 4.15a, the phone's screen

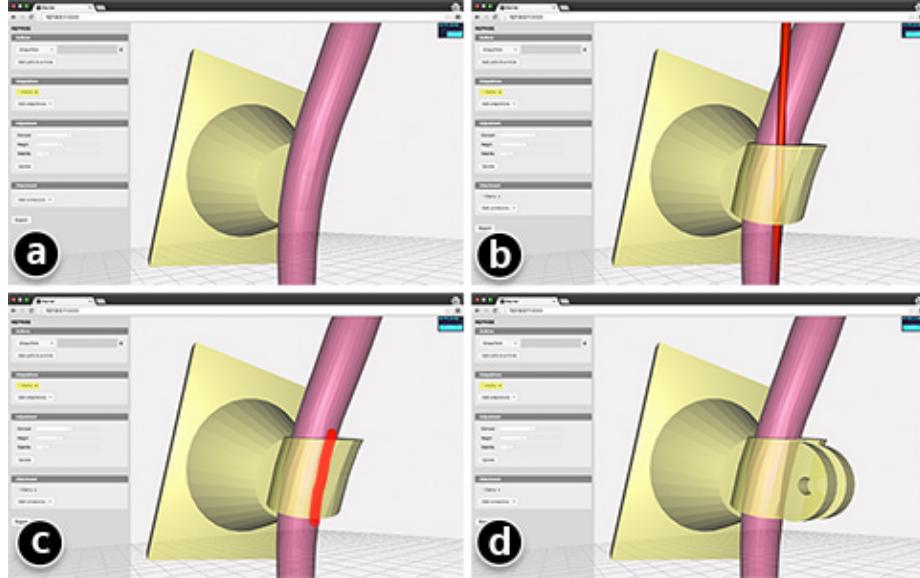


Figure 4.13: A pipe clamp, provided by Reprise, attaches the generated stand (a) to the object. Clicking on the object places the pipe (b). Another stroke cuts it open and adds a pair of clamps (cd).

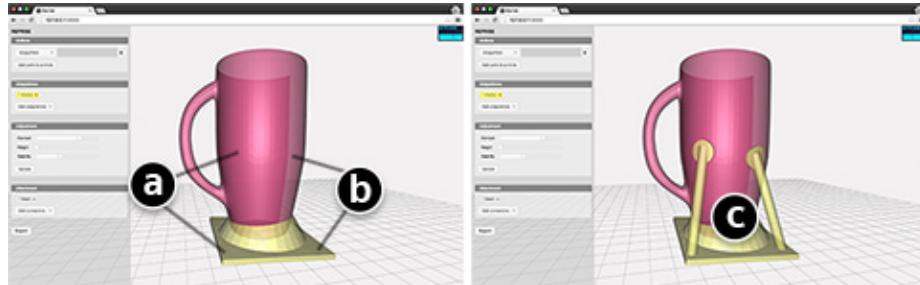


Figure 4.14: Beams can further support the object for an adaptation. To generate one, simply select a point on the object and another point on the adaptation.

is marked as inaccessible (red), and accordingly the generated handle avoids covering the screen area (Figure 4.15b).

4.3.5 Implementation, Measurement and Fabrication

Reprise was implemented using JavaScript with three.js⁵ and OpenJSCAD⁶. The tool can run readily inside a modern browser. Reprise curates a 3D model repository of common household

⁵<http://threejs.org/>

⁶<http://openjscad.org/>

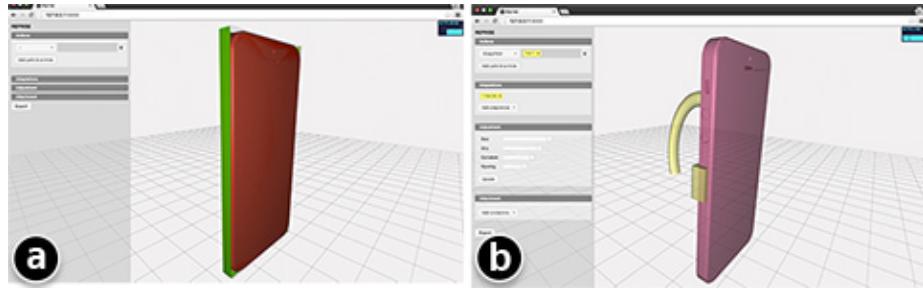


Figure 4.15: As the phone’s screen is marked as inaccessible (red) by an adaptation, the generated handle avoids covering that area.

items and hand tools; alternatively, it also uses Skanect⁷ and the Makerbot Digitizer⁸ for digitizing objects. The spray bottle in Figure 4.8 was scanned using the Skanect system.

For some adaptations, users might need to measure the hand or finger sizes of the person. Reprise comes with a simple measuring tool (Figure 4.16) that can be printed (and cut) on a piece of paper, or made with a laser cutter. The hand measuring part is based on a method for determining the size for a golf glove⁹. The finger part is based-on standard ring-size metrics.

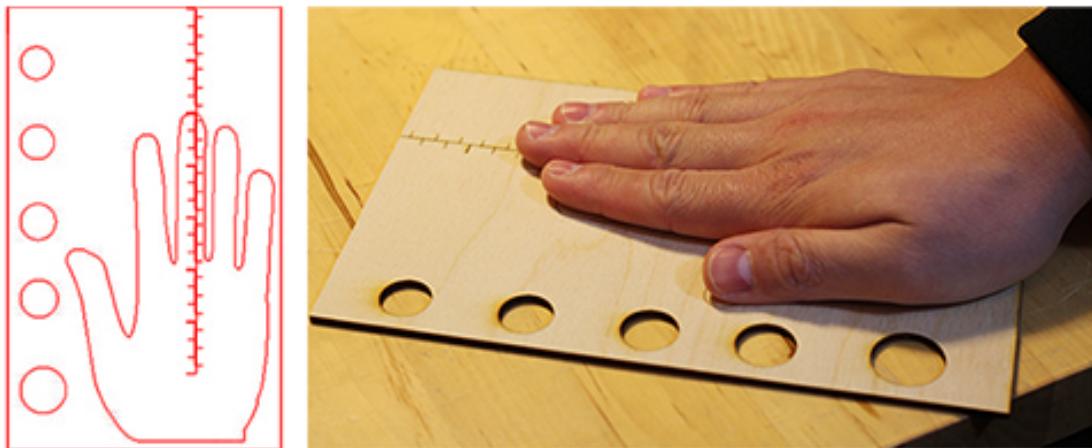


Figure 4.16: Reprise also provides an SVG file for making a measuring tool for adaptations that require fitting users’ hand or fingers. This one shown above was laser cut.

For fabrication, we used material with various softness, ranging from regular PLA, to Semi-

⁷<http://skanect.occipital.com/>

⁸<http://store.makerbot.com/digitizer>

⁹<http://www.jumbomax.com/sizing/>

Flex¹⁰, Nylon-based¹¹ or NinjaFlex¹². All the results—as we demonstrate below—were fabricated using an inexpensive Printrbot Play 3D printer¹³.

4.4 Results and Validation Through Replication

Compared to traditional CAD tools where models are mostly created from scratch, Reprise has integrated application domain specific knowledge that allows users to specify, generate and customize adaptations at a higher level. While there may remain some usability issues with our tool, this application domain targeted approach clearly allows adaptations to be specified much more easily than trying to create them as unstructured arbitrary geometry in a general-purpose tool. However, this ease of use can come at a price, and this raises the important issue of whether the tool can create a wide range of objects to cover an interesting and useful space. To answer this more difficult question, we validate Reprise by focusing on its breadth of expressiveness. In particular, we used it to replicate a wide spread of existing adaptation examples. This demonstrates the potential for Reprise to support the full range of adaptations that are demonstrably useful in the real world.

Specifically, we chose to replicate representative cases from different ‘cells’ in the design space of adaptations (Figure 4.3). Below we showcase our results, discuss issues that arose during the process, and identify cases where Reprise is not yet able to generate the adaptations, which suggest opportunities for future work.

#1 Wrappers/Extensions As shown in Figure 4.17, Reprise was able to replicate all the four wrapper/extension examples in the design space. The virtual hand was effective in positioning the wrapper at the right place on the objects. However, in the case of the cutter handle, we found that the simple cylindrical structures did not fully conform to the curvature of the handles (Figure 4.17d). A new feature for future work would be to ‘curve’ the wrapper so that it can better represent the original geometry of the object.

#2 Handles All types of handle examples in the design space were replicated using Reprise (Figure 4.18). Reprise allows users to explore different handle designs, such as making a ‘closed loop’ for holding and turning a key (Figure 4.18c), or half-open for easier placement of the hand when holding a phone (Figure 4.18a) or pulling open a cabinet door (Figure 4.18b).

A single handle can be used to stabilize the grip of a pen (Figure 4.22a) while it is also possible to combine multiple handles to match different fingers (Figure 4.18d). In particular, for fingers, initially we found it really hard to specify a fixed spatial relationship between them and the handles, as their position and orientation could vary from time to time while holding an object. Later we found that using soft material could mitigate this problem, as its deformable nature offers a range of adjustable configurations. Although Reprise operates at the geometry level, it is possible, as a future step, to fine tune the geometry of a handle as a way to ‘program’ its range of configuration when printed with soft material.

¹⁰<http://www.ninjaflex3d.com/products/semitflex/>

¹¹<http://www.graphene3dlab.com/>

¹²<http://www.ninjaflex3d.com/products/ninjaflex-filaments/>

¹³<https://printrbot.com/shop/assembled-printrbot-play/>

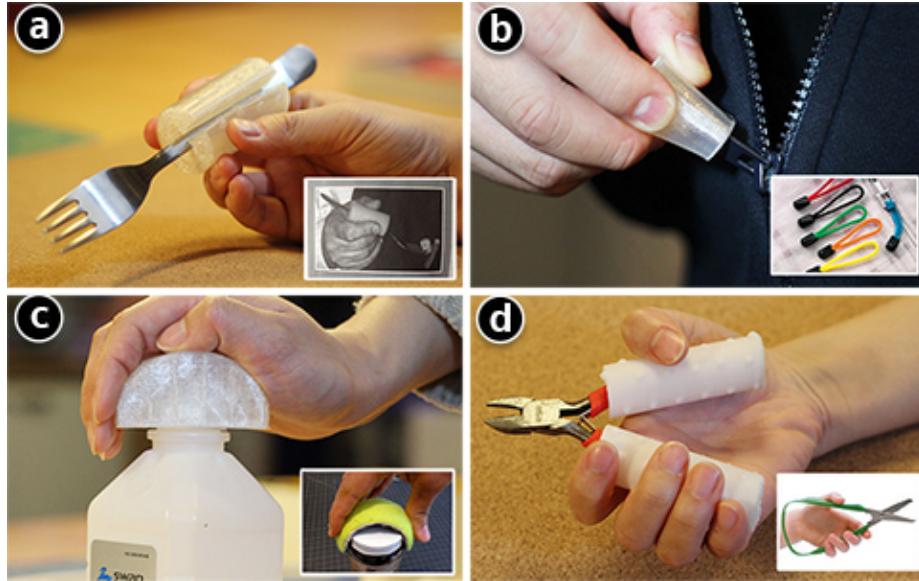


Figure 4.17: Replicated wrapper/extension examples. Ninjaflex was used for the fork wrapper (a), zipper handle extension (b), and the bottle lid wrapper (c). Nylon-based soft material was used for the cutter.

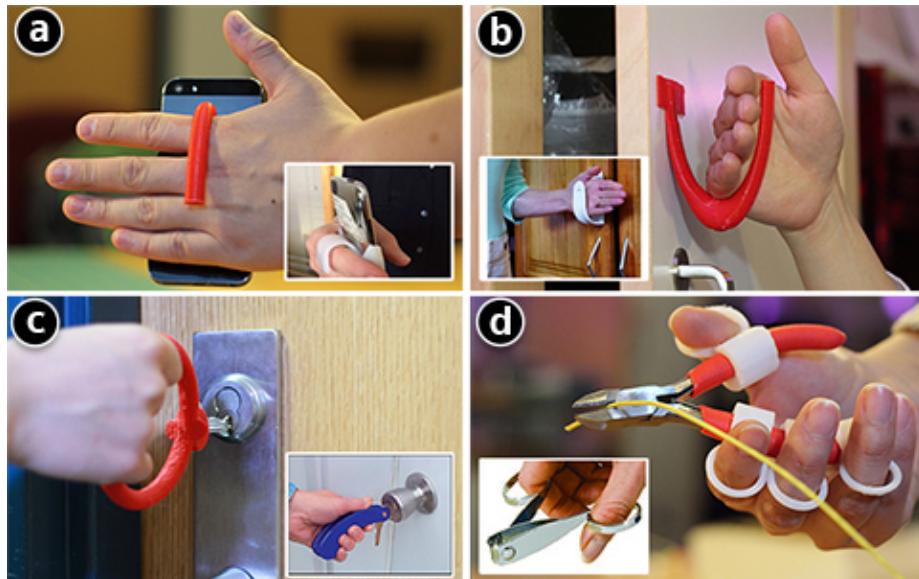


Figure 4.18: Replicated handle examples.

#3 Levers Figure 4.19 shows Reprise' replication of levers. One current limitation we found is that Reprise can only generate levers where part of the existing object has a degree of freedom to be rotated or squeezed. In the future we plan to explore other use cases of a lever, such as for clamping (Figure 4.3-A3) or separating (Figure 4.3-C3) static objects.

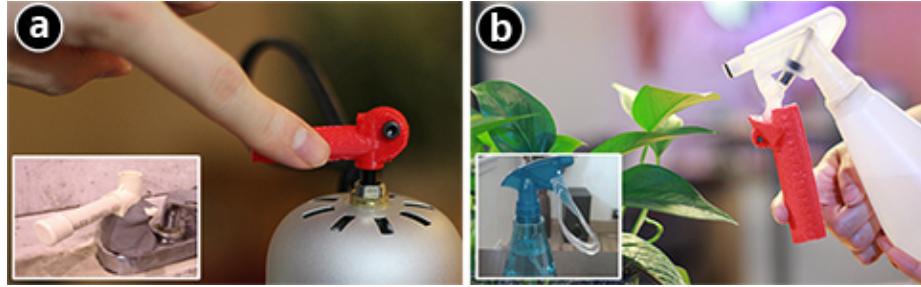


Figure 4.19: Replicated lever examples.

#4 Anchors/Stands Besides the aforementioned cutter, we also replicated two other anchor/stand examples as shown in Figure 4.20. When making these examples, we often found it necessary to add extra fasteners or support structures, such as the pipe clamp for the cutter (Figure 4.20a) and the supporting beams for the mug and the sharpie anchor (Figure 4.20bc). Rather than adding them as a follow-up step, future work could integrate these components parametrically as part of the generated adaptation.

#5 Guides Inspired by the drilling guide example (Figure 4.3-C5), we used Reprise to generate similar structures for guiding a key towards a lock (Figure 4.21a) and putting a sharpie back into its cap (Figure 4.21b). We did not replicate the remote control example as it stands (Figure 4.3-B5), as it seems simple enough to be made with paper, and further it sacrificially occludes other buttons. Instead, we opted for an alternate design—an extension on the button that makes it a larger target for pressing (Figure 4.10b). It is interesting to see that in this case an extension can also serve as a guide, which suggests the possibility of ‘remixing’ categorically different adaptation strategies.

One limitation we found is that Reprise currently only guides linear motion; in the future it should be possible to explore how to generate guides for other types of motion, such as a guide for turning faucets to get (only) mildly warm water.

4.5 Discussion and Summary of Designing Adaptations

Based on our exploration of the design space, tool integration and fabricated results of adaptations, we now discuss existing issues, limitations and potential future work.

Combining different adaptations With Reprise, it is also possible to combine different adaptation strategies. Figure 4.22 shows a series of sharpie adaptations generated by Reprise, ranging from a handle for holding (Figure 4.22a), the aforementioned guide (Figure 4.22b) and anchor (Figure 4.22c), and finally, a combination of the three (Figure 4.22d). Our current approach is to incrementally add new adaptation with the previous ones considered as part of the object. However, the ordering of addition could cause adaptations to conflict with one another. For example, adding a handle might make the sharpie no longer fit in the guide installed earlier. Future work could optimize the order of adding multiple adaptations to reduce conflict, or updating existing adaptations as new ones are added.



Figure 4.20: Replicated anchor/stand examples.

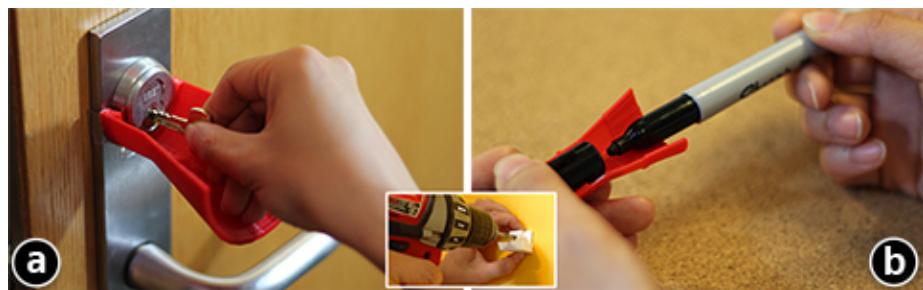


Figure 4.21: Replicated guide examples.

Expanding the feedback loop for customization Rather than just taking the original object as input, future work could enable a more iterative approach where a printed piece of adaptation can be fed into the system to let the user further customize it. Prior work, such as ModelCraft [82] could potentially be extended here to let the user annotate printed results for the next design

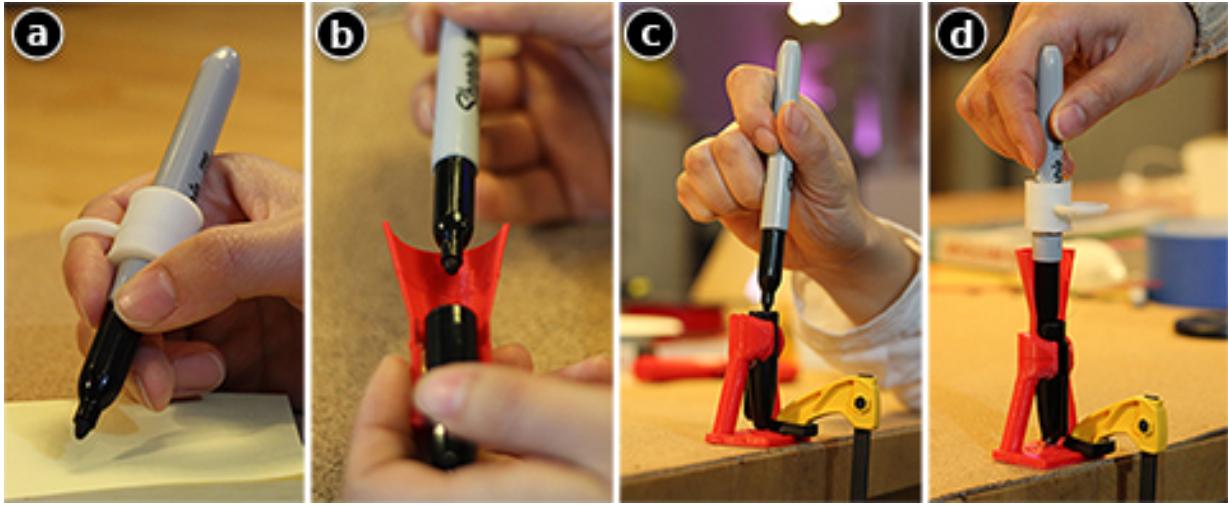


Figure 4.22: Combining multiple adaptations: a handle (a), a guide (b) and an anchor (c) can be combined into one (super) adaptation (d).

iteration.

Scaling up reprise to adapt larger objects Most of the objects Reprise adapted so far are hand-sized items or tools. In future it would be interesting to explore larger scaled adaptations, such as making a bath tub safer to walk in/out, or making room entrances or stairwells more accessible. The challenges are two-fold: how to scale up the current design workflow to incorporate large objects, and how to go beyond the usual volume of 3D printers to fabricate the adaptations.

Combining hand-making and digital fabrication Although we designed Reprise for 3D printing, we believe it may work best as a tool that complements existing hand-making approaches, rather than replacing them. Hand-making can capture users' intuition and creativity (even though the outcome is still limited by the materials available and the users' making skill). Future work could transform Reprise's workflow to incorporate or augment handmade prototypes, such as using 3D printed fasteners to connect handmade parts, or to serve as components that require higher functional precision.

Using vision-based sensors to capture object usage As we designed the virtual hand for specifying different ways of grasping, we realized the possibility of an alternate approach where the person can use an RGB+depth camera (e.g., Kinect) to show their grip by performing it. This seems a more natural way to express one's difficulty with objects. Further, the person's hand shape might also be captured by first gripping a piece of clay [15] and then digitalizing it. Although compelling, this type of vision-based approach has inherent issues, such as the imprecision of the sensors and the general difficulty in reconstructing the captured data into high-fidelity 3D models.

In my work thus far, either extending or adapting existing objects allows for incremental 'delta' for transforming the real world, as their functionality is somewhat dependent on the objects they are attached to. For a non-incremental approach, where brand-new objects are created from scratch, the objects we create will often end up physically interacting with others. A book-

shelf will be used to put on books. A chair will be sit on. A step stool will stood upon. How can facilitate the design process so that objects like these can function properly and reliably with real world objects they will be used with. My next chapter proposes a design tool to support users in conducting such fuctionally aware design tasks.

October 4, 2017
DRAFT

Chapter 5

User-Driven Structure Design to Support Real World Objects

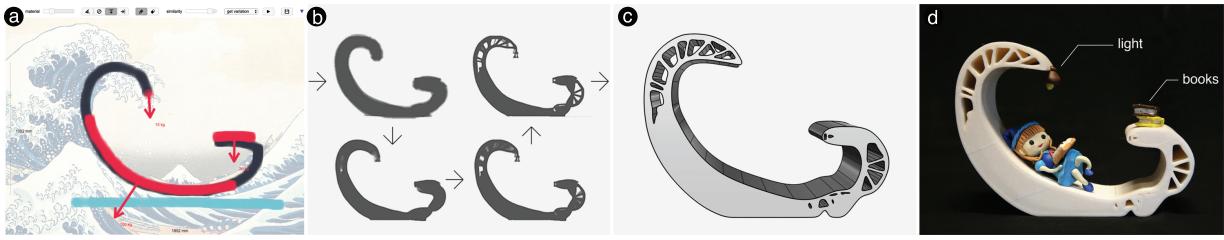


Figure 5.1: Forte is a 2D design tool that takes a user-driven approach of generative design: a user sketches a reading chair inspired by the painting ‘The Great Wave off Kanagawa’ (a) with specified loading scenario (red arrow forces, blue ground); Forte then generates structures (with real-time feedback) to support the loads while resembling the user’s sketch (b), which can then be post-processed to create a 3D fabrication-ready model (cd).

In both of the previous projects—Encore and Reprise, the design task is based on an existing object and the outcome is some sort of extension that enhances a specific function of that object. While this approach works for creating something incremental (e.g., adaptation), it becomes problematic when scaling up—creating self-contained objects such as bookshelf, step stool, or wine rack.

There are several challenges in designing this class of objects. Foremost, as they are no longer incremental add-ons to existing objects, it would be difficult to create a design *in situ* from these real world objects. Consider making a bookshelf for example. It is less straightforward if one has to start the design with some books; rather, a more intuitive approach is the reverse: create the bookshelf first, then see if it has enough space, or is strong enough to accommodate the books. However, for non-expert users, it is hard to create a functional design by intuition. A user might have an intuition about what their bookshelf should look like, and might be able to sketch the design; but they would probably have trouble making sure that this design would work—that is, supporting all the books to be put on it.

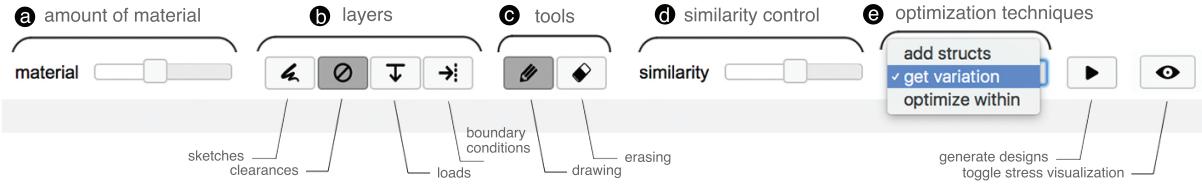


Figure 5.2: Overview of Forte’s default tool bar: adjusting the amount of material for generating the structures (a); different layers for sketching and specifying the loading scenario (b); drawing and erasing tools (c); controlling how the generated results are similar to the original sketch (d); and selecting different design optimization methods (e).

To ensure that a design meets such functional requirement, generative design methods such as *topology optimization* will automatically optimize material distribution given the constraints of space, material amount and other functional requirements [79]. Though promising, such methods provide very few ways for users to directly express or manipulate their designs; instead, users have to map their design ideas—often unintuitively—to mathematical input parameters. Further, it is often assumed that a CAD model has been designed prior to using such methods. As a result, there is a lack of support for users’ ideation process, and they have little direct control of the optimization’s outcome [112], or ways to effectively iterate their designs.

To address this problem, prior work and existing tools have been focusing on interactively specifying the input parameters [1, 29], adding template-based textures to the optimized results [49], or contextualizing them with CAD [78] or freeform sketching [41]. However, none of this work lets users directly control how a design should be generated and optimized.

To help address these issues, we present Forte, a sketch-based 2D design tool that takes a user-driven approach for people to directly express and iterate on their ideas through generative design with real-time visual feedback. Our main contribution is a technical approach that supports a high-level dialog allowing a simpler expression of users’ design intent, resulting in a combination of structural performance and less quantifiable objectives to be achieved in the results. Figure 6.1 shows an exemplar workflow: sketching a reading chair inspired by the famous painting ‘The Great Wave off Kanagawa’¹ with a specified loading scenario; Forte then generates structures to support the loads while resembling the user’s sketch, which can then be post-processed using external tools (e.g., Rhinoceros) to create a 3D fabrication-ready model.

Specifically, with Forte, users can ask the system to add structures to the original sketch, provide a variation with better performance, or generate optimized internal structures. Users can globally adjust a ‘similarity’ slider, which controls how much the system will ‘deviate’ from their initial input; they can also locally edit the design by adding or erasing parts of the sketch, which then interactively prompts the system to update based on this new information. Meanwhile, a visualization informs users of the performance trade-off as they iterate on the design. Building on a fast topology optimization engine [4], Forte enables rapid iteration and exploration, unlike most existing practices that run the process in a non-interactive, batch fashion (even for 2D designs).

¹https://en.wikipedia.org/wiki/The_Great_Wave_off_Kanagawa

We envision Forte will empower professionals who design structures, such as industrial designers, mechanical engineers and architects. To validate our approach, we held design sessions with 10 participants. Our study demonstrates that Forte empowers designers to create and explore a range of optimized designs with custom forms and styles. While our current focus is on a 2D design tool, we showcase three post-processing techniques using external tools to turn a 2D design to a 3D object: *extrusion*, *warping* and *combination*. We report stress analysis on these 3D models under varying loading conditions and finally demonstrate a series of fabricated examples.

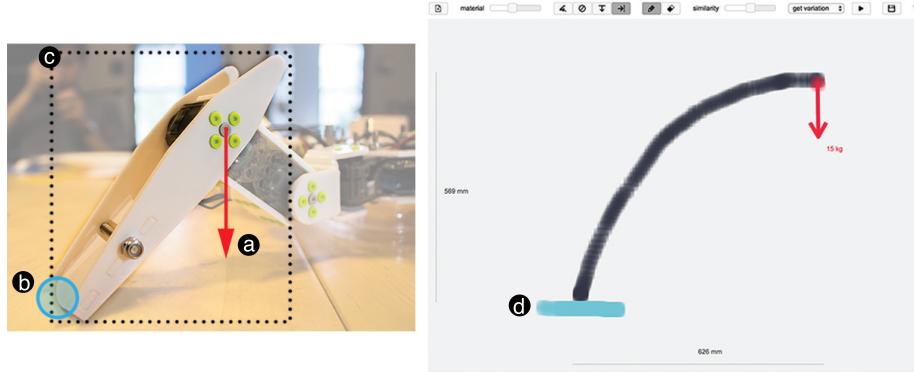


Figure 5.3: From users’ sketch, Forte (right) generates lightweight structures to support loads, such as a robot leg (left) that needs to support the robot’s weight using a specified amount of material.

5.1 Forte: User-Driven Generative Design

In this section, we describe users’ interaction with Forte and its key features for user-driven generative design, while leaving the technical details for later in the next section. Specifically, we will walk through a usage scenario: creating a 2D design of a leg for a quadruped robot (Figure 5.3). The design goal is three-fold: (i) in contact with the ground (Figure 5.3b), the leg needs to be optimized for supporting the weight of the robot’s body (Figure 5.3a), (ii) it needs to be lightweight to ensure the overall efficiency of the robot, and (iii) its form also needs to be customized with respect to other user-defined goals such as aesthetics. Below we first revisit some basic concepts and terminologies and then demonstrate how Forte allows users to jointly achieve these goals.

5.1.1 Basic Concepts and Terminologies

Generative design via topology optimization considers an object as a distribution of a certain amount of material in order to meet certain performance criteria. This paper focuses on structural performance, which is optimizing structures—with only a limited amount of material available—to support a given load. For instance, a robot leg should be designed to support the weight of

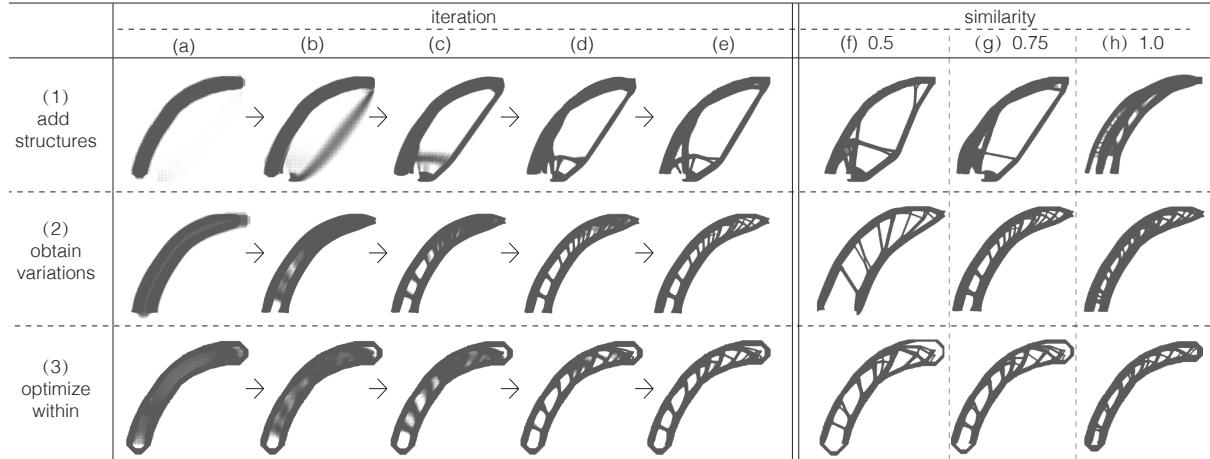


Figure 5.4: Forte’s three types of design optimization iteratively generate structures based on user’s sketch while providing them with real-time visual feedback akin to an animation (a-e), the result of which can be controlled by a global ‘similarity’ value (f-h).

the robot’s body with a lightweight structure. In Forte, a design consists of a sketch (considered below) and a **loading scenario**, which include (i) where the expected **loads** are and how large they are, and (ii) **boundary conditions**, which is where the object is in contact with its environment (e.g., a robot leg will touch the ground, as shown in Figure 5.3). Based on this input, Forte will then generate structures via different types of **design optimization**, i.e., distributing allocated material in a structurally optimized way. Each type of optimization is also constrained by a given **amount of material**, which is typically measured by the percentage of the entire **design domain**—the bounding box of the design object (Figure 5.3c). For example, 30% material means the material used for the robot leg amounts to 30% the size of its bounding box².

5.1.2 Sketching Designs and Loading Scenarios

With Forte, a user sketches a design idea as well as the loading scenario. Each type of sketch is drawn in a different color on its own layer (Figure 5.2b). This allows users to select and focus on one particular layer at a time. Once a layer is selected, it is pushed to the top with the rest of the layers set to semi-transparent.

To start, a user simply selects the design layer and sketches a curve representing the robot’s leg. With Forte, it suffices to provide such a simple representation that succinctly expresses the user’s high-level design intent, i.e., “I want the robot’s leg to curve like this”. As the user sketches the robot leg, Forte also shows the dimensions of the current sketch’s bounding box to inform users of the size of their design. The default scale is 20 mm per pixel, which can also be adjusted by the user in the preference settings.

As a next step, the user would consider the loading scenario. First, to create a load, the user

²We discuss later in the design sessions that participants found this to be a problematic way of measuring material, as it seems hard to get an intuitive estimate based on bounding boxes.

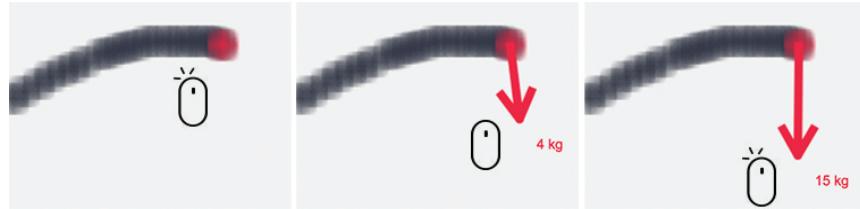


Figure 5.5: Interaction to specify a load.

switches to the load layer (Figure 5.2). They can click on the sketch to indicate a single loading point, or draw to specify a sequence of loading points. As shown in Figure 5.5, as soon as these points are drawn (e.g., mouse released), an arrow is rendered, following the cursor (e.g., where the mouse moves), and indicating the direction as well as the amount of load, which is also shown at the end of the arrow. A click confirms and pins down this load vector. To specify a boundary condition, the user simply draws it on the corresponding layer (Figure 5.3d).

In addition, Forte also supports a fourth type of layer that allows users to lasso-select a **clearance** region that needs to be kept clear, e.g., the space above a chair’s seat or around the tip of a hook (Figure 5.6). This information proactively prevents generating structures at places that will compromise the usage of an object.



Figure 5.6: Lasso-selecting an area of clearance around the tip of the hook (no structure generated here).

5.1.3 Design Optimization with Global Similarity Control

Once the user creates a sketch and specifies the loading scenario, they can now explore three different ways the system can generate structures by design optimization. Forte considers the generation of structures as a spectrum between users’ original idea and the system’s solution to the defined problem. Users can navigate on this spectrum by adjusting a *similarity* value from 0 to 1 (Figure 5.2d): conceptually, a low similarity would allow the system to ‘deviate’ more from the user’s initial sketch while a higher value will keep the generated structures closer to what the user originally draws. Further, Forte’s optimization algorithms run at an interactive rate, providing users with real-time visual feedback after each iteration, as demonstrated in the accompanying video figure. Below we illustrate Forte’s three new generative design optimizations and how similarity allows users to have direct control of the generated results.

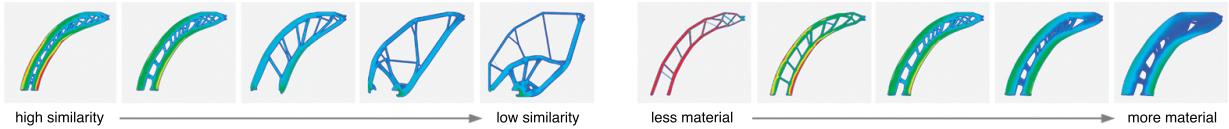


Figure 5.7: Stress visualization informs performance trade-off between different similarity values (left), and the amount of material (right).

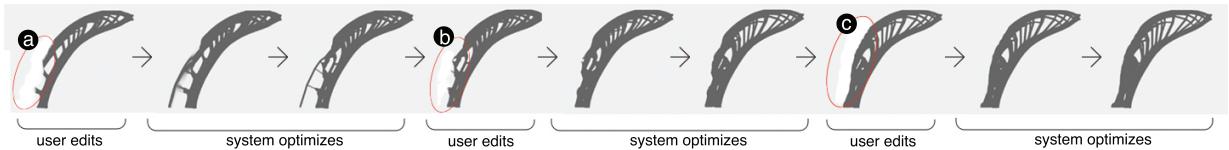


Figure 5.8: Users can suggestively erase part of a generated result (ab), which prompts the system to update the optimization addressing the intent of removing material; with this tool, users can also refine or smoothen a generated result (c).

Add structures To start, Forte can generate additional structures to strengthen the user’s original sketch. For example, as shown in Figure 5.4(1a-1e), a major truss structure is added parallel to the user-sketched robot leg as reinforcement.

The similarity value effectively controls how far away from the original sketch these structures can be added. As the user increases similarity, added structures will appear increasingly closer to the original sketch and eventually become part of it, equivalent to a thickening operation (Figure 5.4(1h)). However, note that the added structures’ position does not change smoothly with the similarity value; rather it tends to ‘snap’ to positions along the way that are structurally meaningful.

Overall, ‘add structures’ is targeted at situations where users only have a partial or incomplete design idea in mind and are unsure whether and how they should add extra structures to provide support for the loads.

Obtain variation Forte provides a second optimization algorithm that can generate variations of the initial sketch. For example, as shown in Figure 5.4(2a-2e), Forte generates two parallel interconnected trusses instead of one user-sketched structure; however, the variation still bears resemblance to the original sketch as it globally follows the curve path specified by the user. As the user increases the similarity value, the generated structures will follow the user’s sketch more closely (Figure 5.7). ‘Obtain variation’ works best when users only have a concept (e.g., a skeleton) of what the design should look like without any structural details, in which case the system will take the initial designs as ‘seeds’ and vary it with generated structures.

Optimize within Forte can also consider the user’s sketch as solid shapes (rather than skeletons) and optimize the internal structures within. For example, as shown in Figure 5.4(3a-3e), Forte reproduces the contour of the user-sketched robot leg while filling in generated structures. By controlling similarity, the user can effectively shrink or expand the contour, e.g., decreasing similarity results in a robot leg of the same curvy shape but much thicker than the original sketch, given the system more space to generate structures. ‘Optimize within’ serves well for

cases where users already have a fully-designed solid shape but want to re-design the interior structures.

For any of these design optimizations, users can adjust how much material can be used in generating the design. For example, as shown in Figure 5.7, reducing the material amount often creates hollow structures with trusses, while a higher value tends to thicken or fill up certain parts of the design.

As the user runs one of the three optimizations, Forte creates and records a trial, and together displays a history of results from all these trials. Selecting a trial not only shows the resulted structures, but also updates the menu so that users can see which of the three optimizations was used, how much material was allocated and what similarity value was applied.

5.1.4 Refining Optimization with Local Suggestive Edits

The similarity value allows users to control the optimization outcome at a global level with respect to the entire design. Complementarily, once a result is generated, users can also perform fine local editing, which will prompt the system to re-run the optimization, taking into account users' underlying intent of the edit. For example, if a user wants a thinner tip of the robot leg, they can use the eraser tool (Figure 5.2c) to scrub that area (Figure 5.8a, shadowed part). The system immediately starts a new optimization and returns an updated result with less material on the tip of the leg. Increasing the erasing 'power' and applying it again finally sharpens the leg, but results in a rough outline (Figure 5.8b). We can also use this tool to smooth the optimized result (Figure 5.8c).

Similarly, the user can also use the pen tool (Figure 5.2c) to draw on an optimization result, such as adding a truss to the leg (Figure 5.9ab). The system will also re-run the optimization and will attempt to generate some structures where the new truss is drawn (Figure 5.9).

This technique is called suggestive editing, as users' action do not directly remove or add to the design; rather, it serves to suggest their intent to the system, e.g., "I want less material here", or "I would like to add something there". The next round of structure generation will take into account such intent and attempt to realize it through the optimization process. However, intent cannot be realized if the removal or addition of material compromises the structural integrity of the design (e.g., removing the entire bottom part of the robot leg, making it unable to stand on the ground).



Figure 5.9: Users can suggestively add material to a generated result (circled strokes), which prompts the system to re-run the optimization and attempt to add structures accordingly.

5.1.5 Informing Performance Trade-offs with Visualization

As users exploratively customize the optimization process, it is important to inform them of the performance trade-offs. Forte allows users to toggle a stress visualization (Figure 5.2) overlaying each optimized result (Figure 5.7). The heat map indicates which part of a design is the weakest; it also allows users to compare different designs' performance and see how the stress situations evolve as the design evolves. Users can also adjust the safety factor: a higher value is equivalent to assuming more weight, thus revealing more structurally problematic areas in the design.

5.2 Implementation

In this section we describe technical details of Forte's implementation.

5.2.1 Sketching Designs and Loading Scenarios

Recall that users' design input consists of a sketch within a design domain as well as loads, boundary conditions (Figure 5.3), and/or clearance (Figure 5.6).

Given these input parameters, first the design domain is discretized into $N = W \times H$ square **finite elements** [113], each of which (denoted as e) is assigned a **density** value $x_e \in [0, 1]$. The user's initial sketch is discretized into a binary density map on this domain with 1's for elements overlapping with the sketch and 0's for the others. Similarly, loads, boundary conditions and clearance are also converted to such density maps. In addition, for loads each density=1 element is also associated with a load vector indicating the amount of the load as well as the direction.

5.2.2 Design Optimizations with Global Similarity Control

The implementations of the three design optimization techniques presented here are developed based on the 88-line program [4], which uses a SIMP (Solid Isotropic Microstructure with Penalization) method for topology optimization. For readers to better understand our approach, we first provide a brief of overview and refer the reader to Andreassen et al. for further details [4].

Overall, the process solves the following compliance optimization problem:

$$\min_x : c(\mathbf{x}) = \mathbf{U}^T \mathbf{K} \mathbf{U} = \sum_{e=1}^N E_e(x_e) \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e, \quad (5.1)$$

subject to:

$$V(\mathbf{x})/V_0 \leq f; \quad \mathbf{K} \mathbf{U} = \mathbf{F}; \quad 0 \leq x_e \leq 1$$

where c is the compliance of the design, which is computed with \mathbf{U} – the global displacement matrix, and \mathbf{K} – the global stiffness matrix determined by the stiffness of the material as well as its distribution (or layout). The objective function is further broken down to the summation of compliances across all elements: E_e is the Young's modulus of e ($E_e = E_{min} + x_e^p(E_0 - E_{min})$), \mathbf{k}_0 is the unit stiffness matrix which together with E_e determines the stiffness of element e ; \mathbf{u}_e is the

displacement vector of e . Finally, \mathbf{U} can be computed from $\mathbf{KU} = \mathbf{F}$, where \mathbf{F} is vector of forces (loads).

At each iteration, once c is computed, x_e is updated as follows:

$$x_e^{new} = \begin{cases} \max(0, x_e - m), & \text{if } x_e B_e^\eta \leq \max(0, x_e - m) \\ \min(1, x_e + m), & \text{if } x_e B_e^\eta \geq \min(1, x_e - m) \\ x_e B_e^\eta & \text{otherwise} \end{cases} \quad (5.2)$$

where m is a pre-set step for density changes, $\eta = 1/2$ is a numerical damping coefficient, and $B_e = (-\frac{\partial c}{\partial x_e})/(\lambda \frac{\partial V}{\partial x_e})$, with λ being a constraint factor to ensure the total volume does not exceed $V_0 f$ —the allocated amount of material where f is the percentage and V_0 is the volume of the design domain.

This process represents the classic topology optimization approach that automates the generation of structures in a ‘black box’. Below we describe our methods of extending it to a user-driven approach, allowing the control of the optimization results via a similarity value.

Add structures We initialize the design domain by assigning each element the value of f (e.g., 15% of the bounding box area). Through iterations the optimization will increase or decrease this value for a given element, thus forming a global structure. After each iteration, we add the intermediate generated structures to the user’s sketch, and take it as the input for the next iteration. Specifically, now each element density is computed as

$$\widehat{x}_e^{new} = \begin{cases} 1, & \text{if } e \in \text{user sketch} \\ x_e^{new} & \text{otherwise} \end{cases} \quad (5.3)$$

To control similarity, we perform a *mass transport* (cf. [13]) after each iteration before adding the intermediate results to the user’s sketch. A mass transport considers two designs (e.g., user sketch vs. optimized result) as two ways of distributing the same amount of mass (computed by summing up elements’ density), and transports the mass in between the two distributions to obtain an interpolation. We use mass transport to interpolate between the optimized result and the user sketch, taking similarity as the weight of interpolation (0: user sketch — 1: optimized result). This allows us to control how much the design will be similar to the original sketch after adding the optimized result.

We implement the mass transport function by computing and interpolating the barycenters of the two input designs as distributions. We compute the distance field of the user sketch and use it as the input distribution to the mass transport function. This provides more fine-grained information at each element (what is the distance of this element to the user sketch) beyond a binary value (whether or not it is part of the user sketch).

Obtain variation Instead of f , we initialize the design domain with the user’s sketch, and then run topology optimization, which will effectively ‘shift’ the material away from user-specified locations to achieve a variation with optimized material layout.

To support user control of similarity, we map the distance field of the user’s sketch to the input design domain as

$$x_e^{initial} = \frac{1}{\sum_{i=1}^N \cos(\frac{\pi d(i)}{2})^s} \cos(\frac{\pi d(e)}{2})^s f \quad (5.4)$$

where $x_e^{initial}$ is the initial material density at element e , d is the normalized distance field function, s is the user specified similarity request, and the \cos function raised to s produces value close to 1 when $d(e)$ is small (close to the user’s sketch) and drops rapidly to 0 when shifting away. s controls how quickly such a drop occurs. Thus a larger similarity value will initialize the optimization (hence the result as well) more closely to the user’s sketch, while a smaller value will loosen the constraint by ‘diluting’ the user’s sketch, and produces a result with wilder variation.

Optimize within We consider the user’s sketch as a ‘skeleton’. As a first step, we expand it using the distance field by setting a new contour around the original sketch. We then run the optimization only within the contour, as follows, at each iteration,

$$\widehat{x_e^{new}} = \begin{cases} x_e^{new}, & \text{if } d(e) < d_c \\ 1, & \text{if } d_c \leq d(e) < d_c + t \\ 0 & \text{otherwise} \end{cases} \quad (5.5)$$

d_c is a distance field value used to define the new contour, and t controls the thickness of the contour. We control similarity by shifting c towards or away from the user’s original sketch.

5.2.3 Refining Optimization with Local Suggestive Edits

Forte enables users to interactively suggest adding to, or removing from, parts of the optimization result, which then prompts the system to re-run the optimization to produce a version that matches the users’ intent, including the new edits. Importantly, in such re-run, the system is initialized based on the result from the previous run, rather than starting from the original sketch.

As a user draws on, or erases, parts of a design on a $W \times H$ domain, we create a mask M of the same dimensions with each pixel’s value $M_e \in [1 - \Delta_{remove}, 1 + \Delta_{add}]$, $\Delta_{remove} \in (0, 1)$, $\Delta_{add} > 0$. Then in the optimization process, we update element e as

$$\widehat{x_e^{new}} = M_e x_e^{new} \quad (5.6)$$

In essence, M_e indicates whether and how much a user would want to add or remove material at this point. If a user performs an erasing action, M_e is set to be in $[1 - \Delta_{remove}, 1]$, which will dampen the to-erase elements’ density at each iteration; if a user performs additional drawing, we integrate the addition to the result as input for the re-run. As M_e is now set to be in $(1, 1 + \Delta_{add}]$, it will amplify the to-add elements’ density at each iteration. All unedited M_e ’s will be set to 1.

5.2.4 Informing Performance Trade-offs with Visualization

We compute the von Mises stress of each optimized result using Biyikli and To’s method [12], which is implemented as an integral component of the optimization process. The values in the stress distribution are then normalized by a material-specific yield stress σ_y and visualized as a heat map. All our examples and demonstrations in the design tool use a material model based on an Extrusion Grade PLA [50]. The yield stress is also divided by the user-specified safety factor, e.g., a $2\times$ safety factor will result in the use of $\sigma_y/2$, thus visually revealing more potentially problematic area in the design.

5.2.5 Software and Hardware

The front end of Forte is written in JavaScript using jQuery³ for UI development. The back end consists of two components: *(i)* the optimization techniques were written in MATLAB, which is then compiled into a standalone service; and *(ii)* a python-based server which relays optimization requests and input parameters from the front end to the MATLAB service, and sends back the results. Everything runs on a MacBook Pro (Retina, 15-inch, Early 2013) with a 2.4 GHz Intel Core i7 and 8 GB 1600 MHz DDR3 memory. In our demonstrations, the front end runs on a Google Chrome web browser.

5.3 Design Sessions

To validate Forte, we conducted informal, qualitative design sessions with 10 participants. The objective of the study is to let participants create their own designs using Forte's user-driven generative approach, and elicit their initial reaction and feedback to the system to reveal existing problems and to inform future improvements.

5.3.1 Participants

We recruited 10 participants from our university (1 female, aged 18-33). Two majored in interaction design, three in architecture and the others in mechanical engineering. All participants reported to have sketching experience and all had experience using CAD tools. Two reported to be knowledgeable about topology optimization, four had passing knowledge and the others had none.

5.3.2 Tasks & Procedure

The design sessions took place in our research lab and lasted between 1 and 1.5 hours for each participant. To begin, participants were given a tutorial by the experimenter on some basic understanding of topology optimization, and how to use Forte by walking through a concrete example of designing a bracket. Next, they were asked to use Forte to create objects of their own design, as well as coming up with loading scenarios. For each design, they were free to choose any of the three optimization techniques and run as many trials as they would like. Participants created 2-4 designs during the course of the study.

Throughout the design session, participants were asked to think aloud as the experimenter took notes. They completed a questionnaire to summarize their experience with Forte, e.g., what they like about the tool, what problems they encountered, and what new features they would like to have in the future.

³jQuery: <https://jquery.com/>

5.3.3 Feedback and Discussion

We analyzed the qualitative data using a method akin to the Affinity Diagram approach [11]. We organized notes of participants' comments and survey response to iteratively develop meaningful and coherent themes.

Overall, participants responded positively to Forte's user-driven generative design approach: "I like how it can produce interesting shapes, i.e., organic, asymmetric shapes which are difficult/time-consuming to hand-draw" (P1); "... pretty nice about making the idea of topology optimization accessible to people" (P5); "computer generating structures from my sketch is interesting; it might be the way computers can design something like humans do in the future" (P10).

Optimization techniques Participants were able to learn and recognize the difference between the three optimization techniques. Further, some developed their own understanding and preferences of the techniques. For example, P1 mentioned that "'add structures' lead to traditional shapes; 'obtain variation' is more creative." P4 liked to use 'add structures' as he wanted "the system to add something different from the original drawing". P9 extensively used 'optimize within' as he wanted to keep the exact shape of the drawing and only optimize what is inside.

Similarity control Participants had no trouble understanding or using the similarity control. P1 commented that "... it allows me to control how much imagination I need". P4 liked that he could control how different the result will look from the original design. P7 summarized the experience of controlling similarity as having "design freedom", and felt he "can design anything and [Forte] will keep it to some extent [after optimizations]".

Real-time feedback Participants found it pleasant and useful to see how the design evolved in real time: "[I] enjoy watching the shape evolving" (P1), "Watching it grow and iterate is really nice—it helps you visualize stresses and know where the strength needs to be" (P2), "It helps a lot to see how things are branching out, happening in the background" (P7).

Local suggestive editing We observed only three participants frequently used local suggestive editing. They all appreciated this capability of refining the optimized design: "... very easy to make quick changes on the fly to see how overall structures change" (P2); "... add and subtract material dynamically which was very helpful" (P7). P10 would continuously use this feature in a sequence of trials to add more reinforcement based on the stress visualization.

Existing problems and suggested features Participants also pointed out existing problems and suggested new features in the tool. One main confusion is the amount of material: P2, P4 and P5 thought it was relative to the sketch, whereas it was actually relative to the sketch's bounding box. P5 also suggested visualizing the actual size of allocated material to give users a more concrete idea. Four participants (P2, P4, P7 and P10) asked how they can 'transfer' an optimized result to the original drawing, or simply just be able run technique B on technique A's result. Such request points to an interesting problem for future work, which is enabling users to manage how an initial design branches out to a complex tree structure. Finally, participants proposed other controls akin to the similarity variable, e.g., keeping the design balanced (P1), controlling the global maximum thickness of the generated trusses (P5), and specifying symmetric structures (P6, P7), all of which suggest fertile opportunities for future work.

5.3.4 Designs Created by Participants

Figure 5.10 shows a series of designs created by the participants. Overall, we observe two recurring approaches of creating these designs. First, some users would take a ‘form-first’ approach, whereby they started by drawing their imagined form and then drove the system to generate structures similar to the form. For example, ‘public bench’ by P1 (Figure 5.10a) started with a single curve as a shared sitting space, while the system generated structures underneath to support it. ‘Cross-legged book case’ by P6 (Figure 5.10e) started with a form that almost looked like a Small Seal Script⁴ character in ancient Chinese. In contrast, other users would take a ‘structure-first’ approach, starting with a basic structure based on the object’s functionality; they explored which of the system’s generated structures can also create interesting forms to the design. For example, ‘eagle crane’ by P2 (Figure 5.10f) was initially drawn as a simple polygon that provides the functionality of a crane. However, as the system ‘morphed’ the input structure, it evolved into an eagle head shaped design. ‘House’ by P8 also started with simple geometry representing a two-story house, a loft and a patio, which then evolved into two distinct design variations (Figure 5.10 g and h, using ‘obtain variation’ and ‘add structures’, respectively).

We also observed different iteration strategies. Most users iterated a design by playing with different similarity values and amounts of material. For example, P3 used Forte to design (the cross-section of) the GE Aircraft Engine Bracket (Figure 5.10c). His first three trials were simply trying out different material-similarity combination to gauge variety of the underlying design space. Finally, quite a few users modified their initial design based on the system’s generated structures. For example, in designing a bike frame (Figure 5.10b), P2 noticed that the system did not keep one of the trusses he drew but instead created a different one. P2 then modified the initial design to a ‘imitate’ the system’s approach. Participants also tried to modify the system’s design. As mentioned above, three participants were observed actively using the local editing techniques. For example, In designing a bow P7 used the eraser tool to iteratively refine the shape of the bow updated by the system ((Figure 5.10d).

Figure 5.10 shows other designs created by our participants.

5.4 Post-processing and Fabricated Results

Forte enables users to formulate, express and optimize their design in a 2D environment. To fabricate their designs into 3D artifacts, we describe three post-processing techniques using external tools (e.g., the Rhinoceros⁵ CAD system) for fabrication and showcase some of the fabricated results.

Extrusion We can extrude a 2D design along a path to create a 3D object. First we convert a generated design from a bitmap to a vector graphic, which can already be used in a laser cutter to create planar objects, such as the legs for the quadruped robot (Figure 5.11a). Further we can also import the vector graphics to a 3D modeling tool for creating a 2D surface and extruding it to a 3D object. We use this technique to create a multi-functional reading chair that consists of an overhead light, a reclined sitting area, and bookshelves (Figure 5.11f). Extrusion is not limited

⁴Small Seal Script. https://en.wikipedia.org/wiki/Small_Seal_Script

⁵<https://www.rhino3d.com/>

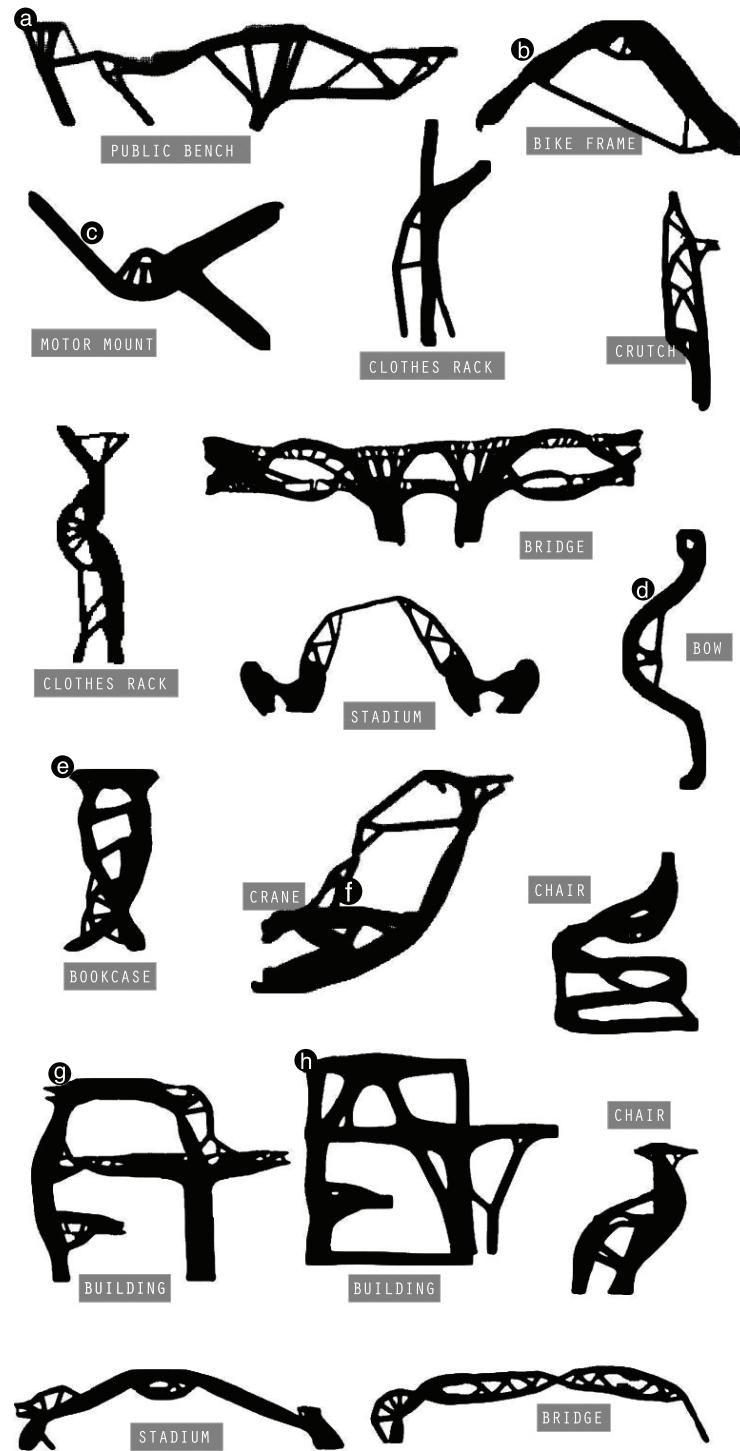


Figure 5.10: Designs created by 10 participants using Forte.

to a perpendicular path: for example, it is possible to extrude the 2D chair profile in Figure 5.11c along a circular path to create a bench.

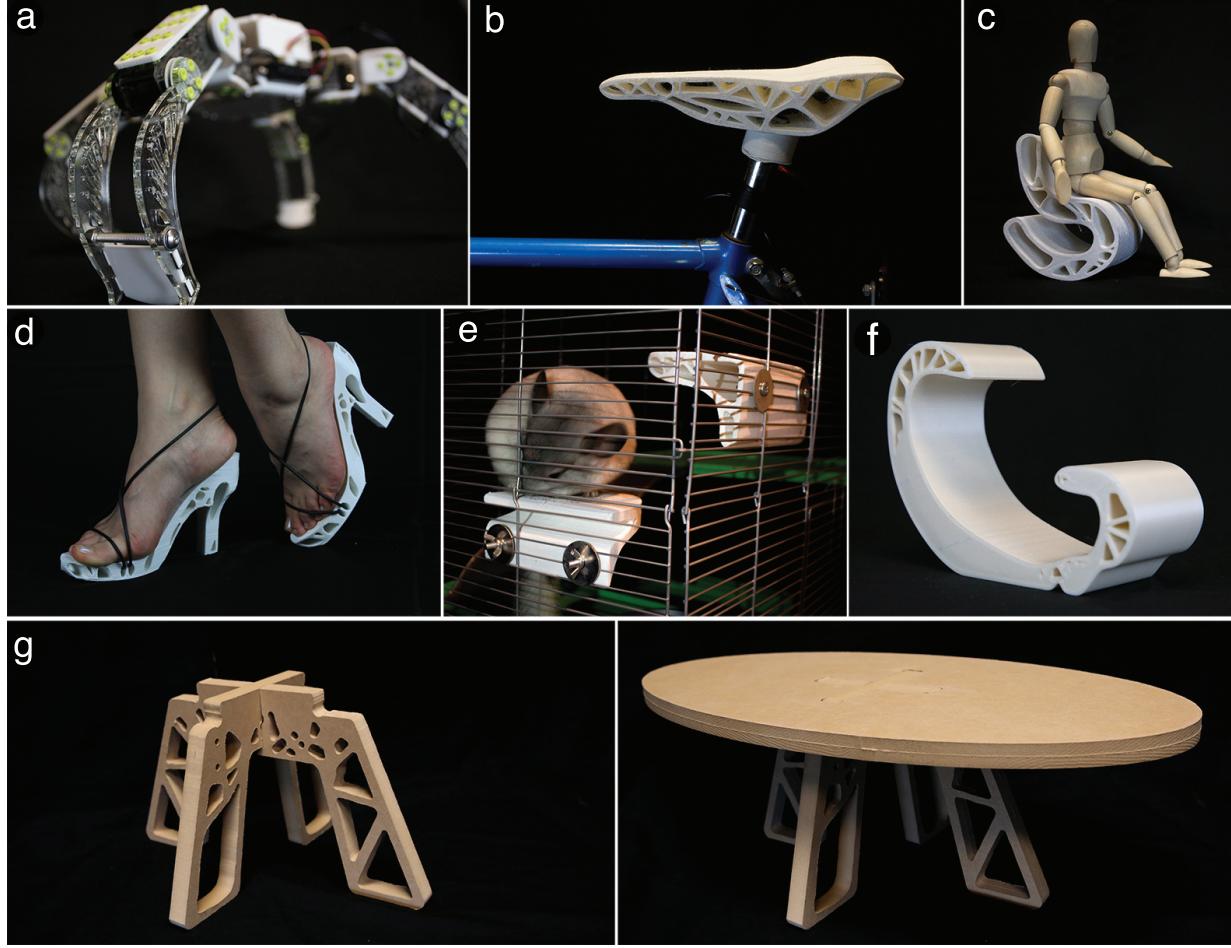


Figure 5.11: Fabricated examples designed from Forte’s generated structures: legs for a quadruped robot (a); a bike seat (b); an S-shaped chair (c); high heels (d); pet jumping platforms (e); a reading chair (f); and a tea table (g).

Warping We can also warp a 2D design into a 3D surface. We used this technique to design a pair of high heel shoes using generated structures as the base (Figure 5.11d). We designed a jumping platform for chinchillas by warping the cross-section to form a rounder top (Figure 5.11e). We designed a bike seat by warping a 2D generated pattern to the shape of a cushion (Figure 5.11b).

Combination We can join multiple planar components made from the 2D designs (e.g., fabricated using laser or CNC cutting) to create 3D structures. For example, we designed a tea table supported by two joined structures (Figure 5.11g).

To test the structural integrity of the 3D models produced in the post-processing step, we performed Finite Element Analysis (FEA) using Autodesk Inventor [37]. As shown in Figure 5.12, we tested each model on a range of increasing loads—from normal to overloading—and found

consistent stress distribution across all the cases with high structural performance⁶. As our current focus is on the 2D design tool, this test is only based on our current examples (as well as specific types of material chosen for running FEA). Further experiments are required in order to validate the general process of converting 2D structural patterns to 3D objects manufactured in a wide range of materials, which we leave as future work.

5.5 Limitations, Trade-offs and Future Work

We discuss limitations and trade-offs to better understand our current system, as well as pointing to potential future work.

Going from 2D to 3D Forte is currently a 2D oriented tool, although our examples have shown three ways of extending it to 3D: extrusion, warping and combination. The implementation of Forte is also extensible to 3D: on the front end, prior work has demonstrated numerous techniques for enabling sketching in 3D [6, 7, 36]; on the back end, the optimization can also handle 3D cases by adding a third dimension to the input parameters (design domain, loads, boundary conditions, etc.) as well as the material stiffness matrix. Perhaps the real challenge of going 3D is maintaining real-time feedback and interactive iteration. Past work has explored engineering solutions for handling 3D generative designs. For example, Aage et al. provide a topology optimization implementation based on PETSc (Portable and Extendable Toolkit for Scientific Computing), which enables incredibly fine discretization, e.g., a (3D) design with 27.6 million design elements running on 24 CPUs (144 cores) only requires 30-60s per iteration [2]. Building upon all this prior work, our next step will develop and extend Forte to a fully 3D design tool.

Ambiguity and imprecision of sketching As mentioned in Gross and Do’s work, sketching is a particularly appealing medium for early ideation and design formation, as it embraces ambiguity and imagination. As discussed in the design session, this is also a strength of Forte, as it allowed participants to use unique hand-drawn forms to generate structures. However, as pointed out by a few participants, Forte lacks support for more precise design, e.g., straight lines, smooth curves, and symmetry. In the future versions of Forte, we would like to add more controlled ways of specifying designs.

Generating and exploring a large space of examples Currently each of Forte’s optimized result is initiated and driven by the users. While this gives users freedom to explore and customize each optimization trial, it becomes a somewhat tedious process as the number of trial increases. In the future we plan to explore ways to automatically generate a large design space of structures from a single user input. To realize this, the challenge is two-fold: *(i)* how to infer and sample more variables from one single user input in order to generate more than one result; *(ii)* how to enable users to efficiently navigate and filter a large number of results. Our future work will explore solutions to tackle these two challenges.

Enabling more semantic controls of design optimization On the input side, we are interested in enabling more semantic controls of design optimization beyond the similarity value. For example, in Yumer et al.’s work [110] they developed a number of semantic metrics for 3D

⁶We define the range of loads based on the usage scenarios of each design. The heatmap visualization is based on the maximum stress amongst all five loads.

models by having the crowd perform pair-wise comparison on a large data set. In the future, we want to let users label and quantify each optimization result, e.g., ‘how organic is this bike frame’ and ‘how thin or muscular is this chair’. By collecting these quantified labels, we hope to map the space of the input parameters (e.g., amount of material, similarity, clearance) to the semantic space of the optimization outcome. This can potentially enable the ‘transfer’ of design, e.g., applying a set of input parameters from an organic-looking bike frame design to turn this chair into a similar style.

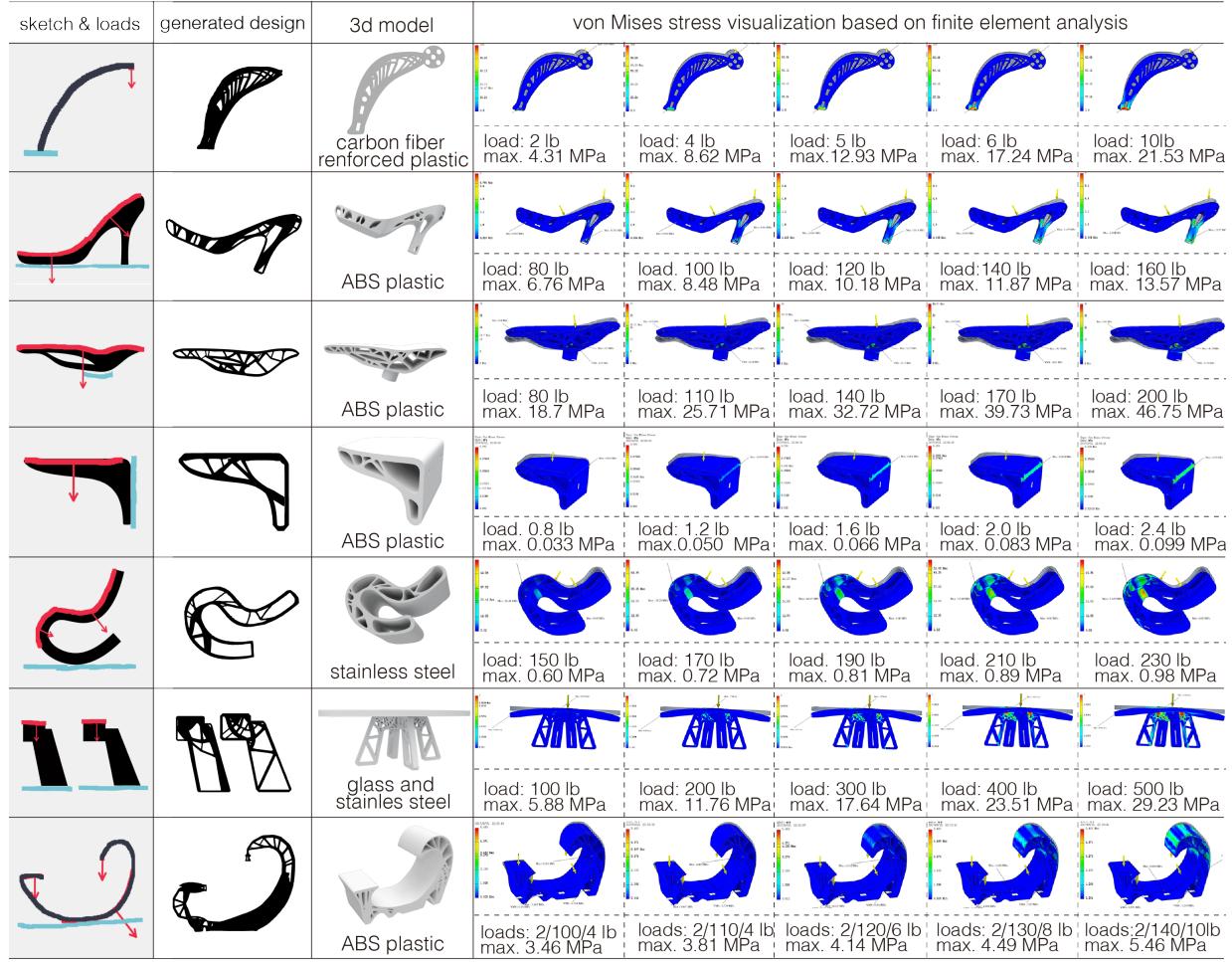


Figure 5.12: All our fabricated examples from the sketches in Forte, to generated 2D designs, and to post-processed 3D models; we performed a Finite Element Analysis on each 3D model based on the input loading scenario, which shows consistent stress distribution across a range of increasing loads.

October 4, 2017
DRAFT

Chapter 6

Embedding Everyday Objects to Augment 3D Prints

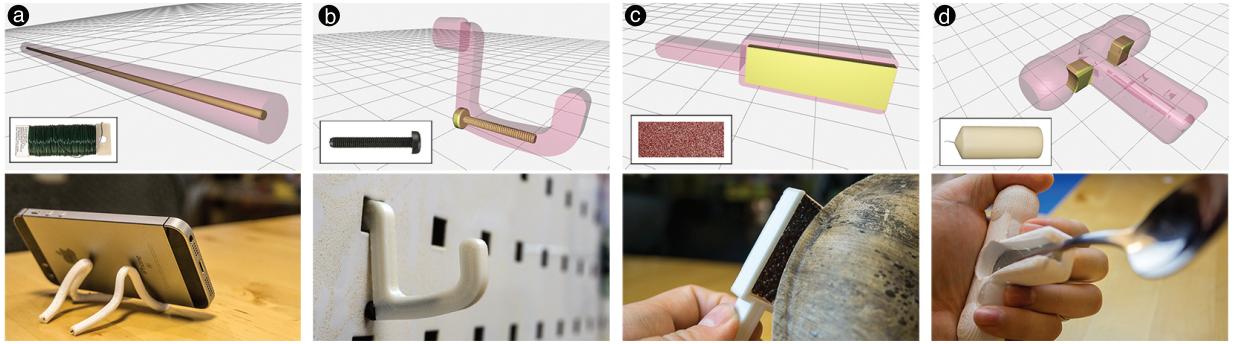


Figure 6.1: We present a library of *embeddables*—everyday objects that can be cut, worked and embedded into 3D printable designs to augment their material properties. For example, inserting a florists’ wire makes elastic TPU-based objects stay in shape (plastic deformation) (a); a screw is used to reinforce a pegboard hook (b); a piece of sand paper becomes the surface of a printed sander (c); the malleability of wax makes it possible to further personalize assistive devices even after they are printed (d).

The previous three projects focus on augmenting existing real-world objects with 3D printed designs. In this chapter, I flip the relationship and explore how real-world objects can be used to augment 3D printed designs. In our everyday life, we are surrounded by objects with a wide range of material properties. Consider gloves as an example. They can be made of leather for insulation against cold (Figure 6.2a), or silicone for protection against heat (Figure 6.2b), or nitrile for elasticity (Figure 6.2c), or rubber for better grip and abrasion resistance (Figure 6.2d), or even infused with copper to provide support for hands and joints (Figure 6.2e).

In contrast, despite the rapid development in multi-material printing, personal fabrication machines (e.g., desktop 3D printers) currently only support a much smaller set of materials. This limits people’s creativity in exploring how their designs might incorporate and benefit from various material properties. It also limits the application of fabricated objects, preventing them

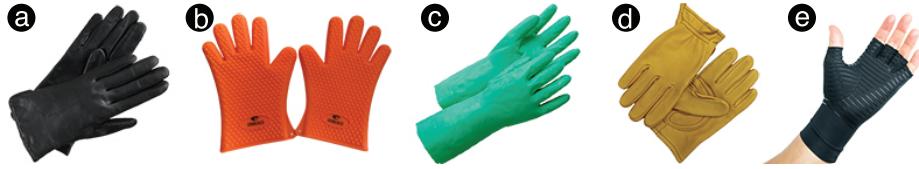


Figure 6.2: Everyday objects are made of various materials, e.g., for gloves, we have leather (a), silicone (b), nitrile (c), rubber (d) and even infused copper (e).

from being adopted for a wider context of usage.

To overcome the current limitations of material variety, one approach is to design *metamaterials*—a composition of material(s) that exhibits behaviors beyond their natural properties by means of carefully constructed spatial variations in the material(s). Prior work has explored metamaterial designs for elasticity [57, 77], acoustics [52], and mechanisms [38]. To incorporate an even larger set of properties, another approach is to embed alternate materials, such as conductive paint for interactivity [71] or state-changeable material for custom deformability [30]. Though promising, prior work tends to focus on one or a few specific properties, rather than an approach that affords the exploration of a holistic set.

The goal of our research is to leverage a repertoire of material richness in everyday objects to enable the exploration of various material properties in people’s custom designs. To achieve this, we develop a library of *embeddables*—everyday objects that can be cut, worked and embedded into 3D printable designs. For example, as shown in Figure 6.1a, by inserting a florists’ wire we can make an elastic object’s stay in shape (plastic deformation), such as making a phone holder from a ‘noodle’ printed in TPU (Thermoplastic Polyurethane); embedding an M3 20mm screw strengthens a pegboard hook along its overhanging structure (Figure 6.1b); embedding a piece of sand paper adds roughness and creates a simple sander (Figure 6.1c); embeddables can also be used for assistive devices: embedding wax into an a spoon handle¹ allows users to sand it to a custom shape where the fingers can grip comfortably (Figure 6.1d).

To enable the use of these embeddables into fabricated designs, we first contribute a systematic categorization of both their geometric and material properties. Specifically, we construct a design space (Figure 6.3) composed of two dimensions: in terms of geometry, we define *degree of workability* (DOW) to characterize how different embeddables can be worked into user-defined shapes; in terms of material, we introduce an extensible taxonomy of material properties relevant to personal fabrication. Combined, these two dimensions produce a large space of examples making use of embeddables in various usage scenarios. Further, it also serves to unify a number of projects from prior work.

Building upon the design space, we then contribute Medley—a tool for the computational design and fabrication of embeddables into users’ custom designs. With Medley, users can import a 3D model, search for embeddables with desired material properties, and interactively edit and integrate their geometry to fit into the original design. Further, Medley also supports the fabrication and embedding process, including instructions for carving or cutting the objects, and generating optimal paths for inserting them.

¹Adopted from ‘The Andrew Fork’: <https://www.thingiverse.com/thing:1816581>

6.0.1 Contribution

To validate the expressiveness of our approach, we showcase a series of examples augmented by embeddables that go beyond the objects' original printed materials. Our main contribution is a library with computational support for the exploration of material properties to augment personal fabrication, closing the gap between its current limitations and the future of multi-material printing.

In the remainder of this paper, we first review related work. Next, we present the design space underpinning the construction of our embeddable library as well as the computational design process. We then demonstrate a wide range of printed results, and describe the technical details, which include specifying the geometry of embeddables, finding optimal insertion paths, and generating instructions and templates to guide the embedding process. We close by discussing current limitations and future work.

6.1 Design Space

To design and fabricate objects with embeddables, it is important to understand (*i*) material: what properties are available and how to find them in everyday objects, and (*ii*) geometric workability: how to work (e.g., cutting, carving) embeddables into specific geometry that fits into certain parts of the design object, which needs to be planned at design time as well as executed in the fabrication process. To structure our understanding of these two questions, we construct a design space as shown in Figure 6.3.

The rows of the design space correspond to embeddables' different material properties. These properties are collected from a survey of related work on embedding elements into fabricated objects (shown in italics in Figure 6.3) as well as literature in material science [39]. Our current focus is on a set of material properties that can be found in common everyday objects, and are relevant to personal fabrication (i.e., have been explored in prior work, or can lead to new design directions). Our current list of properties is by no means exhaustive; rather they are meant to be extensible and more properties can be included under this framework.

The columns of the design space describe the embeddables' *degree of workability* (DOW)—how they can be worked into user-defined shape. Notice that DOW is different from the dimension of an embeddable's geometry. For example, as shown below, a wire has DOW 1 (cut to length) yet it can be bent into 2D or 3D geometry.

- *DOW = 0* are fixed-shaped, atomic objects that will be directly embedded into a design. For example as shown in Figure 6.1b, a screw is embedded into a peg board hook to increase its flexural strength.
- *DOW = 1* are wires, tubes or rods of material that can be cut to certain length to create embeddables. Length is the 1 DOW. For example, Figure 6.1a shows a florists' wire cut to length and inserted into a TPU-based 'noodle' to enable plastic deformation.
- *DOW = 2* are patches, sheets or plates of material that can be cut to certain area to create embeddables, thus having 2 DOW on the X/Y plane. The other dimension, thickness, is fixed. For example, a piece of sand paper is cut to replace certain area in order to create a printed sander (Figure 6.1c).

WORKABILITY MATERIAL PROPERTIES		NON-WORKABLE	WORKABLE		
		DEGREE 0 fixed-shaped, atomic objects	DEGREE 1 wires, tubes, rods, etc.	DEGREE 2 patches, sheets, plates, etc.	DEGREE 3 bars, blocks, lumps, etc.
MECHANICAL STRENGTH	DENSITY				
	SOFTNESS			<i>Embedding fabric Rivera et al. [12]</i>	
	ROUGHNESS				
	TENSILE				
	COMPRESSION				
	FLEXUAL				
	ELASTICITY				
	FLEXIBILITY			<i>Hotflex Groeger et al. [3]</i>	
	PLASTICITY				
	DURABILITY				
CONTACT	MACHINABILITY	N/A by definition			
	WEAR RESISTANCE				
ACOUSTIC			<i>Accousturments Laput et al. [6]</i>		
ELECTRICAL		<i>Embedded electronics e.g., Voxel8</i>	<i>PipeDream Savage et al. [13]</i>	<i>Capricate Schmitz et al. [14]</i>	<i>Capricate Schmitz et al. [14]</i>
OPTICAL			<i>Printed optics Willis et al. [25]</i>		
THERMAL					

Figure 6.3: A design space characterizes the material properties of embeddables, as well as their *degree of workability* (DOW)—whether they are embedded as fixed-shape objects (DOW=0), or can be cut to length (1), to a fixed-thickness profile (2), or to a customized 3D form (3).

- $DOW = 3$ are bars, blocks or lumps of material that can be cut to arbitrary 3D shape within their size limits. For example, by carving out two pieces of finger-sized wax we can embed them to customize the handle of an assistive utensil holder (Figure 6.1d).

Each cell in Figure 6.3 contains one example—either a piece of related work or an embeddable—with the specific material property and DOW. Admittedly each of these is not the only example of a particular cell, or necessarily better than the other examples not shown here. They merely serve to provide a concrete illustration of what embeddables are possible within a given cell, and potentially a starting point for interested readers to explore other similar embeddables. On the other hand, one embeddable might have more than one property relevant to fabrication. For example, an embeddable might exhibit a similar set of properties, such as lumber has nice tensile, compressive and flexural strengths; in other cases, one embeddable might present quite different properties, such as kitchen sponge is both soft and has high surface friction due to its porousness.

6.2 A Library of Embeddables → Rich Materiality

Based on the design space, we have constructed an extensible library of embeddables. To validate its expressiveness, in this section, we first demonstrate results showing how this library enables the exploration of rich material properties in 3D printed objects. We leave the technical details in the next few sections where we then describe how to computationally design with embeddables from the library.

It is beyond the scope of this paper to include every single cell in Figure 6.3; rather, our strategy is to create a set of examples that sample broadly across the design space. In so doing, our goal is to validate how this space holistically affords the enabling of rich material properties beyond the small set supported by today’s desktop 3D printers.

Deformation As shown earlier in Figure 6.1a, inserting wires (e.g., florists’ wire, a straightened paper clip) can enable elastic material (e.g., TPU-based prints) to perform plastic deformation, i.e., being able to stay in a deformed shape. As another example, we can use this kind of embeddables to create adjustable hinges (Figure 6.4) for a pair of Harry Potter glasses².

Density A Bulbasaur phone stand³ designed for earlier devices is too light to stably hold large smart phones (e.g., iPhone 8 Plus). We embed three nuts to put more weight on its base, leveraging their higher density (Figure 6.5). Please see our accompanying video figure for a demo.

Softness People with hand injuries or motor impairment might have difficulty holding hand tools. Similar to creating adaptations [18], we can integrate embeddables like a kitchen sponge to create a softer grip of a wrench (Figure 6.6), which also tightens the grip as sponge has higher friction coefficient due to its porousness.

Friction As another example of friction, in Figure 6.7, a regularly 3D printed door stop would slip on a carpet. Embedding a piece of rubber sheet to replace its bottom increases friction and makes it a functional door stop. Further, rubber also has nice **abrasion resistance**, making the door stop more durable.

²<https://www.thingiverse.com/thing:12993>

³<https://www.thingiverse.com/thing:1726679>

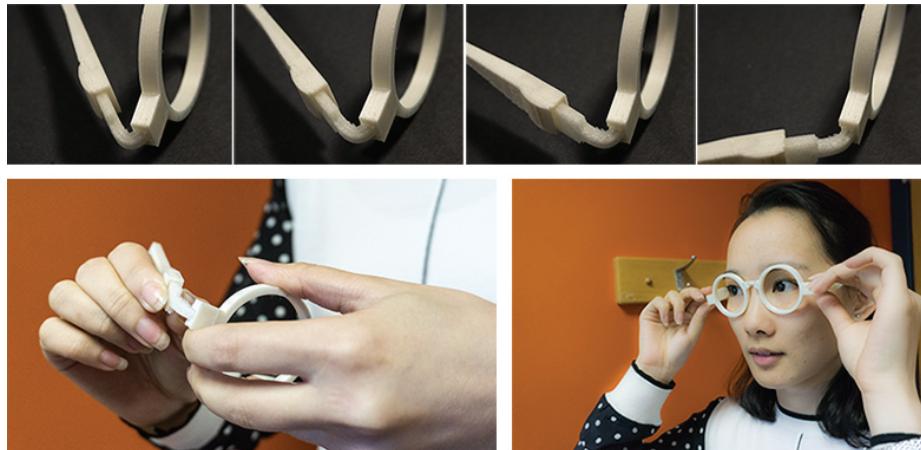


Figure 6.4: Inserting wire makes adjustable hinges for this pair of Harry Potter glasses.

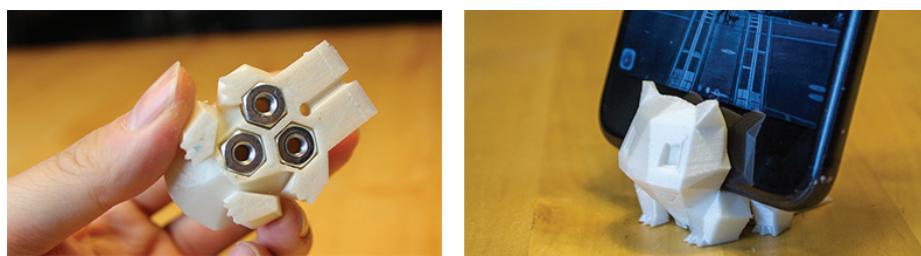


Figure 6.5: Embedding higher density nuts adds weight to stabilize this printed stand for large smart phones.

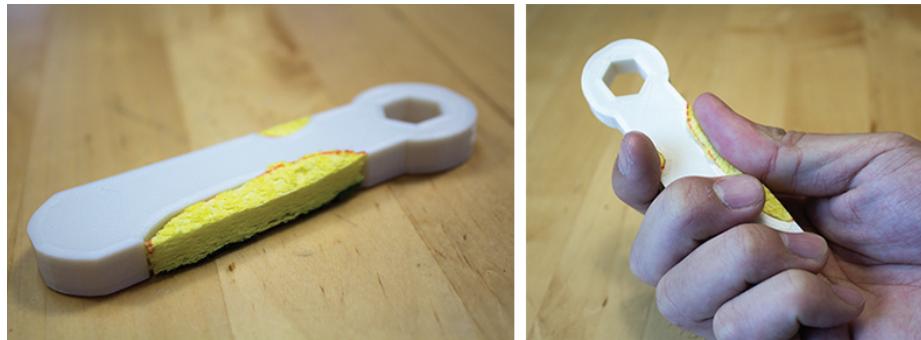


Figure 6.6: Embedding sponge softens and tightens the grip of this printed wrench.

Roughness As shown earlier in Figure 6.1c, embedding a piece of sand paper adds roughness and makes a simple sander.

Flexural strength As shown earlier in Figure 6.1b, a screw adds flexural strength to a 3D printed peg board hook, as it approximates replacing part of the hook with a stronger material. Embedding a wire straightened from a paper clip into a spoon avoids snapping during breakage,

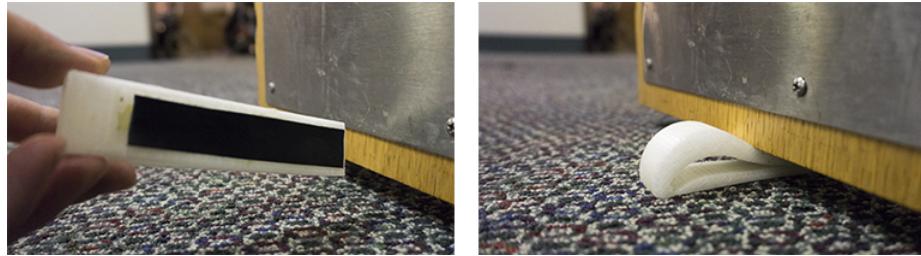


Figure 6.7: Embedding a piece rubber to make a functional door stop.

as snapping could be dangerous to the user (Figure 6.8).



Figure 6.8: Embedding wire straightened from a paper clip into a spoon avoids snapping, which could be dangerous to the user during breakage. Please also see our accompanying video figure.

Malleability As shown earlier in Figure 6.1d, embedding wax (carved from a candle) into an a spoon handle allows users to sand it to a custom shape that the fingers can grip comfortably.

Optical properties As shown in Figure 6.9, for a 3D printed lamp shade, embeddables (e.g., transparent marbles) can add rich optical properties, e.g., making part of the lamp shade more translucent, refracting light to brighten the upper part that is otherwise a bit dark.

Thermal properties Some embeddables have high heat conductivity, such as coins, which can be used to create simple heat sinks on a printed laptop stand (Figure 6.10). Others can insulate, such as silicone oven mitts, which can be used to make printed utensils (e.g., tongs) heat resistive (Figure 6.11).

Having demonstrated a series of embeddable results, the next few sections describe the technical approaches underpinning the design and fabrication process.



Figure 6.9: Embedding clear marbles into a lamp shade makes it more translucent, and can also refract light to brighten the upper part.

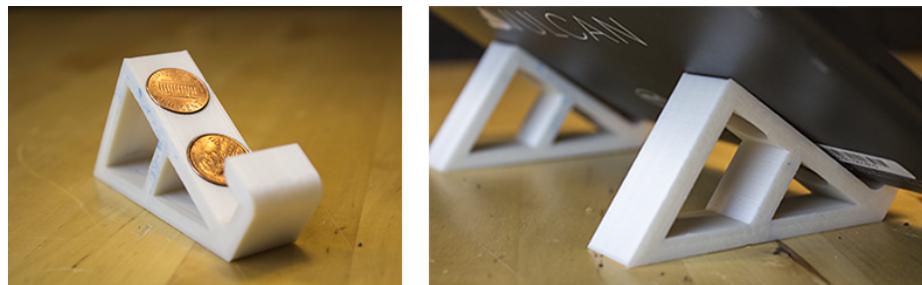


Figure 6.10: Embedding coins to laptop stands creates a heat sink.

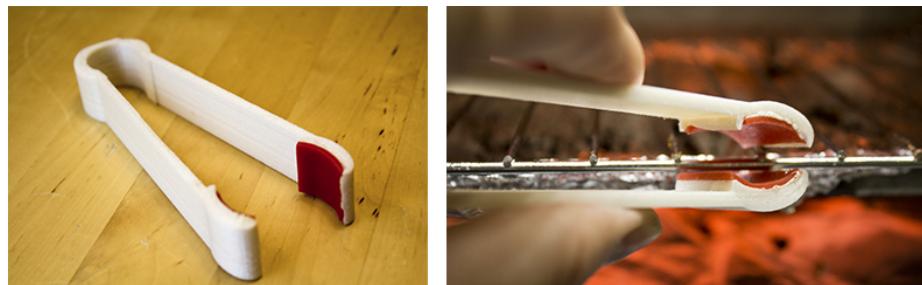


Figure 6.11: Embedding scraps cut from a silicone oven mitt into a pair of tongs makes it possible to be used with hot objects.

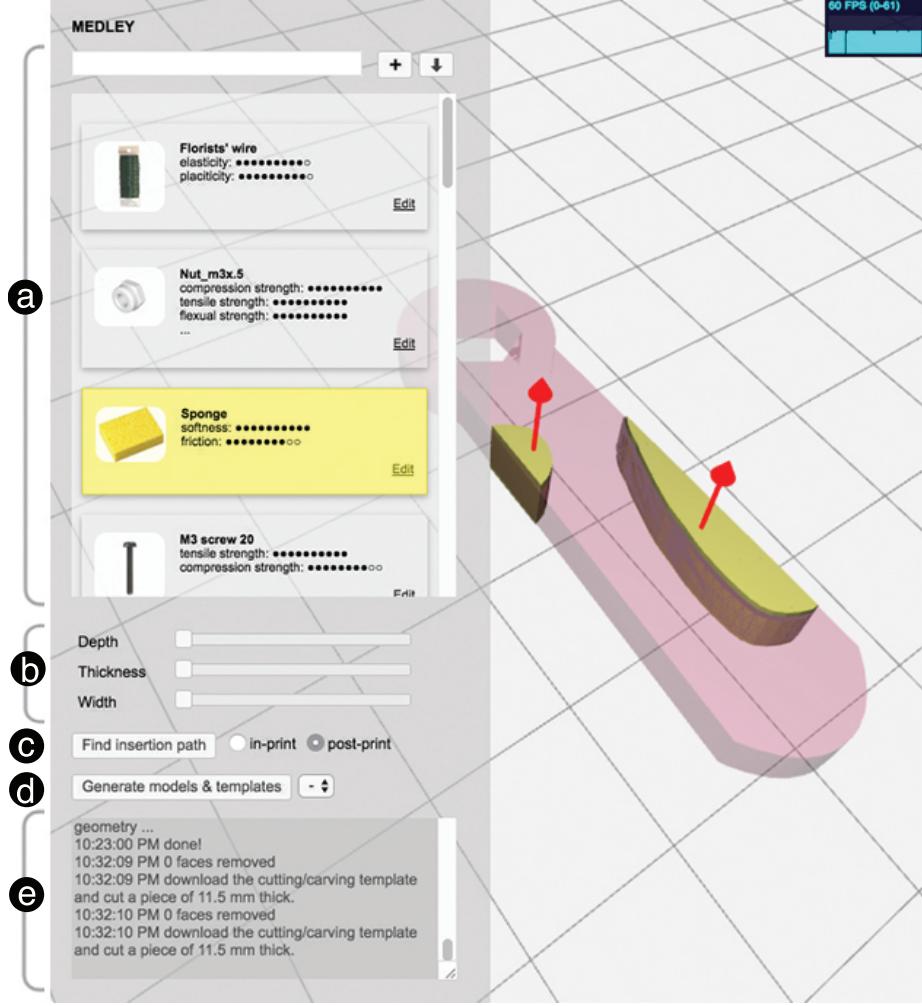


Figure 6.12: Overview of Medley, the computational tool that supports the library of embeddables, from searching an embeddable by material properties (a), to interactively specifying its geometry and placement (b), to finding an optimal insertion path (c), and to generating fabrication-ready model (d) and instructions for the crafting process (e).

6.3 Overview of Design and Fabrication Process

We develop Medley—a computational tool to support the design and fabrication of objects with embeddables (Figure 6.12). A design object (e.g., a wrench) can be imported to the tool. As shown in Figure 6.12a, the library lists each embeddable as a card that contains a thumbnail image, a name, and a short list of material properties. Typing in the search bar finds a specific embeddable, as it dynamically filters embeddables that match the input keywords.

Clicking on an embeddable highlights and selects it (Figure 6.12a), and applies it to the subsequent actions, which we briefly describe here and discuss in further details in the next few sections:

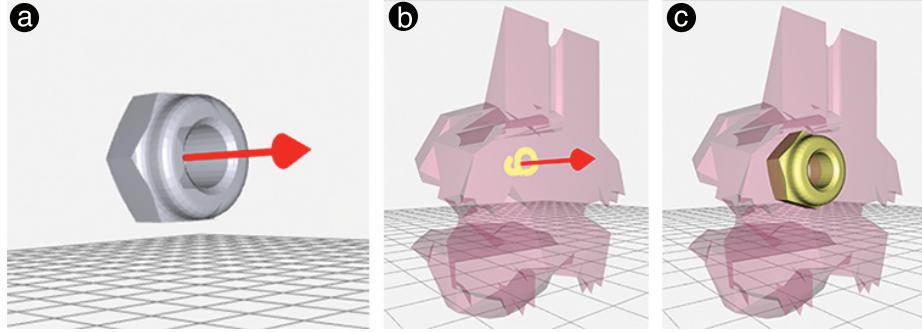


Figure 6.13: Specifying the geometry of a $DOW=0$ embeddable (e.g., a nut) by ‘drawing’ it on a design object (b), which aligns its pre-computed principal axis (a) with that of the sketched points’ (bc).

- **Specifying the geometry of embeddables.** Given the DOW of an embeddable, Medley enables the creation of custom embeddable geometry by directly sketching on the design object, and manipulating its DOW using simple slider widgets (Figure 6.12b).
- **Finding optimal paths to insert embeddables.** Once an embeddable is created and placed into the design object, Medley searches for an optimal insertion path with minimal removal of the design object’s original volume. It supports inserting the embeddable both during and after the printing process (Figure 6.12c).
- **Generating guides for crafting embeddables.** Finally Medley generates a fabrication-ready 3D model with carved out insertion path, as well as guides to cut embeddables into specified shapes (Figure 6.12de).

6.4 Specifying the Geometry of Embeddables

Once an embeddable is selected, the next step is to specify its geometry *in vivo* on a design object, such as the position and orientation of a fixed-shape embeddable ($DOW = 0$), or the shapes for workable embeddables ($DOW > 0$).

One challenge here is that different DOW s constrains how an embeddable can be created in different ways, which could significantly increase the complexity of the interface. As detailed below, Medley addresses this by providing a unified sketch-based input vocabulary that enables simply drawing on the surface of the design object to initiate the placement of the embeddable, or the creation of its shape. A second challenge is aligning an embeddable to certain parts of the design object. This task could be cumbersome if only using translation/rotation/scaling operations in the Cartesian system, as the X/Y/Z coordinates do not necessarily align with the object’s geometry. To address this, Medley leverages the design object’s geometry as a constraint to simplify the editing and transformation task, reducing the variables from nine (translating/rotating/scaling with respect to each of X/Y/Z) to at most three (depth, width and thickness). Below we detail our approaches to tackle these two challenges.

DOW = 0 For a fixed-shape embeddable, we pre-compute its principal axis \mathbf{n}_e —an axis that

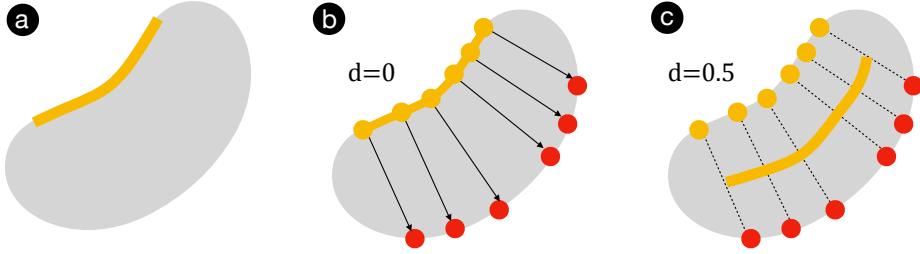


Figure 6.14: Top view of ‘pushing’ an embeddable to a design object: after sketching a to create the geometry (a), each sketch point raycasts along the reversed direction of its normal to find another control point (b), and by interpolating pairs of these points we can ‘push’ in an embeddable, which is driven by a depth value d between 0 and 1.

represents the global orientation of an object—by first fitting all its vertices to a plane using Singular Value Decomposition (SVD) [85], and then taking the plane’s normal as \mathbf{n}_e (Figure 6.13a). As one sketches on the design object, we compute the principal axis \mathbf{n}_s from the sketch points (Figure 6.13b). First, we convert the sketch points $\{P_0, P_1, \dots, P_N\}$ to a polyline L . If L is closed (i.e., a forming a polygon), we apply the similar SVD method; if L is open, we compute \mathbf{n}_s as $\frac{1}{N-1} \sum_{i=1}^N (P_i - P_0)$. We then position the embeddable at the centroid of the sketch P_c , and orient it by aligning \mathbf{n}_e to \mathbf{n}_s (Figure 6.13c).

Once the embeddable is aligned and placed on the surface at P_c , we can push it into the design object by adjusting the ‘depth’ slider. To enable this interaction, we first raycast from P_c along the reverse of its normal, which will hit Q on the other side of the design object. We use P_c and Q as control points: given a depth ratio d read from the slider, we position the embeddable at $(1-d)P_c + dQ$. This approach is applied to all the embeddables with different DOW’s. A more general illustration of the process is shown in Figure 6.14.

DOW = 1 enables two ways of creating a ‘tunnel’ for putting in an embeddable: continuously drawing a freeform path (Figure 6.15ab) and discretely clicking at points that consecutively connect to one another (Figure 6.15cd). Both actions return a polyline L . To make sure the material can be inserted into the tunnel, we straighten L so that it does not go under the embeddable’s bend radius limit r_e . In particular, we iteratively evaluate each three consecutive vertices on L , and fix those whose curving radius is smaller than r_e . We then extrude geometry along the ‘straightened’ L using a circular cross section with the embeddable’s radius.

Similar to $DOW = 0$, it is possible to push the embeddable into the design object using the ‘depth’ slider (Figure 6.15f). This is implemented by creating two control points for each vertex along L and interpolating them with the given depth ratio.

DOW = 2, 3 There are two ways to create geometry for a $DOW = 2$ embeddable (e_2), both of which can be easily extended to a $DOW = 3$ embeddable (e_3):

- Drawing a freeform *open* line L (Figure 6.16), which is similar to the input for $DOW = 1$. However, instead of extruding a ‘tunnel’, we expand the polyline L . First we compute a fitting plane using SVD based on all the sketch points as well as their normals (one such normal is obtained by interpolating the vertex normals of the face that contains a sketch point). Then as shown in Figure 6.16, we expand L bi-directionally along each sketch

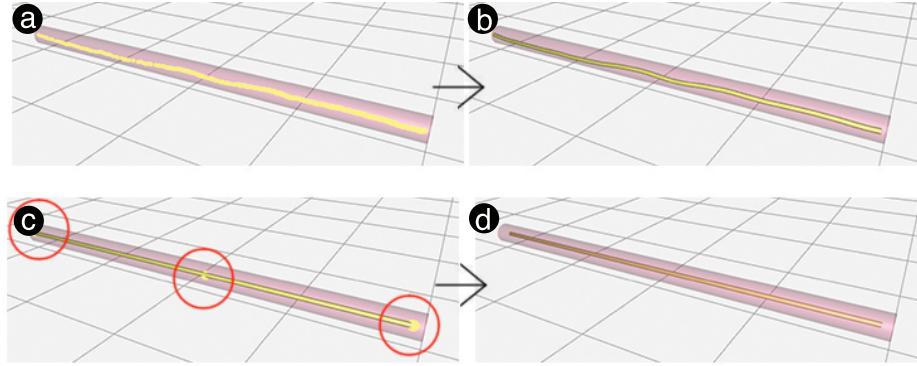


Figure 6.15: Specifying the geometry of a $DOW=1$ embeddable (e.g., a wire) by drawing a continuous line (ab) or clicking at discrete points (cd).

point's normal (Figure 6.16b), which creates a plane cutting into the design object (Figure 6.16c); next, we extrude the plane bidirectionally along its own normal (Figure 6.16d), thus creating a full geometry representing the embeddable (Figure 6.16e).

The amount of these two expansions corresponds to thickness and width, respectively. While $DOW = 2$ embeddables have fixed thickness and can only vary width, $DOW = 3$ embeddables allows for adjustable thickness. Further, both can adjust their depth using the aforementioned control point approach applied on $DOW = 1$.

- Drawing a freeform *closed* line (or pressing ‘shift’ to automatically close an open line) indicates a polygon S rather than a polyline (Figure 6.17a). To create the embeddable geometry, we first retrieve a set of neighboring faces from the design object within a certain geodesic distance from the center of the polygon (Figure 6.17b). We then re-tessellate these faces: for a given face f , we retain it if its vertices are all in S , and remove it if they are all outside of S ; otherwise we retessellate and include the part of f that intersects with S (Figure 6.17cd).

These new tessellated faces create the top side of the embedded geometry (Figure 6.17e). Next, we again obtain a set of control points from raycasting, which create the bottom side of the embeddable geometry. We then add the lateral area between the two sides to finish the whole geometry.

Depth and thickness are adjustable (with thickness only applicable to $DOW = 3$) using the top and bottom vertices as control points.

To summarize, an embeddable’s geometry can be created using a unified technique—sketching on the design object. In other words, selecting an embeddable is similar to choosing a type of paint brush that is subsequently applied to the sketching action. Medley handles the geometric and material details (e.g., DOW , bend radius) while keeping the sketch-based interaction consistently to the minimum across different types of embeddables (e.g., drawing one single stroke). Further, to address the challenge of alignment, Medley uses the design object’s geometry as constraint so that the positioning and orienting of an embeddable is in relation to the design object’s own geometric and spatial properties, such as using points intersected with the design object as

control points to position the depth of an embeddable.

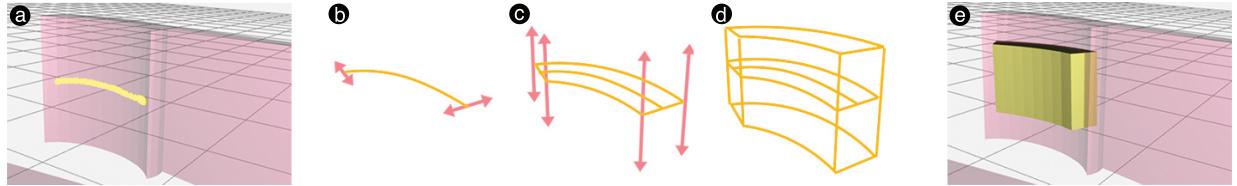


Figure 6.16: Specifying the geometry of a DOW=2, 3 embeddable by drawing an open freeform line (a), which is then expanded into a 3D shape with controllable thickness and width (b→e).

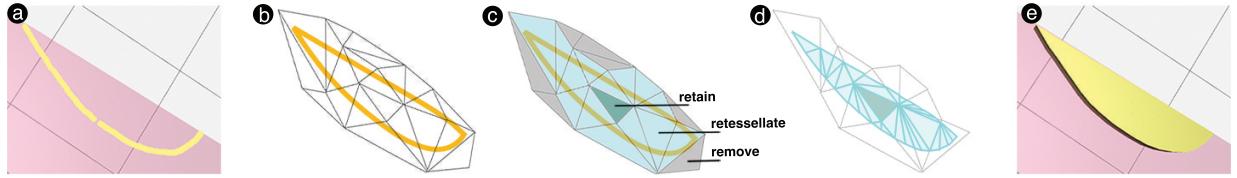


Figure 6.17: Specifying the geometry of a DOW=2, 3 embeddable by drawing an (automatically) closed line (a), which is then retessellated to a surface extrudable into a 3D shape (b→e).

6.5 Finding Optimal Paths to Insert Embeddables

Once an embeddable's geometry is specified, the next step is to find ways to insert it into the design object. In terms of timing, an embeddable can be inserted while printing the design object (in-print), or after the printing is finished (post-print). We can also choose whether the insertion process allows for deformation of the embeddable. For example, inserting a wire causes it to undergo bending while being inserted; inserting a nail, on the other hand, should require no deformation. In our current system, we only consider deformational insertion for $DOW = 1$ with high stiffness (e.g., metal wires but not threads or strings). Figure 6.18 summarizes the $2 \times 2 = 4$ ways of inserting embeddables.

Below we provide details for each of these scenarios, in particular how Medley searches for the optimal path by minimizing the amount of the design object that needs to be removed in order to insert the embeddable.

6.5.1 Deformational Insertion of DOW=1 Embeddables

For $DOW = 1$ we can create a polyline ‘tunnel’ L to contain the embeddable inside the design object. To generate a path for inserting the embeddable, we extend L to the nearest surface. Medley computes such paths for both in-print and post-print insertions, and provides a solution with minimized path length. The first column of Figure 6.18 illustrates the follow insertion path finding scenarios.

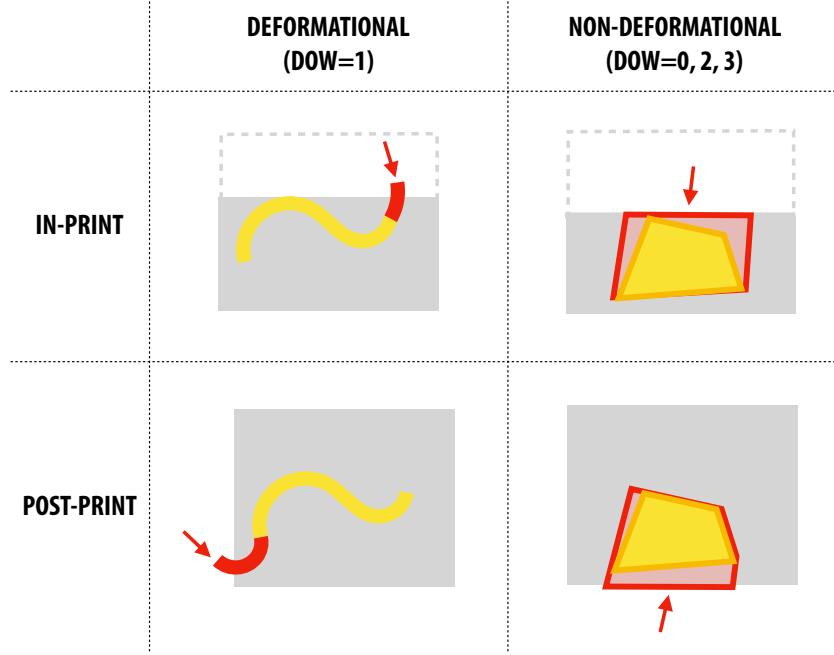


Figure 6.18: A 2x2 space of inserting embeddables that considers (i) whether insertion occurs during or after printing the design object; and (ii) whether the embeddables undergo deformation during insertion.

In-print insertion assumes the entrance of insertion is at the layer where the print job will be paused and the embeddable inserted. First we orient the design object to a user-specified printing direction. Next, along that direction we compute the pausing point, which is a point on L with the highest value when projected on the printing direction. Now, we can extend either one of L 's end points to the layer defined by the pausing point. As shown in Figure 6.19, we compute the tangent vector of an end point, and generate a circular extension path whose radius equals the embeddable's bend radius. Amongst all such possible paths in 3D space, we discard those that cannot reach the pausing layer, and for those that can, we search for the one with the shortest length. We perform this process twice, once for each end point, and obtain the shortest path to insert the embeddable at the pausing layer.

Post-print insertion We perform a similar shortest extension path searching process, except the destination is on the surface of the design object, rather than a horizontal sliced plane.

6.5.2 Non-deformational Insertion of DOW=0, 2, 3 Embeddables

For non-deformational insertion, the insertion path needs to accommodate a sliding volume of the embeddable's geometry (e.g., without undergoing bending). Given a user-defined printing direction, Medley finds a insertion path that minimizes such volume.

In-print insertion. Similar to the deformational case, we first compute the pausing layer. Next, we perform a step-wise (step=5°) search across possible insertion directions at the pausing

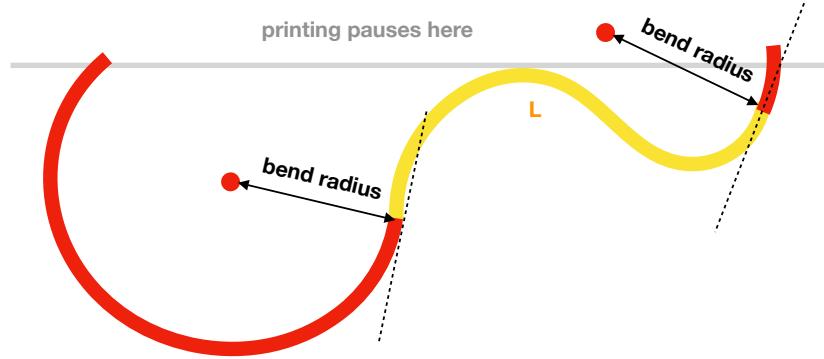


Figure 6.19: Finding an optimal insertion path for DOW=1 by extending each of the end points along its tangent vector, and along a circular path of radius equal to the bend radius of the material. We compute and search for the shortest amongst all possible paths.

layer. For a given insertion direction θ , we compute the minimum volume of the design object that needs to be removed in order for the embeddable to be inserted. To compute this volume, we first discretize the embeddable based on the printing layer height Δh : given θ , the to-remove volume can be discretized into N layers, each perpendicular to θ and has a height of $\Delta h / \cos(\theta)$ as shown in Figure 6.20.

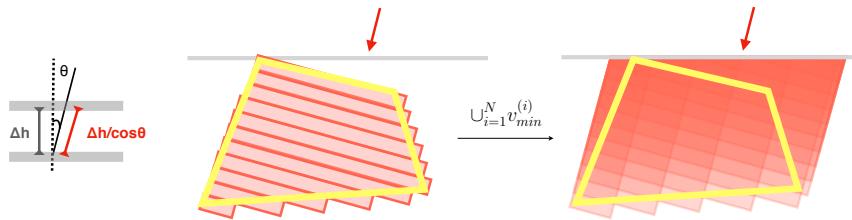


Figure 6.20: Finding an optimal insertion path for DOW=2, 3: for a given insertion direction θ , we compute the minimum volume that needs to be removed in order to insert an embeddable; we then search step-wise across possible insertion directions to find one with the minimum to-remove volume.

The minimum volume of the entire N -layered embeddable can be computed as $V_{min} = \cup_{i=1}^N v_{min}^{(i)}$, where $v_{min}^{(i)}$ is the minimum volume for inserting the i^{th} layer, which can be obtained by sliding (or extruding) that layer from its position along θ until its entirety is outside of the (partially printed) design object; \cup is a union operator. As shown in Figure 6.20, the minimum volume for inserting an embeddable is the minimum volume for inserting each of its layers at their corresponding position. Our search process finds θ_{min} with the minimum V_{min} amongst all possible insertion directions.

Post-print insertion. We perform a similar search process, except for each θ the volume for insertion needs to start from the surface of the design object, rather than a pausing layer.

6.6 Generating Guides for Crafting Embeddables

As a final step, we perform a Boolean operation, subtracting the geometry of the insertion path from the design object. As shown in Figure 6.21, one problem is that this operation will generally leave an ‘hole’ on the object’s surface (for post-print insertion), or in its inner structure (for in-print insertion). To mitigate this problem, we generate a ‘cap’ that can be 3D printed to fill in the hole once an embeddable is inserted. Figure 6.22 shows a example of embedding a nut and a cap (printed in advance) during the printing process.

For non-deformational insertion ($DOW = 0, 2, 3$), we first compute the convex hull of the embeddable, and then subtract it from the geometry of the insertion path to obtain the cap. This approach, however, does not work for deformational insertion, as the printing material might not have the bend radius to allow it to be inserted into the hole. Second, to insert a thin wire/rod, one often needs to cut it longer than the insertion path in order to hold and push it in, which will already fill the hole once inserted. Given these considerations we currently do not to add a cap to deformational insertions.

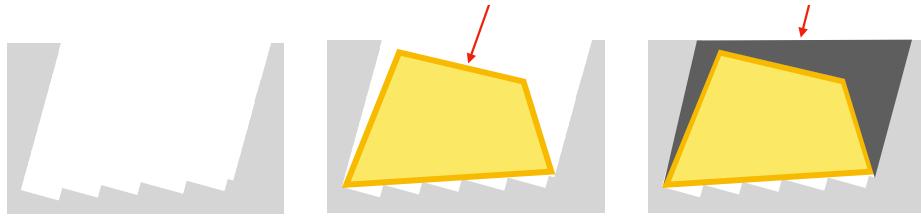


Figure 6.21: Generating a cap to fill up empty space created by removing volume for the insertion path



Figure 6.22: Demonstration of inserting a cap during the printing process: an embeddable (a nut) is inserted (ab), and then a cap (c), after which the printing continues (d) and finishes (e).

Embeddables with non-zero DOW also needs to be cut or carved to match the geometry specified in the tool. We generate guides to aid this crafting process. For $DOW = 1$, we output the length l of the embeddable by summing up the distances of the segments on the polyline drawn by the user. For $DOW = 2, 3$, there are two types of geometry. One is extruded from a polyline, and thus can be obtained by cutting or carving a $l \times w \times t$ piece, where l , w and t correspond to the drawn length, the adjustable width, and the embeddable’s material thickness (fixed for $DOW = 2$, adjustable for $DOW = 3$), respectively. The other type of geometry is extruded from

a polygon. We generate a thin piece of the polygon that can be printed and then used as templates for cutting/carving a piece of embeddable with t thickness. For example, to craft the embeddables for the wrench (Figure 6.6), we printed two templates alongside the wrench (Figure 6.23a). As shown in Figure 6.23b, we then used the templates to trace the designed profile on a sponge and then carved out the embeddable with the thickness informed by Medley’s output (Figure 6.12e).

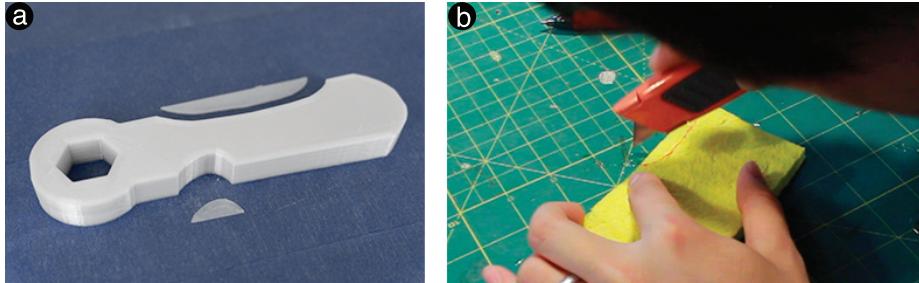


Figure 6.23: For freeform custom embeddable geometry, we generate printable templates to assist the crafting process: in the case of making the soft wrench handle, we can use the printed templates to trace the embeddable’s shapes on a sponge before carving them out.

We apply two strategies to attach embeddables to printed objects. First, having calibrated our 3D printers, we slightly scale the generated insertion path (between 0.5 and 1 mm smaller) to press fit the embeddables, such as fitting a wire into a ‘noodle’, and fitting a piece of sand paper into the slot on a printed sander (Figure 6.1c). In other cases where the objects and the embeddables cannot press into one another, we use a versatile and temperature resistant adhesive⁴.

In this section we discuss outstanding issues, existing limitations and future work for embeddables.

Extending the current library of embeddables Our embeddable library is extensible. As shown in Figure 6.12, clicking ‘Edit’ on each embeddable card brings up a dialog for editing the embeddable. We can change the basic information, such as clicking the name to edit it, or dragging a new image to the dialog to update the thumbnail. More importantly, one can specify information related to material properties. The first step is to select the *DOW*, based on which the dialog further request additional information, such as a 3D model for *DOW* = 0 object, radius for *DOW* = 1 (e.g., radius of a wire), thickness for *DOW* = 2 (e.g., thickness of a sheet), and minimum bend radius for *DOW* ∈ {1, 2} (e.g., how much a wire or a sheet can be bent). Using the a similar dialog, we can also add a new embeddable to the library.

Guidance for quantifying and comparing embeddables Currently our library employs a simple rating approach (scale 1 to 10) to quantify the material properties of embeddables, which enables searching for specific embeddables with a target property. As the library grows larger, it would be useful to compare material properties across embeddables, and sort the search results accordingly. One possible direction for future work is to employ [110]’s method, using the crowd to perform pair-wise comparison for a given material property, based on which a ranking can be computed across all embeddables. Further, we also would like to explore ways to inform users

⁴Currently we use Gorilla Original Glue: <http://www.gorillatough.com/gorilla-glue>

of different material properties and provide more support for them to understand and explore the usage of different embeddables.

Limitations of workability and insertion While our library and fabricated examples demonstrate a wide range of embeddables, there are objects that cannot be used as embeddables due to a lack of workability or difficulty with insertion. For example, a bike spoke has a nice combination of tensile and compression strengths, and has a small dimension to fit in thin structures; however, it is difficult to cut without specialized tools. Our future work plans to provide a list of tools, which can be incorporated into the library, as a way to inform users how to cut or carve a particular embeddable. Some objects are easy to cut, but much more difficult to insert into a printed design. For example, carbon fibre has great structural properties and has been used in industry for strengthening components such as bike frames or airplane turbines. However, a single thread of carbon fibre is fairly soft and very difficult to insert into a ‘tunnel’ in ways similar to inserting a metal wire. In the future we plan to develop ways to insert such soft embeddables (e.g., perhaps by extending [67]’s approach with fabric).

Simulation to inform embeddable choice and design While our library and tool allow people to explore and design with different embeddables, as future focus we should also provide more guidance on: *(i)* what are the existing problems of the design, *(ii)* what embeddables can be used to solve or mitigate such problems, and *(iii)* once inserted whether an embeddable has improved the design.

Automatically suggesting and generating designs Even with simulation, it still requires manually choosing and applying embeddables that are selected from a potentially large library. In the future we will experiment with ways to automatically generate design suggestions based on the input object and problem description (e.g., “increasing the strength around this area”), which is similar to approaches demonstrated in prior work [88]. Further, it is also interesting to search for existing designs from other users that match the current design problem, and also use them to generate more design suggestions.

Chapter 7

Summary and Timeline

Most people started 3D printing with a goal of making something that is actually useful for their daily life, which is a very *real* motivation. However, such a straightforward problem does not come with a straightforward solution. Foremost, 3D printing and many other personal fabrication technologies assume creating objects from scratch, which is oblivious of things that already exist in the real world. These technologies are good at converting geometry into physical objects, but by default they have no idea what and how these objects will be attached to, how they will be installed or assembled, how and how well they will function in the real world context. These questions beyond the conventional fabrication process need to be answered early on and integrated into the design and fabrication workflow, so that there is a guarantee that the things we make are not just physical instantiation of digital shapes, but are also functional in relation to the real world. My thesis work consists of a series of attempts to loop knowledge and awareness of the real world into design iterations. Specifically,

- I have explored attachment techniques for extending existing objects with 3D printed components, which is achieved by analyzing geometry and customizing the 3D printing process (Chapter 3).
- Given these attachment techniques, I have continued to explore how to design such attachments, specifically, how a design tool allows users to describe the usage of real world objects, based on which they can quickly generate 3D printable parts that adapt these objects in their own customized ways (Chapter 4).
- Beyond incrementally augmenting/adapting existing objects, I developed a tool for people to design structures that support existing objects: users can express their ideas via sketching and can customize how the system will optimize the design generatively (Chapter 5).
- I demonstrate that the relationship is reciprocal between 3D printed and existing objects, that existing objects can in turns augment 3D printed designs' material variety by being embedded and replacing part of the original printed material (Chapter 6).

Through this research, I have sought to enable people to *make fabrication real*: to design and fabricate things with considerations of real-world objects, that function in the real world context, and also importantly, that is originated, expressed and iterated by people. I hope this work will pave our way to a future where novel design tools can help people harness fabrication technology to make or augment almost anything in the real world.

October 4, 2017
DRAFT

Bibliography

- [1] Niels Aage, Morten Nobel-Jørgensen, Casper Schousboe Andreasen, and Ole Sigmund. Interactive topology optimization on hand-held devices. *Structural and Multidisciplinary Optimization*, 47(1):1–6, 2013. 5
- [2] Niels Aage, Erik Andreassen, and Boyan Stefanov Lazarov. Topology optimization using petsc. *Struct Multidiscip Optim*, 2014. 5.5
- [3] Harshit Agrawal, Udayan Umapathi, Robert Kovacs, Johannes Frohnhofer, Hsiang-Ting Chen, Stefanie Mueller, and Patrick Baudisch. Protopiper: Physically sketching room-sized objects at actual scale. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 427–436. ACM, 2015. 2.4.2
- [4] Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S Lazarov, and Ole Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011. 5, 5.2.2
- [5] Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. Spin-it: optimizing moment of inertia for spinnable objects. *ACM Transactions on Graphics (TOG)*, 33(4):96, 2014. 2.2.2
- [6] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Illovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 151–160. ACM, 2008. 5.5
- [7] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Everybodylovessketch: 3d sketching for a broader audience. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 59–68. ACM, 2009. 5.5
- [8] Richard T Barrett. Fastener design manual. 1990. 3, 3.1
- [9] Gino van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Journal of Graphics Tools*, 2(4):1–13, 1997. 3.3.1
- [10] Dustin Beyer, Serafima Gurevich, Stefanie Mueller, Hsiang-Ting Chen, and Patrick 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*, pages 1799–1806. ACM, 2015. 2.2.1, 2.3.1
- [11] Hugh Beyer and Karen Holtzblatt. *Contextual design: defining customer-centered systems*. Elsevier, 1997. 5.3.3

- [12] Emre Biyikli and Albert C To. Proportional topology optimization: A new non-sensitivity method for solving stress constrained and minimum compliance problems and its implementation in matlab. *PloS one*, 10(12):e0145041, 2015. 5.2.4
- [13] Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. In *ACM Transactions on Graphics (TOG)*, volume 30, page 158. ACM, 2011. 5.2.2
- [14] Eric Brockmeyer, Ivan Poupyrev, and Scott Hudson. Papillon: designing curved display surfaces with printed optics. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 457–462. ACM, 2013. 2.2.2
- [15] Erin Buehler, Amy Hurst, and Megan Hofmann. Coming to grips: 3d printing for accessibility. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*, pages 291–292. ACM, 2014. 4.5
- [16] Erin Buehler, Stacy Branham, Abdullah Ali, Jeremy J Chang, Megan Kelly Hofmann, Amy Hurst, and Shaun K Kane. Sharing is caring: Assistive technology designs on thin-giverse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 525–534. ACM, 2015. 4.2
- [17] Xiang 'Anthony' Chen, Stelian Coros, Jennifer Mankoff, and Scott E Hudson. Encore: 3d printed augmentation of everyday objects with printed-over, affixed and interlocked attachments. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 73–82. ACM, 2015. 4, 4.3.4
- [18] Xiang'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E Hudson. Reprise: A design tool for specifying, generating, and customizing 3d printable adaptations on everyday objects. In *the 29th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2016. 2.3.1, 6.2
- [19] Lo Chih-Ching. A new approach to cnc tool path generation. *Computer-Aided Design*, 30(8):649–655, 1998. 3.3.1
- [20] Brenda M. Coppard and Helene Lohman. *Introduction to Splinting: A Clinical Reasoning and Problem-Solving Approach*, 3e. Mosby, 2007. ISBN 0323033849. 4.3.1
- [21] Scott Davidoff, Nicolas Villar, Alex S Taylor, and Shahram Izadi. Mechanical hijacking: how robots can accelerate ubicomp deployments. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 267–270. ACM, 2011. 2.3.1, 4
- [22] Lieven De Couvreur and Richard Goossens. Design for (every) one: co-creation as a bridge between universal design and rehabilitation engineering. *CoDesign*, 7(2):107–121, 2011. 4.2
- [23] Jérémie Dumas, Jean Hergel, and Sylvain Lefebvre. Bridging the gap: Automated steady scaffoldings for 3d printing. *ACM Transactions on Graphics (TOG)*, 33(4):98, 2014. 2.2.1
- [24] George Ferguson, James F Allen, Bradford W Miller, et al. Trains-95: Towards a mixed-initiative planning assistant. In *AIPS*, pages 70–77, 1996. 2.4.1
- [25] Ran Gal, Olga Sorkine, Tiberiu Popa, Alla Sheffer, and Daniel Cohen-Or. 3d collage: expressive non-realistic modeling. In *Proceedings of the 5th international symposium on*

Non-photorealistic animation and rendering, pages 7–14. ACM, 2007. 2.3.1

- [26] Wei Gao, Yunbo Zhang, Diogo C Nazzetta, Karthik Ramani, and Raymond J Cipra. Revomaker: Enabling multi-directional and functionally-embedded 3d printing using a rotational cuboidal platform. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 437–446. ACM, 2015. 2.2.1
- [27] Ian Gibson, David W Rosen, Brent Stucker, et al. *Additive manufacturing technologies*, volume 238. Springer, 2010. 2.1.1
- [28] Ronald L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information processing letters*, 1(4):132–133, 1972. 3.4
- [29] Grasshopper. Stress topology optimization with millipede. <http://www.grasshopper3d.com/forum/topics/stress-topology-optimization-with-millipede>, 2017. (Accessed on 08/23/2017). 5
- [30] Daniel Groeger, Elena Chong Loo, and Jürgen Steimle. Hotflex: Post-print customization of 3d prints using embedded state change. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 420–432. ACM, 2016. 2.3.1, 6
- [31] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM, 1999. 2.4.1
- [32] Scott E Hudson. Printing teddy bears: a technique for 3d printing of soft interactive objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 459–468. ACM, 2014. 2.1.1, 2.2.3
- [33] Amy Hurst and Shaun Kane. Making making accessible. In *Proceedings of the 12th International Conference on Interaction Design and Children*, pages 635–638. ACM, 2013. 4
- [34] Emmanuel Iarussi, Wilmot Li, and Adrien Bousseau. Wrapit: computer-assisted crafting of wire wrapped jewelry. *ACM Transactions on Graphics (TOG)*, 34(6):221, 2015. 2.4.2
- [35] Takeo Igarashi and John F Hughes. A suggestive interface for 3d drawing. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 173–181. ACM, 2001. 2.4.1
- [36] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *Acm siggraph 2007 courses*, page 21. ACM, 2007. 5.5
- [37] Autodesk Inventor. About shape generator. <https://knowledge.autodesk.com/support/inventor-products/learn-explore/caas/CloudHelp/clouddhelp/2016/ENU/Inventor-Help/files/GUID-D74F47F3-FE22-44EF-85BE-7C6B1F56DCF9-.htm.html>, 2017. (Accessed on 08/31/2017). 5.4
- [38] Alexandra Ion, Johannes Frohnhofer, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. Metamaterial mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 529–539. ACM, 2016. 6

- [39] David RH Jones and Michael F Ashby. *Engineering materials 1: An introduction to properties, applications and design*. Elsevier, 2011. 6.1
- [40] Michael D Jones, Kevin Seppi, and Dan R Olsen. What you sculpt is what you get: Modeling physical interactive devices with clay and 3d printed widgets. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 876–886. ACM, 2016. 2.2.2
- [41] Rubaiat Habib Kazi, Tovi Grossman, Hyunmin Cheong, Ali Hashemi, and George Fitz-maurice. Dreamskech: Early stage 3d design explorations with sketching and generative design. 2017. To appear at the 30th Annual ACM Symposium on User Interface Software and Technology. 5
- [42] Robert Kovacs. Bottleprint: Scaling personal fabrication by embedding ready-made objects. Ph. D. Retreat of the HPI Research School on Service-oriented Systems Engineering, 2016. 2.3.1
- [43] Yuki Koyama, Shinjiro Sueda, Emma Steinhardt, Takeo Igarashi, Ariel Shamir, and Wojciech Matusik. Autoconnect: computational design of 3d-printable connectors. *ACM Transactions on Graphics (TOG)*, 34(6):231, 2015. 4, 4.3.4
- [44] Gierad Laput, Eric Brockmeyer, Scott E Hudson, and Chris Harrison. Acousturments: Passive, acoustically-driven, interactive controls for handheld devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2161–2170. ACM, 2015. 2.2.2, 2.3.1
- [45] Dingzeyu Li, Avinash S Nair, Shree K Nayar, and Changxi Zheng. Aircode: Unobtrusive physical tags for digital fabrication. *arXiv preprint arXiv:1707.05754*, 2017. 2.3.1
- [46] Rong-Shine Lin and Y Koren. Efficient tool-path planning for machining free-form surfaces. *Journal of engineering for industry*, 118(1):20–28, 1996. 3.3.1
- [47] Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. Legolization: optimizing lego designs. *ACM Transactions on Graphics (TOG)*, 34(6):222, 2015. 2.3.1
- [48] Robert MacCurdy, Robert Katzschmann, Youbin Kim, and Daniela Rus. Printable hydraulics: A method for fabricating robots by 3d co-printing solids and liquids. *arXiv preprint arXiv:1512.03744*, 2015. 1.2
- [49] Jonàs Martínez, Jérémie Dumas, Sylvain Lefebvre, and Li-Yi Wei. Structure and appearance optimization for controllable shape design. *ACM Transactions on Graphics (TOG)*, 34(6):229, 2015. 2.3.2, 5
- [50] MatWeb. Natureworks ingeo™ 2002d extrusion grade pla. <http://www.matweb.com/search/DataSheet.aspx?MatGUID=1e288619764846d2b794bd077e7f1bba&ckck=1>, 2017. (Accessed on 09/05/2017). 5.2.4
- [51] Donald JR Meagher. *Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer*. Electrical and Systems Engineering Department Rensselaer Polytechnic Institute Image Processing Laboratory, 1980. 3.3.1

- [52] Gianluca Memoli, Mihai Caleap, Michihiro Asakawa, Deepak R Sahoo, Bruce W Drinkwater, and Sriram Subramanian. Metamaterial bricks and quantization of meta-surfaces. *Nature Communications*, 8:14608, 2017. 6
- [53] Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. Wireprint: 3d printed previews for fast prototyping. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 273–280. ACM, 2014. 2.2.1, 3.3.1
- [54] Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofer, and Patrick Baudisch. fabrickation: fast 3d printing of functional objects by integrating construction kit building blocks. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3827–3834. ACM, 2014. 2.2.1, 2.3.1
- [55] Stefanie Mueller, Martin Fritzsche, Jan Kossmann, Maximilian Schneider, Jonathan Striebel, and Patrick Baudisch. Scotty: Relocating physical objects across distances using destructive scanning, encryption, and 3d printing. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 233–240. ACM, 2015. 2.2.3
- [56] Donald A Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013. 4
- [57] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. Elastic textures for additive fabrication. *ACM Transactions on Graphics (TOG)*, 34(4):135, 2015. 6
- [58] Huaishu Peng, Jennifer Mankoff, Scott E Hudson, and James McCann. A layered fabric 3d printer for soft interactive objects. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1789–1798. ACM, 2015. 2.1.1
- [59] Huaishu Peng, Amit Zoran, and François V Guimbretière. D-coil: A hands-on approach to digital 3d models design. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1807–1815. ACM, 2015. 2.4.2
- [60] Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. On-the-fly print: Incremental printing while modelling. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 887–896. ACM, 2016. 2.2.1, 2.2.3
- [61] Pinterest, 2016. <http://www.pinterest.com/>. 4, 4.2
- [62] Vinayak Zhang Yunbo Ramani Karthik Piya, Cecil. Realfusion: An interactive workflow for repurposing real-world objects towards early-stage creative ideation. In *Proceedings of the 42th Graphics Interface Conference*, 2016. 2.3.1
- [63] J Rose Plaxen. *Adapt my world: Homemade adaptations for people with disabilities*. Seven Locks Press, 2005. 4, 4.2
- [64] Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. Make it stand: balancing shapes for 3d fabrication. *ACM Transactions on Graphics (TOG)*, 32(4):81, 2013. 2.2.2, 3.5
- [65] Raf Ramaker, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. Retrofab: A

- design tool for retrofitting physical interfaces using actuators, sensors and 3d printing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2016. 4
- [66] Raf Ramakers, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. Retrofab: A design tool for retrofitting physical interfaces using actuators, sensors and 3d printing. *Proc. of SIGCHI*. ACM, 2016. 2.3.1
 - [67] Michael L Rivera, Melissa Moukperian, Daniel Ashbrook, Jennifer Mankoff, and Scott E Hudson. Stretching the bounds of 3d printing with embedded textiles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 497–508. ACM, 2017. 6.6
 - [68] Suzanne Robitaille. *The illustrated guide to assistive technology and devices: Tools and gadgets for living independently*. ReadHowYouWant. com, 2010. 4, 4.2
 - [69] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. Sketchchair: an all-in-one chair design system for end users. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, pages 73–80. ACM, 2011. 2.3.2
 - [70] Valkyrie Savage, Colin Chang, and Björn Hartmann. Sauron: embedded single-camera sensing of printed physical user interfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 447–456. ACM, 2013. 2.2.2
 - [71] Valkyrie Savage, Ryan Schmidt, Tovi Grossman, George Fitzmaurice, and Björn Hartmann. A series of tubes: Adding interactivity to 3d prints using internal pipes. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 3–12. ACM, 2014. 2.3.1, 6
 - [72] Valkyrie Savage, Sean Follmer, Jingyi Li, and Björn Hartmann. Makers’ marks: Physical markup for designing and fabricating functional objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 103–108. ACM, 2015. 2.2.2
 - [73] Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B Goldman, Gautham Mysore, and Wilmot Li. Lamello: Passive acoustic sensing for tangible input components. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1277–1280. ACM, 2015. 2.2.2
 - [74] Ryan Schmidt and Nobuyuki Umetani. Branching support structures for 3d printing. In *ACM SIGGRAPH 2014 Studio*, page 9. ACM, 2014. 2.2.1
 - [75] Martin Schmitz, Andreas Leister, Niloofar Dezfuli, Jan Riemann, Florian Müller, and Max Mühlhäuser. Liquido: Embedding liquids into 3d printed objects to sense tilting and motion. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 2688–2696. ACM, 2016. 2.2.2, 2.3.1
 - [76] Jon Schull. Enabling the future: Crowdsourced 3d-printed prostheticsas a model for open source assistive technologyinnovation and mutual aid. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*, pages 1–1. ACM, 2015. 1.2

- [77] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. Microstructures to control elasticity in 3d printing. *ACM Transactions on Graphics (TOG)*, 34(4):136, 2015. 6
- [78] Kristina Shea, Robert Aish, and Marina Gourtovaia. Towards integrated performance-based generative design tools. *Digital Design, ECAADE*, pages 253–264, 2003. 5
- [79] Ole Sigmund. A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, 21(2):120–127, 2001. 5
- [80] Pitchaya Sitthi-Amorn, Javier E Ramos, Yuwang Wangy, Joyce Kwan, Justin Lan, Wenshou Wang, and Wojciech Matusik. Multifab: a machine vision assisted platform for multi-material 3d printing. *ACM Transactions on Graphics (TOG)*, 34(4):129, 2015. 2.2.3
- [81] Mark A Skylar-Scott, Suman Gunasekaran, and Jennifer A Lewis. Laser-assisted direct ink writing of planar and 3d metal architectures. *Proceedings of the National Academy of Sciences*, page 201525131, 2016. 2.1.1
- [82] Hyunyoung Song, François Guimbretière, Chang Hu, and Hod Lipson. Modelcraft: capturing freehand annotations and edits on physical 3d models. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 13–22. ACM, 2006. 4.5
- [83] Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. Cofifab: Coarse-to-fine fabrication of large 3d objects. *ACM Transactions on Graphics*. 2.2.1
- [84] Lynn Stoller. *Low Tech Assistive Devices: A Handbook for the School Setting*. Therapro, Inc., 1999. ISBN 1933940026. 4.2
- [85] Gilbert Strang. Introduction to linear algebra, 2011. 6.4
- [86] Barna Aladar Szabo and Ivo Babuška. *Finite element analysis*. John Wiley & Sons, 1991. 3.4
- [87] Alexander Teibrich, Stefanie Mueller, François Guimbretière, Robert Kovacs, Stefan Neubert, and Patrick Baudisch. Patching physical objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 83–91. ACM, 2015. 2.3.1, 4
- [88] Michael Terry and Elizabeth D Mynatt. Side views: persistent, on-demand previews for open-ended tasks. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 71–80. ACM, 2002. 4.3.3, 6.6
- [89] Alan S Tetelman and Arthur J McEvily. *Fracture of structural materials*. Wiley, 1967. 3.7.2
- [90] Thingiverse, 2016. <http://www.thingiverse.com/>. 4, 4.2
- [91] Cesar Torres, Wilmot Li, and Eric Paulos. Proxyprint: Supporting crafting practice through physical computational proxies. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 158–169. ACM, 2016. 2.4.2
- [92] Steve Tsang, Ravin Balakrishnan, Karan Singh, and Abhishek Ranjan. A suggestive inter-

- face for image guided 3d sketching. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 591–598. ACM, 2004. 2.4.1
- [93] Erva Ulu, Emrullah Korkmaz, Kubilay Yay, O Burak Ozdoganlar, and Levent Burak Kara. Enhancing the structural performance of additively manufactured objects through build orientation optimization. *Journal of Mechanical Design*, 137(11):111410, 2015. 2.2.1
- [94] Nobuyuki Umetani and Ryan Schmidt. Cross-sectional structural analysis for 3d printing optimization. *SIGGRAPH Asia*, 5:1–4, 2013. 2.2.1, 3.4, 3.7.2
- [95] Nobuyuki Umetani, Takeo Igarashi, and Niloy J Mitra. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph.*, 31(4):86–1, 2012. 2.3.2, 2.4.1
- [96] Tatyana Vasilevitsky and Amit Zoran. Steel-sense: Integrating machine elements with sensors by additive manufacturing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5731–5742. ACM, 2016. 2.2.2
- [97] Marynel Vázquez, Eric Brockmeyer, Ruta Desai, Chris Harrison, and Scott E Hudson. 3d printing pneumatic device controls with variable activation force capabilities. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1295–1304. ACM, 2015. 2.2.2
- [98] Guanyun Wang, Lining Yao, Wen Wang, Jifei Ou, Chin-Yi Cheng, and Hiroshi Ishii. xprint: A modularized liquid printer for smart materials deposition. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5743–5752. ACM, 2016. 2.2.3
- [99] Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W Gellersen. Mix-fab: a mixed-reality environment for personal fabrication. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3855–3864. ACM, 2014. 2.2.3
- [100] Christian Weichel, John Hardy, Jason Alexander, and Hans Gellersen. Reform: Integrating physical and digital design through bidirectional fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 93–102. ACM, 2015. 2.4.2
- [101] David Werner. Disabled village children. *Palo Alto, California: The Hesperian Foundation*, 14, 1987. 4.2
- [102] Wikipedia. Fab@Home — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Fab%40Home&oldid=725172260>, 2016. [Online; accessed 30-July-2016]. 2.1.4
- [103] Wikipedia. Fastener — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Fastener&oldid=724820940>, 2016. [Online; accessed 17-June-2016]. 3
- [104] Wikipedia. Laminated object manufacturing — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Laminated%20object%20manufacturing&oldid=686975516>, 2016. [Online; accessed 01-August-2016]. 2.1.1

- [105] Wikipedia. Stereolithography — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Stereolithography&oldid=725718453>, 2016. [Online; accessed 29-July-2016]. 2.1.1
- [106] Wikipedia. Selective laser sintering — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title>Selective%20laser%20sintering&oldid=714526189>, 2016. [Online; accessed 29-July-2016]. 2.1.1
- [107] Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. Printed optics: 3d printing of embedded optical elements for interactive devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 589–598. ACM, 2012. 2.2.2, 2.3.1
- [108] Karl DD Willis and Andrew D Wilson. Infrastructs: fabricating information inside physical objects for imaging in the terahertz region. *ACM Transactions on Graphics (TOG)*, 32(4):138, 2013. 2.2.2, 2.3.1
- [109] Therese Willkolm. Assistive technology solutions in minutes ii: Ordinary items, extraordinary solutions. 2013. 4, 4.2
- [110] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)*, 34(4):86, 2015. 5.5, 6.6
- [111] Yahan Zhou, Shinjiro Sueda, Wojciech Matusik, and Ariel Shamir. Boxelization: folding 3d objects into boxes. *ACM Transactions on Graphics (TOG)*, 33(4):71, 2014. 3.3.1
- [112] Yahan Zhou, Evangelos Kalogerakis, Rui Wang, and Ian R Grosse. Direct shape optimization for strengthening 3d printable objects. In *Computer Graphics Forum*, volume 35, pages 333–342. Wiley Online Library, 2016. 5
- [113] Olgierd Cecil Zienkiewicz, Robert Leroy Taylor, Olgierd Cecil Zienkiewicz, and Robert Lee Taylor. *The finite element method*, volume 3. McGraw-hill London, 1977. 5.2.1