Intermittent Interaction with Fabrication Devices

by

Ludwig Wilhelm Wall

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirements for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2024

© Ludwig Wilhelm Wall 2024

EXAMINING COMMITEE MEMBERSHIP

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Lora Oehlberg Associate Professor, Dept. of Computer Science, University of Calgary
Supervisor	Daniel Vogel Professor, School of Computer Science, University of Waterloo
Supervisor	Oliver Schneider Associate Professor, Dept. of Management Science and Engineering, University of Waterloo
Internal Member	Craig S. Kaplan Professor, School of Computer Science, University of Waterloo
Internal Member	Matthew Brehmer Assistant Professor, School of Computer Science, University of Waterloo
Internal-external Member	Daniel Harley Assistant Professor, School of Interaction Design and Business, University of Waterloo

AUTHOR'S DECLARATION

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

STATEMENT OF CONTRIBUTIONS

This dissertation includes first-authored peer-reviewed material that has appeared in conference and journal proceedings published by the Association for Computing Machinery (ACM), or are forthcoming (at the time of writing). The ACM's policy on reuse of published materials in a dissertation is as follows [1]:

> *"Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included."*

The following list serves as a declaration of works included in this dissertation:

Portions of Chapter 3:

Ludwig Wilhelm Wall, Alec Jacobson, Daniel Vogel, and Oliver Schneider. 2021. Scrappy: Using Scrap Material as Infill to Make Fabrication More Sustainable. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 665, 1–12.
https://doi.org/10.1145/3411764.3445187

Portions of Chapter 4:

Ludwig Wilhelm Wall, Oliver Schneider, and Daniel Vogel. 2023. Substiports: User-Inserted Ad Hoc Objects as Reusable Structural Support for Unmodified FDM 3D Printers. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 34, 1–20.
https://doi.org/10.1145/3586183.3606718

1 https://authors.acm.org/author-resources/author-rights. Accessed July 21, 2024

ABSTRACT

Personal fabrication devices such as fused deposition modeling 3D printers are fairly slow, often produce lots of waste, and a fully automated fabrication process can take away the cognitive value of crafting something by hand. Interactive fabrication is instead often limited in scope or fidelity of the fabricated items as it requires constant user guidance. However, intermittent user interactions during the fabrication process are sufficient to alleviate downsides of the fabrication process, and direct user involvement can provide a similar personal connection as crafting does. This thesis explores the space of intermittent user interactions in between automated and interactive fabrication. An initial investigation enables physical user interactions during ongoing fabrication processes to reduce internal support material waste. A study with fabrication experts reveals sources of waste and a desire to intervene to take sustainable action. To address external support material as an additional source of waste we investigate more complex or more frequent user interactions. A technical evaluation quantifies the trade-off between increased user involvement and material savings. We investigate frequent interaction during fabrication by enabling sewing on unmodified 3D printers. Our technical evaluation situates the design space of fabricated seams next to traditional stitch patterns. We then examine the spectrum of user involvement during fabrication as a whole through parametric partial assembly tasks. The qualitative analysis of our survey reveal the perceived value of intermittent interaction, scheduling preferences, and design guidelines how systems can offer effective manual interaction options.

ACKNOWLEDGMENTS

First, I would like to thank my supervisors Dan and Oliver for their invaluable guidance and unwavering support, and for giving me the opportunity to pursue my Ph.D. with them at the University of Waterloo. I would like to extend my thanks to Alec Jacobson for lending his expertise and direct support to my first Ph.D. publication. I feel fortunate to have an outstanding and well-rounded committee: Daniel Harley, Craig Kaplan, Matthew Brehmer, and my external examiner, Lora Oehlberg. Finally, I would like to thank my family for supporting me throughout my journey from afar.

TABLE OF CONTENTS

List of Figures xi
List of Tables xiii
1 Introduction 1
1.1 Research Goals 4
1.2 Research Contributions 4
2 Related Work 7
2.1 Interactive Fabrication 7
2.1.1 Assisted Crafting 7
2.1.2 Handheld Fabrication Tools 7
2.1.3 Stationary Fabrication Tools 8
2.2 Print-Pause-Print Approach 9
2.2.1 Improving the printing result 9
2.2.2 Improving the printing process 10
2.2.3 Expressive Iteration 11
2.3 Fully Automated Personal Fabrication 11
2.3.1 Saving Support Material 11
2.3.2 Saving Printing Material 12
2.3.3 Construction systems 12
2.3.4 Summary 13
3 Scrappy - Enabling Intermittent Interaction 14
3.1 Background and Related Work 16
3.1.1 Optimizing prints to reduce material usage 16
3.1.2 Recycling filament material 16
3.1.3 Printing onto objects 16
3.1.4 Inserting objects into prints 17
3.2 Formative interviews 17
3.3 Workflow and examples 19
3.3.1 Populating the Scrap Library 20
3.3.2 Examples 22
3.4 System Implementation 23
3.4.1 Fusion 360 add-in 24
3.4.2 Insertion engine 25
3.4.3 Custom slicer 27
3.4.4 Scrap library 29
3.5 Evaluation 30
3.6 Discussion 32
3.7 Conclusion 33
3.7.1 Summary 33
4 Substiports - Optimizing Multiple Interactions 35
4.1 Related Work 36
4.1.1 Printing Adjustments to Reduce Support 36
4.1.2 Model Adjustments to Reduce Support 37

4.1.3 Mechanical Methods to Replace Support 37
4.1.4 Printing onto Objects 38
4.2 Walkthrough 39
4.2.1 Scheduling User Interactions 40
4.3 Ad Hoc Replacement Objects 41
4.3.1 Technical Tests of Object Properties 41
4.3.2 Interaction Density and Complexity 43
4.3.3 Insertion Feasibility Test 45
4.4 Implementation Details 45
4.4.1 Separate Support into Clusters 46
4.4.2 Find Solutions for Each Support Cluster 46
4.4.3 Combine into Global Solutions 49
4.4.4 G-code Modifications 51
4.5 Evaluation 51
4.5.1 Test Dataset of 3D Models 52
4.5.2 Ad Hoc Objects 52
4.5.3 Procedure 52
4.5.4 Independent Variables 53
4.5.5 Dependent Variables 53
4.5.6 Results 54
4.6 Discussion 57
4.6.1 Ad Hoc Objects for Sustainability 58
4.6.2 Comparison with Previous Approaches 59
4.6.3 Alternative types of support structures 59
4.6.4 Cost-Benefit of Mid-Print Interactions 60
4.6.5 More Complex Replacement Objects 61
4.6.6 Improving Mechanized Approaches 61
4.6.7 Practical Considerations 62
4.6.8 Limitations 63
4.7 Conclusion 63
4.7.1 Summary 64
5 FabricAtion - Extending Machine Capabilities 65
5.1 Related Work 66
5.1.1 Multipurpose Fabrication 66
5.1.2 Working with Layers and Fabric 66
5.2 Approach and Design Space 67
5.2.1 Overview 68
5.2.2 Sewing with a 3D Printer 68
5.2.3 3D Printing Needle and Thread 69
5.3 Walkthrough 70
5.3.1 Starting from flats 71
5.3.2 Defining seams 71
5.3.3 Printing the lower “threads” and the “needles” 71
5.3.4 Cutting fabric shapes 72
5.3.5 Printing the upper thread onto the needles 73
5.3.6 Subsequent Interactions 73
5.3.7 Repair and Alteration using Making Tools 74

5.4 Stitch Pattern Design Space 74
5.4.1 Seam types 74
5.4.2 Parameterized stitch pattern 75
5.5 Implementation Details 76
5.5.1 Planning seam locations 76
5.5.2 Machine code generation 77
5.5.3 Printed guides 77
5.6 Technical Evaluation 78
5.6.1 Apparatus 78
5.6.2 Experiment 1: Seam Strength by Material 80
5.6.3 Experiment 2: Seam Strength by Hole Size 81
5.6.4 Experiment 3: Seam Strength by SPI and Width 81
5.6.5 Additional Test: Needle Size 82
5.7 Discussion 83
5.7.1 Additional variety in seam properties 84
5.7.2 Printing beyond the seams 84
5.7.3 Future Work 85
5.8 Conclusion 86
5.8.1 Summary 87
6 Evaluating Intermittent Interactions 88
6.1 Background and Related Work 89
6.1.1 AI Agents 90
6.1.2 Notifications and Interruptions 90
6.1.3 Slow Interactions at Home 90
6.1.4 Labour leads to love 91
6.1.5 Guiding Assembly Processes 91
6.1.6 Semi-Automated Fabrication 92
6.2 Study 93
6.2.1 Participants 94
6.2.2 Apparatus 94
6.2.3 Procedure 95
6.2.4 Design 98
6.3 Results 99
6.3.1 Mental Demand, Annoyance, Enjoyment, and Contribution 99
6.3.2 Delay Time, Session Time, and Assembly Time 100
6.3.3 Model Value 102
6.4 Codes from Participant Interviews 102
6.4.1 Enjoyment Through Engagement 102
6.4.2 Enjoyment Through Progression 104
6.4.3 Task Partitioning is Loved and Hated 104
6.4.4 Opposing Strategies for Picking Schedules 105
6.4.5 Increased Desirability and Perceived Value 106
6.5 Discussion 107
6.5.1 Guidelines 107
6.5.2 Formalized Schedule Preferences 108
6.5.3 Predicting Preferred Schedules 108

6.6 Conclusion 109
7 Conclusion 111
7.1 The Future of Intermittent Interaction 112
7.2 Towards the Plenitude 113
7.3 Final Word 114

Bibliography 115

Appendices
Appendix A 133
Appendix B 134
B.1 Substiports Model Test Set 134
B.2 Testing Printing on Objects 135
B.3 Support Generation Methods Comparison 140
Appendix C 142

LIST OF FIGURES

Figure 1.1 Intermittent Interaction user involvement spectrum. 5
Figure 3.1 The Scrappy interaction workflow. 14
Figure 3.2 Fusion 360 scrap previews. 20
Figure 3.3 Failed print scrap generator. 21
Figure 3.4 Example scrap inclusions. 23
Figure 3.5 Scrap inclusion optimization process. 27
Figure 3.6 Hollow mesh generation process. 28
Figure 3.7 Slicing results adjustment. 29
Figure 4.1 Substiports interaction workflow. 35
Figure 4.2 Scrap insertion interaction examples. 39
Figure 4.3 Print times changes with availability scheduling. 40
Figure 4.4 Interaction density and complexity space. 43
Figure 4.5 Replacement objects placement process. 47
Figure 4.6 Slicing layer height adjustments. 51
Figure 4.7 Sustiports material savings. 54
Figure 4.8 Substiports print time savings. 55
Figure 4.9 Material savings changes by interaction parameters. 56
Figure 4.10 Replacement solutions variation examples. 58
Figure 4.11 Complex replacement objects mock-up. 61
Figure 5.1 Fabric items with 3D printed seams. 68
Figure 5.2 Printed needle and thread layout. 69
Figure 5.3 Tools to aid inserting fabric. 73
Figure 5.4 Printed seam stitch pattern parameters. 76
Figure 5.5 Tensile strength test apparatus. 79
Figure 5.6 Experiment 1: UTS by printing material. 80
Figure 5.7 Experiment 2: UTS by pre-cut hole size. 81
Figure 5.8 Experiment 3: UTS by stitches per inch. 82
Figure 5.9 Seam strength by stitch pattern width. 83
Figure 5.10 Interlocking printed seam sections. 84
Figure 5.11 Fabric seams created using a laser cutter. 86
Figure 6.1 Apparatus: Locked boxes containing Lego. 95
Figure 6.2 Lego model variations used in the study. 95
Figure 6.3 Assembly instruction examples. 97
Figure 6.4 Diverging bar chart of questionnaire results. 100
Figure 6.5 Delay time distribution histogram. 101
Figure 6.6 Delay time by step number. 101
Figure 6.7 Session time based on interaction density. 102
Figure 6.8 Assembly time per participant. 103
Figure 7.1 Foldable wall mounted 3D printer design. 113
Figure A.1 Scrap objects overview. 133
Figure B.1 Model test set. 134
Figure B.2 Material adhesion test. 137

Figure B.3 Stability test. 138
Figure B.4 Prolonged idle test. 139
Figure B.5 Support types test. 139

LIST OF TABLES

Table 3.1 Printing time and material savings. 31
Table 3.2 Validation and mesh generation times. 31
Table 4.1 Search time by option set and search quality. 57
Table 6.1 Latin squares used to counterbalance study design. 98
Table 6.2 Scheduling objectives for planning interactions. 108
Table B.1 Non-filament material adhesion results. 135
Table B.2 Slicer support material usage. 140
Table B.3 Slicer support printing times. 140

1

INTRODUCTION

3D printing has the potential to be a highly sustainable form of manufacture as it is a local, additive fabrication technology that is essentially geometry agnostic. Local manufacture means that items need not be shipped around the globe. Bioplastics made from renewable sources, such as PLA, can be used as fabrication material and can also be produced locally. Since almost any geometry can be fabricated, extensive set-ups like first creating casting moulds is not required. Subtractive processes, such as stamping metal, turning a lathe, or CNC machining necessarily produce cut-off waste. As an additive process, 3D printing does not necessitate creating waste. In the ideal case, only the material required to build the item needs to be added.

For local, decentralized manufacture, the most impactful machine is the one that is used by the most people. In personal fabrication, that is, digitally assisted manufacture that happens at home, Fused Deposition Modeling (FDM) 3D printing fulfills that role. FDM operates by heating a continuous strand of thermoplast filament within a hotend nozzle. The filament is heated beyond the glass transition temperature and deposited in thin layers onto the build plate, or onto the growing shape, where it cools and hardens. The movement of the print head that contains the heated nozzle is controlled by a computer, and the desired geometry is built from the bottom up layer by layer. 3-axis FDM printers offer an affordable and accessible entry to personal fabrication since they are easy to set up and use. Even after more sustainable fabrication devices become available, entry-level machines will likely remain relevant for a long time, just like many other types of household appliances suffer from slow adoption rates [20].

Yet, commodity FDM printers need to employ some wasteful printing strategies to work around limitations of the technology. For example, parts of a model may start being fabricated separately and are only joined together later in the process. Such geometry would essentially be floating mid-air for some time during fabrication, and will require printing support structures that allow the viscous printing material to settle in the desired location despite the pull of gravity. Support structures may be required on the outside and within the model geometry itself. Internal support acts as "infill" that can increase the mechanical properties of the object, but may also be superfluous to its functionality. External support is often required to print disconnected parts of a model or steep overhangs, and is simply called "support". Overall, these support structures can waste significant amounts of material. In a 2023 survey [106], 41% of users considered supports the biggest source of waste in 3D printing.

The inclusion of these wasteful processes is required to ensure that commodity printers can fabricate arbitrary geometries, but detracts from how sustainable the technology can be in practice. It is desirable to find ways to

make the process less wasteful that are compatible with existing 3D printing machines and commodity technology.

Current 3D printing workflows typically require user interactions at the start and end of a fabrication job, as well as occasional interventions during the process itself. Each print job needs to be started, which may require additional setup, such as providing the desired spool of filament, changing, cleaning, or preparing the build plate, changing the printing nozzle, or configuring machine settings. After each job, the fabricated model needs to be removed and cleaned from supports, brims or rafts, and potentially assembled, the build area needs to be cleared up in preparation for following jobs, and filament may need to be stored to prevent hydration. During the print, user assistance may be required to change or refill the filament spool, to remove build-up of debris on the nozzle, or to recover from errors caused by bumping into the machine, resuming after a power outage, or a clogged nozzle. In each case, the user performs a task that would be unfeasible or impossible to automate, but is reasonably quick and easy for a person to do. Such interactions are commonplace and typically accepted without question. Adding a few additional interactions that are similarly fast and easy may be acceptable to most users, especially if they can provide substantial benefits. But what kind of benefits can be achieved with a limited amount of brief user interactions?

Interleaving human and machine processes is a well established approach in many areas. It is commonly referred to as human-in-the-loop interaction (HITL). In software systems for example, human input is used to steer training and select critical data during machine learning processes [7], or to drive simulations used to test software systems or train vehicle operators [1, 62, 88, 100]. Additionally, safety critical domains, such as air traffic control, keep humans in the loop to ensure safe operation among various automated processes [60, 64]. More broadly speaking, any partly automated process that requires occasional user input, such as driving a semi-autonomous vehicle, could be considered a kind of human-in-the-loop interaction. For example, a simple installation of a software package will typically halt a few times for users to make selections or provide confirmations.

Turn-taking has experienced a recent renaissance due to large language model (LLM) applications. As it relies on written or spoken dialogue for interaction, turn-taking feels natural, and users can rely on their vast experience of engaging in dialogue with people. In fact, simultaneous inputs and outputs during a dialogue is typically considered a communication error [98]. During the advent of computer interfaces, the speed of the back and forth of turn-taking was limited by long processing times. Batch processing reduced required user inputs and effort by grouping commands and jobs together. This also allowed making better use of a limited computation resource by reducing idle times when the system waits for user input. Now, the speed of turn taking is typically limited by the speed of user input, as it takes time for a person to understand the presented information and catch up to the actions that the system has taken. Batch processing remains useful as users

can often evaluate the results of a batch as a whole, requiring less of the user's attention, which is now typically the limited resource.

Today, in most digital applications, users can directly manipulate the system state. This "direct manipulation" has become the dominant form of interaction in software systems where applicable. By working at the exact pace of user input, this typically avoids both waiting times and the need to catch up to actions performed by the system. However, interaction frequency remains a spectrum and interaction methods where the user and the system take alternating turns remain relevant in applications where direct manipulation is not feasible or not desirable.

Direct manipulation in software systems tends to be intuitive and fast. When crafting something by hand in a workshop, interactions are also intuitive, but not necessarily fast. For example, marking a pencil line on a wooden beam can be done as fast as drawing a line in illustration software, but cutting the beam along that line takes time. Some crafting operations, like waiting for glue to set and harden, take so long to complete that they necessitate extensive waiting periods between operations. Bound to physical limitations, crafting can be thought of as slow direct manipulation with turn-taking. Making substantial changes in the physical world takes time, as increasing speed requires increasing force, and when force becomes excessive, being productive turns to being destructive. Consider how the washing machine is decades older than the PC, yet, the time it takes to wash clothes has not changed drastically. As such, they are operated in a turn-taking manner, with requisite loading and unloading interactions, and potential interventions mid-operation to add fabric softener or additional laundry.

Current personal fabrication machines follow the turn taking paradigm, as their execution speed is very slow. Making use of established human in the loop software interactions during fabrication seems incompatible when pauses between interactions can last for hours. Even particularly fast machines, such as laser cutters, fail to reach interactive speeds. Further physical and machine limitations make personal fabrication less sustainable than it ideally could be. However, fabrication machines sit right at the intersection between the analog and the digital world, transforming digital data into analog artifacts. Can personal fabrication follow the footsteps of the history of computation, where software systems help us use make better use of hardware? Since the machines still need to operate in the analog world, the user may facilitate this transition through physical user interactions. By adding a few quick user interactions, we may be able to improve personal fabrication by making it faster and more sustainable. In other words, keeping users in the loop to assist during automated fabrication processes may be a way to make them more sustainable. But what does including human in the loop interactions mean for prolonged processes where interactions could be spread out over hours? Are users willing to participate under these conditions for their own benefit or in the name of sustainability?

Prior work that makes use of user interactions during printing pauses typically ignores the effort required to intervene. This thesis however adopts the view that a successful combination of human in the loop interactions

and personal fabrication that extends beyond minimal interventions requires software that acknowledges the user as part of the process. Making the user part of the process also allows to directly address physical limitations of fabrication processes that make them slow and wasteful.

We demonstrate how this combination can improve personal fabrication by making it more versatile and sustainable, and provide insights and tools to assist in the creation of future software tools that make use of repeated brief interactions.

This thesis investigates keeping the human in the loop during fabrication using *intermittent interactions*, which describes a turn-taking approach to interacting with semi-automated fabrication processes that require periodic user interventions where interactions can also be initiated by the system after a waiting period. I define intermittent interaction with fabrication devices as periodic user interventions during an ongoing process with prolonged breaks between interactions, typically lasting tens of minutes to several hours, to accomplish manual tasks that would be impractical or impossible for an automated system to perform. Intermittent interactions allow working with personal fabrication machines to make better things, and working around machine limitations to make things better.

1.1 RESEARCH GOALS

The high-level research objectives of this thesis are improving the sustainability and output options of personal fabrication, and to examine how software systems can reduce the user effort of interacting with fabrication devices during the process.

In particular, we investigate three core research questions in projects spread throughout the usage space of intermittent interaction:

RQ 1 How can intermittent interaction reduce the material and energy usage of FDM 3D printing to make commodity fabrication devices more sustainable?

RQ 2 How can intermittent interaction enable novel output options using unmodified personal fabrication devices?

RQ 3 What are the subjective costs and benefits associated with intermittent interaction, and how can the perceived costs of interacting be parameterized and reduced?

1.2 RESEARCH CONTRIBUTIONS

We contribute four projects in pursuit of these research questions (Figure 1.1).

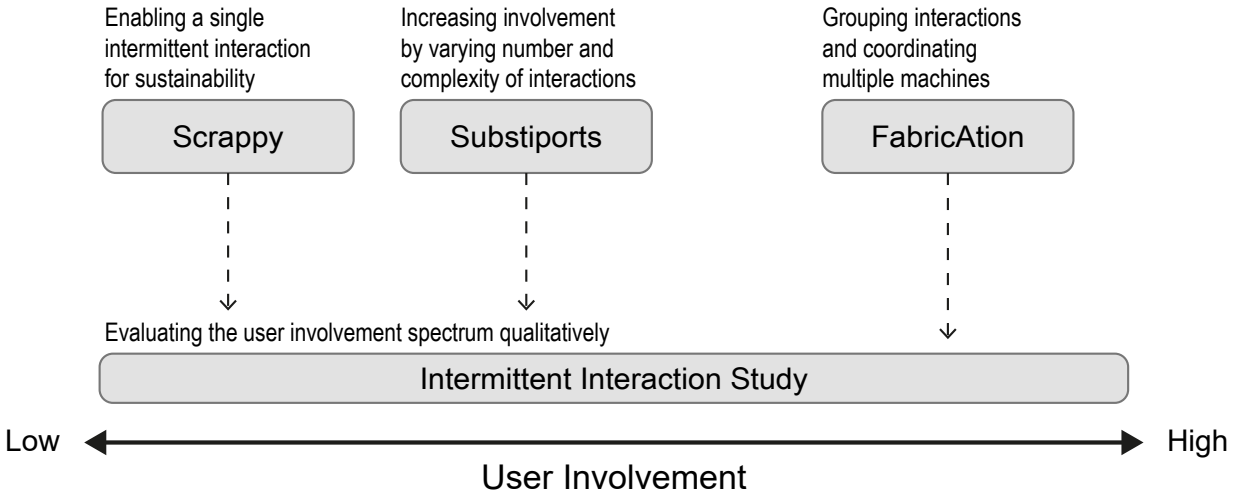


Figure 1.1: The user involvement spectrum of Intermittent Interaction. The first three projects aligned left to right explore increasing levels of user involvement. The fourth project evaluates the spectrum as a whole.

Scrappy and Substiports

The first two research projects answer *RQ 1* by building software systems that enable intermittent interactions to reduce printing wasteful internal and external support structures.

The creation of printed support materials is often the most severe source of waste in personal fabrication. Additionally, fabricating supports considerably increases fabrication times and can decrease the surface finish quality of fabricated objects. For example, external support structures introduce additional travel moves between the model itself and the support, and necessitate crossing the borders of the model, which can leave marks or thin strands of excess material on the surface. Support material touching the model can bond the two together, which can leave further surface blemishes. Wasting printing material further increases running costs, and slow operation makes using the machines less attractive to current and potential new users. We introduce two systems to demonstrate how intermittent interaction can improve existing personal fabrication workflows. Through these software systems, Scrappy and Substiports, foreign objects are used to replace part of the internal or external support structures of a fabricated object to lower the costs of creating the object.

The Substiports project pushes sustainability improvements further by enabling multiple user interactions with more complex interaction opportunities. Even with additional interactions, long gaps remain between each of them due to the slow speed of 3D printing. Our system replaces external support structures with user-placed objects with variable sizes and ad-hoc towers. As external support often surrounds models, to support most of it, a user may have to be involved in the fabrication process multiple times and at different stages of the process to effectively address this source of wasting

material. As the system considers the complexity and density of interactions during optimization to evaluate various replacement options, the system also provides insights into *RQ 3*.

FabricAtion

To answer *RQ 2* and investigate intermittent interactions that approach interactive fabrication, we designed a novel fabrication method where the user fulfills an instrumental role in the construction of the final object, rather than improving an existing workflow. Our system enables a new fabrication process inspired by sewing through sets of user interactions and fabrication steps that are combined into batches. This method uses 3D printed "needles" and "thread" to attach layers of fabric. The automation provided by the fabrication machines can be used to guide users without sewing expertise. This project explores interactions that require the attention of the user for periods of time between batched jobs by grouping multiple short tasks into a single interaction and by combining fabrication steps to run in parallel.

Intermittent Interaction Study

The final project gathers qualitative data observing the intermittent interaction participation spectrum as a whole and to answer research question *RQ 3*. It is an in-situ multi-day user study where participants are asked to perform repeatable synthetic assembly tasks that mimic the frequency and complexity of intermittent interaction during fabrication. Specifically, locked boxes with LEGO pieces are handed out to participants. The LEGO pieces are used to perform manual assembly tasks, which represent the high precision and low effort tasks of prior systems. Custom assembly instructions control the complexity of each assembly step and are provided through a custom online server, which handles timed notifications that mimic interaction requests during a fabrication process. The timing of the notifications ensures that each assembly requires 4 hours to complete. The study is designed to provide complementary data to the quantitative measures gathered in previous projects. Study insights are used to define design guidelines for incorporating intermittent interaction into software systems to make research results more accessible for practical application.

To situate these projects, we describe relevant related work in the following chapter before presenting and discussing each research project in detail.

2

RELATED WORK

The work related to the research projects in this thesis can broadly be categorized into three groups. The first group covers interactive fabrication, in which a user works in collaboration with the machine during the entire process. The second group handles works that leverage pausing a fabrication process to interject some additional step. These steps are often performed fully automatically by motion controlled systems, such as robot arms, instead of by a user. The third group contains works that aim to improve traditional fully automatic personal fabrication.

2.1 INTERACTIVE FABRICATION

Interactive fabrication uses real-time user inputs to control digital fabrication devices. In the most basic form, this approach resembles crafting with hand tools, but makes use of digital tools instead.

2.1.1 *Assisted Crafting*

Approaching interactive fabrication are methods that use traditional hand tools, but also make use of digital guidance systems to assist unskilled users. These methods may best be described as assisted crafting. As an example, augmented reality techniques can be used to indicate areas of a work piece that should be removed to carve out a model [35, 86]. Drill Sergeant [94] augments a variety of analog tools to project visual in-situ instructions that help to complete project tutorials.

2.1.2 *Handheld Fabrication Tools*

A common approach to allow users to steer the fabrication process directly is to put the tool directly into the user's hands. Interactive fabrication systems that make use of handheld devices can broadly be thought of as digital hand tools enhanced by a support system. In these systems, one role of the user is positioning the tool around the work piece to be able to engage with it from almost arbitrary angles.

For example, digital handheld milling tools [128–130] provide the milling capabilities of a rotary tool (e.g. a Dremel) and combine it with a digital 3D model of the work piece that is used to guide or document the process. By tracking the current location of the tool, the 3D model can be used to prevent milling material that should be part of the model by halting the tool as it approaches the model boundary. The accuracy of the process can be improved by including a small-scale motion control system within the tool

that can apply fine adjustments to the position of the tool head. Alternatively, the positional data can be used to create a new model or make adjustments to an existing one by tracking which points in space have been milled away and should therefore no longer be part of the 3D model.

Since the tool needs to be light enough to be moved both by the user and the included motion control system, the power and speed of operation are limited. As a result, application scenarios for the tools are limited to soft materials, such as balsa foam, or to smaller sized models. Manufacturing times would become untenable for larger models or harder materials. 3D printing pens, which operate similar to FDM 3D printers by additively building up a model, suffer from a similar issue. Creating a large model that would take multiple days to fabricate on a 3D printer with a handheld pen would require excessive amounts of user effort and would have to be broken up into many interactive fabrication sessions. As a result, prior work that makes use of 3D printing pens suggest application scenarios that are limited to for example small scale tools and models or simple sensory circuits [89, 124, 131].

2.1.3 *Stationary Fabrication Tools*

Other interactive fabrication systems use stationary devices that put less of a physical burden on the user. They allow engaging with sturdier materials or larger projects by increasing the speed of the fabrication process. Despite the increased speed, these systems are often still primarily designed as exploratory tools for drafting and prototyping, as the creation of high fidelity objects would require extensive active time investments.

Higher speeds can be achieved by making use of faster fabrication processes or by making adjustments to the process, such as restricting the output to a simplified version of the object [79]. RoMA [78] for example fabricates objects through a motion controlled robot arm that is being used as a 3D printer. A user provides spatial inputs which are then manifested physically as a printed wireframe outline by the robot arm in real time. This work allows designing and printing directly on existing objects by 3D scanning parts of the object. RoMA is primarily designed as a prototyping tool and requires extensive additional hardware, such as a robot arm, to print onto existing objects. FormFab [70] makes use of formative fabrication, reshaping a plate of material that becomes pliable when heated. The process introduced by this project is highly expressive, but lacks precision compared to other fabrication methods.

Turn-by-Wire [105] and Interactive Construction [68] make use of a digitally driven lathe and a laser cutter respectively. These subtractive fabrication methods are fast enough that they offer the potential to perform interactive fabrication with sturdy materials to create high fidelity models intended for production rather than drafting. The steps taken during an interactive fabrication session can be recorded, modified, and re-run without further user interaction on other work pieces. For large work pieces, lengthy fabrication sessions may be required, potentially including idle times of several minutes

while the user waits for a fabrication step to complete. Both projects could however be employed as a special form of intermittent interactions in which all interactions are performed at the start of the fabrication process. To do so, the fabrication steps are recorded in an interactive session on a small and soft work piece to avoid waiting times. Afterwards, the fabrication of a production piece is performed automatically on a larger work piece by running an adjusted, scaled up version of the recorded instructions.

2.2 PRINT-PAUSE-PRINT APPROACH

The body of work most related to this thesis contains interruptions to an otherwise automated fabrication process. The interruption often allows a user to perform an interactive task, but is sometimes used to allow a separate or even external part of a system to perform an automated action without interfering with the fabrication process. Outside of personal fabrication, this approach has been coined Print-Pause-Print (PPP).

2.2.1 *Improving the printing result*

The PPP approach has been applied in the construction of a variety of 3D printed fluidic devices. For example, Pinger et al. [80] proposed observing and manually pausing the fabrication process once it reaches the desired layer height. During this pause, the observer places a precut thin membrane onto the printed part, which acts as a barrier that separates a fluid chamber into two parts. Gaal et al. [30] integrated different types of material into a 3D printed microchannel. The integrated material is used as an electric sensor to create an artificial tongue, as the impedance of the material changes when in contact with fluids during tasting. Custom machine code was added to the fabrication process to create a pause. Inserted materials were affixed with tape to secure them in place. Further examples include integrating a magnetic stirbar into a fluid chamber [95], and filling chambers with liquid reactants [53, 54] during a printing pause before they are sealed once printing continues. Banna et al. [10] attempted to insert sensors that monitor water quality during a printing pause. The approach was abandoned due to misalignment issues.

These projects are limited to a single, manually guided printing pause. Given the limited application of pauses and reliability issues being noted, guidelines and systems to support mid-print interactions may be desirable.

Some previous work proposes interaction methods that use or resemble intermittent interactions and also aims to improve personal fabrication in general, rather than fluidic devices.

2.2.1.1 *Printing onto objects*

Integrating existing objects into the design and fabrication of 3D models has been explored previously in various ways, including printing on top of other objects. Encore [18] is a method to print on top of existing objects to

enhance them, and to print through loops of existing objects, such as the handles of scissors. To do so, the loop is placed inside a printed trough during fabrication, before printing on top of the trough which closes a second loop through the first one.

Attempting to fit a fabricated part to an existing object often requires multiple iterations before achieving the desired fit. This process can be sped up by incorporating the target object directly in the fabrication process. The fabrication process can be customized to specific objects for example by integrating a touch screen into the fabrication machine that allows capturing the object's outline and modelling around it directly in Fitter [3], or by pushing an object into resin actively being cured to create a perfect mold or fixture around it [127].

2.2.1.2 *Inserting objects into prints*

Printy3D [125], HotFlex [34], and other systems [22, 27, 97] incorporate electronic components and physical widgets to enhance the functionality of fabricated objects. These components are inserted during printing, which is a form of intermittent user interactions. To investigate the design of nesting objects that can be inserted during fabrication, Jacobson created a generalized way of approximating the maximum scale at which one model fits within another [41].

2.2.2 *Improving the printing process*

With the help of complex motion controlled mechanisms, the printing process itself can be improved. For example, by replacing some of the exterior support structures or by moving elevated platforms to print on into place. Xu et al. [37, 121] propose subdividing the build plate and integrating a motion controlled system to dynamically raise parts of the build plate during fabrication. This makes the build plate programmable, as the height of each segment of the build plate can be adjusted by inserting lifting rods of varying length before printing starts. Parts of the build plate are then raised automatically to their mechanically pre-programmed heights during fabrication and used as an elevated printing surface to replace printed supports. The build plate is segmented into a grid of 11 x 9 square segments, each 12.7 by 12.7 mm. A layout optimization algorithm provides user instructions on how to program the build plate by manually inserting a grid of rods. The system does not take the user effort for this front-loaded interaction into account. Intermittent interactions can instead also be used to provide elevated printing surfaces.

Yigit et al. [123] introduce a simulated system that uses pre-fabricated reusable modular support replacement objects. A virtual robot arm is programmed to place modular blocks onto the build platform during a simulated printing process. In this approach, support towers are generated on a 1x1 cm grid, and a sliding window algorithm is used to place 4x4x4 cm blocks. An average of 75.5% support material savings are reported for two test models.

2.2.3 *Expressive Iteration*

Some projects perform multiple iterations of fully automated fabrication with manual crafting steps in between iterations. Multiple iterations of such turn taking in combination closely resemble intermittent interaction usage pattern. ReForm [117] allows designing 3D models by using clay prototypes. Users manipulate the clay using their hands to make quick free-form adjustments. During the automated parts of fabrication, the printer scans the model, extrudes new clay and mills away parts of the model. 3D Pen + 3D Printer [102] explores the benefits to expression and creativity via delaying parts of the design process until the base version of a model has been printed. By allowing users to make manual adjustment with hand tools like a 3D printing pen, the work shows that creative customization tasks can be separated from the bulk effort of the fabrication process.

2.3 FULLY AUTOMATED PERSONAL FABRICATION

Prior work has established several techniques that make fabrication more sustainable, or that speed up iteration. For example, trying to limit the amount of support required to achieve a successful print is a key topic in FDM 3D printing.

2.3.1 *Saving Support Material*

The geometry of the support itself can be adjusted to reduce material usage. For example, supports can be merged into smaller columns [38], they can form truss, scaffolding, or tree structures [23, 26, 44, 110], or the support volume can be filled with material efficient micro structures [101].

In some cases printing support material underneath overhangs can be avoided entirely. Print on Air [11, 25] creates horizontal overhangs by carefully attaching material sideways onto existing geometry by making use of reduced printing speeds and strong cooling. This method was shown to be successful for overhangs of 90° relative to the printing direction. A similar effect of depositing material sideways onto existing geometry can be achieved by adjusting the slicing process to divide the printed model into non-planar slices [72]. When following these non-planar tool paths the print head moves slightly up and down relative to the build plate. This up and down motion aids in the sideways deposition of material and can enable printing overhangs of slightly above 90°, limited by the geometry of the print head.

RoboFDM [119] offers a different way that can potentially avoid all support material. By attaching the build plate to a robotic arm, models can be printed without being restricted to a single building direction. By rotating the build plate during fabrication, parts of the model that would have conventionally required external support material can often be positioned such that the already printed parts of the model provide the surface to print on instead.

2.3.2 *Saving Printing Material*

Other prior work aims to reduce the amount of material used to create the model. For example, WirePrint [67] approximates the objects' surface by printing a wireframe version of the geometry, but this reduces fidelity and is therefore only applicable for certain kinds of rapid prototyping. If object detail is important, material consumption can be reduced by optimizing the required infill to support all exterior walls of the object [46], or to optimize only for expected loads [33, 120].

Patching Physical Objects [104] enables printing onto existing objects with the purpose of making physical adjustments to the model or performing repairs. Under the right circumstances this can avoid printing support material entirely by avoiding to print a new model, but also requires additional hardware in the form of a depth camera, a mill and a 5-axis rotating platform.

PackMerger [111] converts the printed model into a shell, divides it into segments, and arranges the segments to minimize the support material required to print each segment. This method is not applicable to models with high mechanical requirements and requires additional user effort to glue the printed parts together to assemble the model.

When printing multiple models at once, slicers commonly place the models sufficiently far apart from each other that printing one does not interfere with the printing process of another. Jiang et al. [45] propose arranging multiple models in such a way that smaller objects are placed underneath the overhangs of larger models. This way, the smaller models can act as an elevated platform to print on and replace some of the support material required to print the larger models.

Instead of directly reducing material usage, fabrication can be made more sustainable by recycling the material of prints that are no longer in use. It has, however, been shown that the mechanical properties of recycled materials degrade with each recycling iteration [77], and the recycling process itself is energy intensive. Systems that are deployed in changing environments can be made more sustainable by employing a modular system design that allows reusing most of the parts while only fabricating and switching modules when circumstances change [96].

2.3.3 *Construction systems*

One way to make fabrication more sustainable is to create or incorporate a construction system that approximates 3D models. Prior work has already incorporated ready-made objects, such as specific bottles or cans, in 3D models to replace parts of large structures instead of printing them. TrussFab [55] for example uses certain types of bottles to create sturdy human scale truss structures, and ReFabricator [122] approximates large arbitrary shapes using everyday objects. This enables the creation of structures that surpass the printing volume of the printer that helped create them. By printing only connectors and approximating the overall shape of the 3D model, these

projects also reduce the amount of printing material that would otherwise be needed to fabricate them. Fabrickation [69] instead uses LEGO bricks to replace most of the models' geometry while only 3D printing the parts of the object in high fidelity that are crucial to its function. While mainly intended to reduce print times, this system allows reusing parts of the models once they are no longer needed. Although these works improve the sustainability of prints, they lower the output fidelity by approximating the desired original shape.

2.3.4 *Summary*

The space of intermittent interactions during fabrication sits between automated personal fabrication and interactive fabrication, and techniques introduced in either domain may be relevant to inform the design of human-in-the-loop fabrication systems. Introducing intermittent interactions to both fully automated and fully user driven applications could improve their respective workflows, such as reducing either their usage costs, or the user effort required to perform them. And little work exists that makes use of more than one interaction during long duration processes. We begin our exploration of the usage space at the low interaction end of the spectrum, aiming to improve fully automated fabrication.

In the following chapter, we present a software system for fused deposition modelling 3D printing that replaces infill material with scrap to reduce material and energy consumption. Example scrap objects include unused 3D prints from prototyping and calibration, household waste like coffee cups, and off-cuts from other fabrication projects. To achieve this, our system integrates into an existing CAD workflow and manages a database of common items, previous prints, and manually entered objects. While modelling in a standard CAD application, the system suggests objects to insert, ranked by how much infill material they could replace. This computation extends an existing nesting algorithm to determine which objects fit, optimize their alignment, and adjust the enclosing mesh geometry. While printing, the system uses custom tool-paths and animated instructions to enable anyone nearby to manually insert the scrap material.

3

SCRAPPY - ENABLING INTERMITTENT INTERACTION

Personal fabrication methods naturally generate scrap. Our formative study showed that failed prints, tests, and offcuts occur naturally in the iterative, creative practice of designing and testing physical objects. While experts might strive to use material efficiently, there will still be waste; with novices, there will be even more. These scrap objects are often thrown out, but some end up cluttering work spaces with the hope they can be used in the future. In some fabrication environments, bags of waste objects are stored for years with the hope of finding an eventual use. The process of re-melting waste filament to create new spools is commonly known, but almost never implemented, since it requires specialized machines and reduces filament quality.

Meanwhile, fused deposition modelling (FDM) 3D printers, the most consumer friendly and widespread computer aided fabrication tool, use time, energy, and material to create infill for critical internal support structures. This infill sparsely fills the internal volume, most commonly using a repeating pattern such as a honeycomb. We propose to repurpose scrap by integrating it into FDM prints, reducing the material consumption of infill while also finding a new use for scrap objects. Compared to recycling methods like re-melting old prints into new filament [9], our approach requires no additional equipment, does not degrade printed material, and can reuse many other types of scraps, objects, and materials. With the exception of food and other materials that can decay quickly, most materials are suitable as scrap in our system. Examples include single-use packaging, such as jars, lids, and boxes, discarded hardware, such as CDs, floppy discs, and tools that come with construction kits, or consumable items, such as markers, chopsticks, glue sticks, and empty tape spools. Since our system works on a wide variety of scrap materials, even users who already have the means to re-melt their old

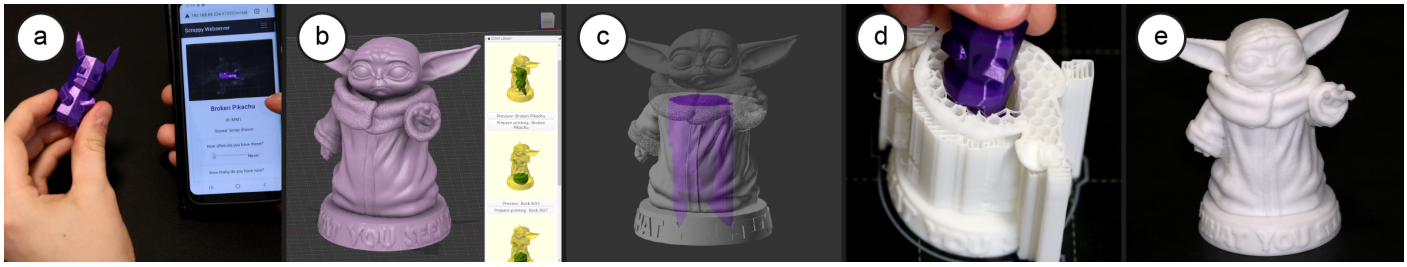


Figure 3.1: Scrappy is a system that lets users use scrap objects as infill to reduce time, cost, and material: (a) a library tracks scrap objects, like an old print, as well as other common objects; (b) an add-in for Fusion 360 suggests scraps from the library that can fit into a model, sorted by saved printing time; (c) an algorithm finds the best scrap placement and generates a modified model; (d) a custom slicer optimizes infill and adds machine commands to pause the printer with instructions how to insert the scrap into the object; (e) the final printed object has the scrap inside, reducing the material, time, and energy needed to print.

prints into new filament can benefit from our system. During testing, we were able to continue printing on all tested materials, which were stone, wood, metal and plastic.

Recycling is a conscious choice that many Do It Yourself (DIY) supporters have already made. It takes some additional time and effort for example to wash out an empty jar so that it can be recycled, instead of just putting it in the trash. Recycling requires an object to be reduced to its raw materials so that it can be re-injected into the manufacturing cycle, which requires additional energy. Similar to recycling, using our system requires some additional effort. Unlike recycling objects that have no more use though, our system provides a way to directly repurpose them in their current form. This repurposing effort in turn provides direct cost savings to the user, as well as contributing to a more sustainable lifestyle.

We introduce Scrappy, a system that automatically searches a library of scrap objects for ones that fit into a CAD model, ordered by how much infill it would replace. The implemented system shown consists of a Fusion 360 plugin to monitor the modelling process, search for fitting scrap objects in the background, and compute a hollowed 3D model and printing instruction files to fabricate the model with the selected scrap inside (Figure 3.1). Our insertion engine that handles the search for possible scrap inserts is based on an existing 3D nesting algorithm [41]. A web-based scrap library keeps track of scrap objects, helps users find them in the workshop, and allows them to add new scrap. A customized slicer software generates printer code optimized for scrap insertions. We demonstrate how Scrappy handles common scrap, including previous outdated or broken prints, single use packaging, broken hardware, and many more, showing that scrap-filled objects require less material, time, and energy to fabricate.

The costs for this approach are a single short user interaction during fabrication, additional computation, and maintaining a scrap library. Most importantly, scrap has to be inserted into the model while it is being printed. Determining which scraps can fit inside a model and adjusting the model accordingly requires additional computation. And finally, the system requires knowledge about which scrap objects are available for insertion, and this set of scraps must be filled. Our system design minimizes these costs. It supports the user in maintaining a scrap library, providing simple ways to add new scrap, automatically erasing integrated scrap from the database, and providing guidance for identification and location of binned scrap objects. This effort in return provides cost savings to the user, as well as contributing to more sustainable fabrication.

We contribute the idea of re-using many types of scraps as 3D printed infill, and a working system enabled by a novel combination of a custom add-in, slicer modification, a purpose-built end-to-end database, and a novel extension of a geometry processing algorithm.

3.1 BACKGROUND AND RELATED WORK

Prior work has established several techniques that make fabrication more sustainable, though often as a by-product of trying to speed up iteration. We discuss prior work through the lens of sustainability, even though most of these works were motivated by speeding up iteration or increasing print sizes.

3.1.1 *Optimizing prints to reduce material usage*

Developing ways to reduce the amount of material necessary to print a model saves time and money, and indirectly also increases sustainability. For example, systems from Mueller et al. [67] and Peng et al. [79] approximate the object's surface, by printing a wireframe version of the geometry, but this reduces fidelity and is therefore only applicable for certain kinds of rapid prototyping. If object detail is important, material consumption can be reduced by optimizing the required infill to support all exterior walls of the object [46], or to optimize only for expected loads [33, 120]. Our system follows a different approach to reduce infill and can be applied in conjunction with infill optimization for even greater material savings. It furthermore works not only for prototypes that approximate the geometry, but also for production quality prints.

3.1.2 *Recycling filament material*

A different approach to sustainable fabrication is to directly recycle the material of prints that are no longer in use. It has, however, been shown that the mechanical material properties of recycled materials degrade with each recycling iteration [77], and the recycling process itself is energy intensive. Systems that are deployed in changing environments can be made more sustainable by employing a modular system design that allows reusing most of the parts while only fabricating and switching modules when circumstances change [96].

3.1.3 *Printing onto objects*

Attempting to fit a fabricated part to an existing object often requires multiple iterations before achieving the desired fit. This process can be sped up by incorporating the target object directly in the fabrication process. Reducing the amount of iteration also makes the process more sustainable. Integrating existing objects into the design and fabrication of 3D models has been explored in various ways. Encore [18] is a method to print on top of existing objects to enhance them, and to print through loops of existing objects, such as the handles of scissors. To do so, the loop is placed inside a printed trough during fabrication, before printing on top of the trough which closes a second loop through the first one. Other projects aim to customize the fabrication process to specific objects, for example by integrating a touch screen into the

fabrication machine that allows capturing the objects outline and modelling around it directly in Fitter [3], or by pushing an object into resin actively being cured to create a perfect mold or fixture around it [127].

3.1.4 *Inserting objects into prints*

Printy3D [125], HotFlex [34] and other systems [22, 27, 97] incorporate electronic components and physical widgets to enhance the functionality of fabricated objects. Similar to our system, these components are inserted during printing. Medley [17] creates fittings during the modelling of the object that allow adding parts such as screws, metal wire, or sandpaper after fabrication to enhance the object's mechanical properties. Our system also involves integrating foreign objects, but for a different goal. To investigate the design of nesting objects, Jacobson created a generalized way of approximating the maximum scale at which one model fits within another [41]. Our inclusion engine is based on their algorithm, but solves a different problem. Both will be discussed further in section 3.4.2.

3.2 FORMATIVE INTERVIEWS

To help guide our work, we interviewed two experts who run makerspaces (P1, P2) and two people with makerspace experience who now work in industrial fabrication (P3, P4). We discussed challenges in fabrication and prototyping, and attitudes and practices regarding sustainability in fabrication. The interviews were one hour long on average. Three participants were male, and one was female.

SOURCES OF WASTE We found that makerspaces are regularly visited by people who do not have a lot of experience making things and who do not own fabrication equipment themselves. A lack of experience makes it more likely to encounter problems. They may require more iteration before successfully fabricating their intended design. Makerspaces thus generate lots of waste (P1, P2).

Expert users, especially those who run makerspaces, care about being sustainable (P1-4). Makerspace managers (P1, P2) recycle what they can, and store the rest. One of the makerspaces (P1) essentially stores all PLA waste indefinitely, since it can not be recycled easily by their municipality. Re-melting waste into filament to create new spools would be beneficial, but it is difficult. Waste material first has to be ground up into small shreds and its thickness has to conform to tight tolerances. The machinery required to create quality filament would require an investment comparable in size to a small printer farm, and none of the makerspaces frequented by our participants had such equipment. One participant (P4) heats up and presses old filament from support material into plates for laser cutting, but these plates are smaller than desired.

Much of the stored material has a "recognizable shape" (P1), so repurposing some of it seems feasible. Makers often hold on to failed prints and early prototypes (P1, P3, P4), sometimes until they move (P4). They keep these objects as a reminder to avoid specific printing issues (P1, P4), out of sentimental reasons (P4), or in the hope that they can one day reuse them somehow (P1). All four participants stored some kinds of waste material like failed prints or off-cuts at least temporarily, and when asked, all were willing to store failed prints and other scrap for longer if they could benefit from them in the future.

PRINT TIME IS VALUABLE Since there are usually more users than machines in a makerspace, makers often have limited access to machines. Printing time is as much of a valuable resource as the filament itself. The managers of the makerspaces (P1, P2) provide guidance, training, and proven printing settings to those with less experience. They try to optimize these printing settings for speed and to minimize material usage, but increasing the chance of successful prints usually still takes priority. They encourage makers to use other prototyping methods, such as making sketches and paper prototypes, to make low fidelity prototypes, and to print models at a smaller scale first. Makers however rarely make use of these methods, as "they need permission to make it low fidelity." (P1). Makers still require two to twelve iterations before finishing a project, with more experienced makers requiring fewer iterations than novice makers.

COLLABORATION Makers often collaborate on their projects, for example, by sharing printing tips and by brainstorming on how to realize challenging designs. They prefer watching the printing process or at least being in the same room to be able to intervene quickly in case of a printing issue (P1, P3, P4). They even collaborate with strangers, for example by asking someone else to watch their print while they are away (P4). Most would even intervene when they notice an issue without being asked to watch over the process, or empty the build plate and start a queued up print of another maker.

AVAILABLE SOFTWARE AND HARDWARE Our participants have experience in other types of fabrication beyond FDM printing, for example with stereolithography resin printing (P2, P3), using CNC routers or laser cutters (P1, P2, P4), or designing clothes digitally (P3). They use a variety of software tools to support them in their fabrication efforts, and all of them were experienced in Autodesk Fusion 360 [8]. Their experience was not limited to digital fabrication either, and all had experience in other domains like woodworking, machining, electronic repairs, and so on.

SUMMARY Our interviews showed that makers and makerspaces store a variety of essentially useless objects. They desire faster processes and being more sustainable, both in using less material and in finding ways to reuse their waste materials. They also regularly observe fabrication processes and are willing to interact with the processes of other makerspace users within

the same space. Fusion 360 is a commonly used 3D modelling software in makerspace environments. These results were encouraging, and we tried to incorporate ways to repurpose common waste materials with recognizable shapes, such as old and failed prints and off-cuts, as we designed our system.

3.3 WORKFLOW AND EXAMPLES

We walk through how our system works, using the example of designing and printing a Tardis (the phonebooth model in Fig. 3.2). Before beginning the project, this user or others have populated a library of 3D models of scrap objects. These could be past prints or objects that might otherwise be thrown out. In our example, we assume the model of an unwanted salt-shaker was entered using a procedurally generated *jar* geometry.

The user begins the project in Autodesk Fusion 360 by modelling or loading the Tardis. Our Fusion 360 add-in examines the Tardis geometry and the geometries of all available scraps and recommends feasible scrap objects that could be inserted within the Tardis model. This recommendation can either occur in the background as the user updates the Tardis geometry, or on demand by clicking a button. The add-in provides previews for how different fitting scrap objects could be integrated within the Tardis, such as the unwanted salt-shaker. These different options are sorted in descending order by their volume to make it easy to identify the most beneficial insertion objects. The previews also allow users to identify and avoid inserts with potential printing issues, such as gaps around the inserted scrap that are too wide, or to make changes to the model that could accommodate scrap alignments that are easier to print. Our algorithm for determining insertion feasibility ensures that the scrap material does not stick out of the model and collide with the moving print head.

The user can then either load a preview scrap object directly into the scene to inspect its alignment further, or request the hollowed geometry: the original exterior of the Tardis with an embedded inner cavity snugly fitting the scrap. Both identifying possible scrap inserts and generating the hollowed meshes are performed on a dedicated server. This separation of tasks ensures that the 3D modelling process runs smoothly, even while potentially computationally intensive procedures are performed in the background. The user decides to print the Tardis using the first scrap shown in Fig. 3.2 (b).

Our add-in exports the exterior and interior surfaces to our scrap-aware 3D printing slicing algorithm, and the printing begins. The generated G-Code commands include a PAUSE command at the moment when the user should insert the scrap, moving the print head out of the way for clearance. Precise insertion instructions are simultaneously visualized on the printer's display screen. The user inserts the scrap and confirms that printing should resume.

The final result is a 3D printed model of a Tardis with a salt shaker embedded inside, saving time, material, and energy.



Figure 3.2: Fusion 360 integration: (a) our add-in panel displays a list of compatible scrap with previews; (b) detail of previews showing compatible scraps that would fit into a large model, such as a plastic salt shaker, a beer can, a soda can, or a cardboard packaging box.

3.3.1 *Populating the Scrap Library*

The scrap library can be populated by uploading an existing 3D model, using the failed print generator, using one of the packaging generators, or the plate outline generator. In the previous example, the salt shaker mesh was created using the jar packaging generator.

UPLOAD EXISTING MODEL When a 3D model of the scrap object readily exists, we can simply load it into the library. This covers a large number of scenarios: previous successful prints, objects with standardized or well-known geometries found online (e.g., CD cases, bottle caps, GameBoys), or 3D models acquired from unique physical objects through 3D scanning.

FAILED PRINTS 3D printing does not always succeed. As a result, most makerspaces have an abundance of partial prints and failed prints. Our scrap library captures a common type of failed print, where printing proceeds successfully up until a point of failure. A variety of issues can lead to this kind of failure, such as a clogged printing nozzle, running out of filament, or the print getting knocked loose off the build plate. For other types of failures that continue printing, but with corrupted geometry, a user could cut through a failed print along the z-axis plane below the height at which the error occurred, only using the lower half with known geometry for scrap. In either case, the digital 3D model is faithful to the physical artifact until the height reached along the print axis of the failure or cut. This distance can be determined by the time of print failure or simply by measuring the height. Based on this height, we intersect the original 3D model with a half-space, recovering just the portion of the successful print. Debris (e.g., "FDM spaghetti") should be pruned before including in the scrap library. Figure 3.3 shows the result of such a printing failure and the output of our generator.

PACKAGING GENERATORS Many common objects have a simple shape parameterized by a few easy-to-determine measurements. For example, disposable coffee cups are generally conical and well approximated once the bottom and top radii are measured with a ruler or caliper. We provide web-

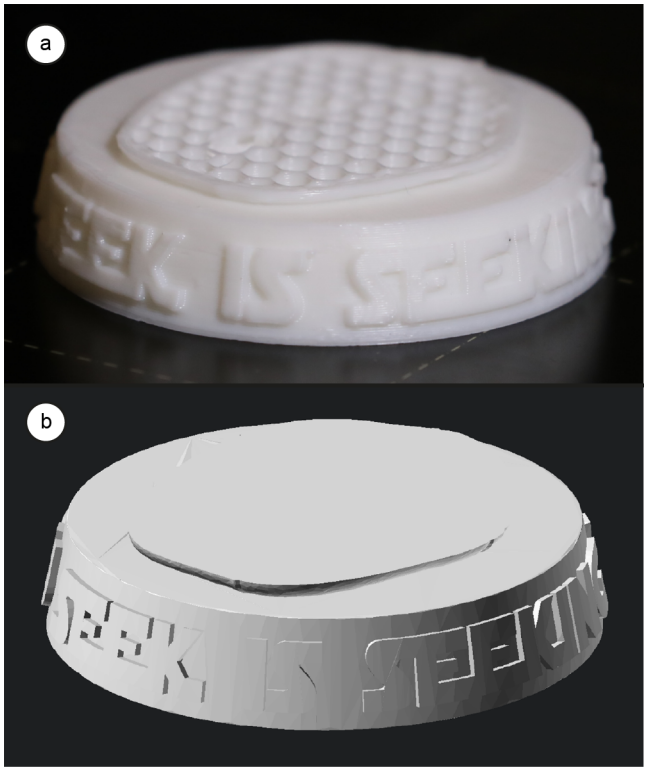


Figure 3.3: Comparison between the result of a failed 3D print and the geometry created by our generator: (a) A 3D print that failed mid-fabrication. (b) The mesh generated from our failed print generator.

based 3D-model generators for seven types of objects: boxes, jars, lids, two different types of cans, and two different types of cups. These generators use descriptively named measurements to approximate any of these common forms of one time packaging, including options for generating rounded corners and slopes.

Some types of packaging have large internal voids. For the purpose of inserting them as scrap it can be beneficial to treat these voids as filled with material as well. Simple physical scrap modifications can avoid potential bridging issues caused by the voids. For example, adding two strips of masking tape across the opening of a coffee cup provides enough support to print upon and bridge the void. Our packaging generators contain export flags to treat internal voids either as hollow or as filled to enable such scrap modifications.

PLATE OUTLINE GENERATOR Many 3D shapes (e.g., wood off-cuts and laser cutting scraps) are one-dimensional extrusions of a 2D shape. For such scraps, we can generate 3D geometry by allowing the user to input a 2D outline (e.g., measured with a flat bed scanner or a camera-phone measurement app) and a height value. Precise digital outlines may already be available for off-cuts from laser cutting.

We provide an initial set of common scrap objects. This library of models can be accessed, altered, and extended via a website called Scrap Library. Our initial set contains 3D model files portraying common household garbage, such as disposable coffee cups, lids of jars, and over a dozen of different types of food containers. We furthermore initially include famous objects for 3D printing, such as a 20mm calibration cube, or the 'benchy' boat model. Also included are 3D models of common collectibles, such as an Ikea hex key, for a total of 40 models. Renderings of all 40 objects can be seen in Figure A.1 in Appendix A.

We provide software tools to maintain the library and simplify retrieval of scrap items in the makerspace or at home. Our library keeps track of customizable metadata for each scrap object. For example, users can specify how many of these objects they have on hand, or whether they generally have repeatable access to this kind of scrap, such as a daily disposable cup of coffee. The scrap library also holds an image of the object (either uploaded by the user, or automatically generated from the 3D model), and provides a unique ID that can be used to mark and later re-identify visually similar objects. It furthermore allows entering a name and a description per scrap item, such as a note where the user would usually keep this item stored, and finally also the option to upload a picture of the item in its storage location.

3.3.2 *Examples*

Our system can make use of many different kinds of scrap materials and geometries. The example objects that we fabricated highlight some of them. The example objects show a range of use cases, such as decorative models, tools, parts of mechanisms, and printer maintenance.

The example shown in Figure 3.4a inserts an outdated model into a new one. Geometries that have a flat surface on one side are well suited as scrap objects. Many 3D models have a flat bottom which securely affixes the model to the build plate during fabrication, which makes them well suited for inserting as scrap. Figure 3.4b demonstrates a jar lid inserted into a gear. The shape of the lid forms a bridge-like shape. During FDM 3D printing, support material is often used to create bridges, which supports perimeters that are printed at a steep angle, or that are disconnected from the rest of the model. This scrap performs a similar function. The broken handle inserted into the scraping tool shown in Figure 3.4c performed a similar function to the model that it is inserted into. Geometries of objects that perform similar functions are likely to resemble each other. As such, it is likely that such as scrap object is fitting well into such a model.

Figure 3.4d shows a model that is often printed repeatedly to calibrate 3D printers. With many flat sides and a large volume compared to its surface area, this geometry is likely to fit many different models. Since this type of model is being printed repeatedly, it can also be used to print a larger version of the same model using a comparable amount of material as the original print. Since this model is used for calibration purposes, it needs to maintain

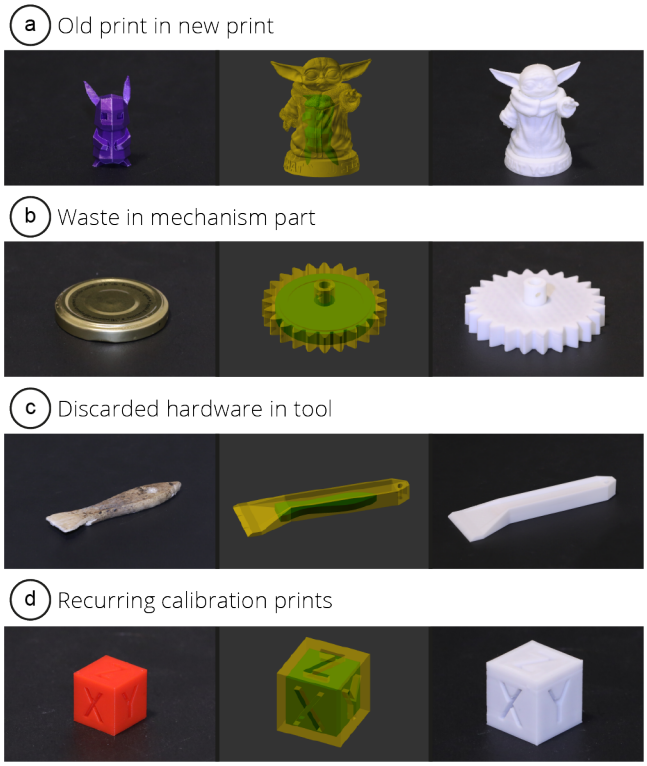


Figure 3.4: Examples showing scrap, computed mesh, and printed model: (a) an outdated print inside a new one; (b) household waste inside a mechanical part; (c) the broken handle of a hand tool inside another tool, (d) and a recurring print used to calibrate printers inside a bigger version of the same model.

truthful printing dimensions and the look of the outer walls of the object. Our system does not alter outer walls, and we found that in practice, our algorithm and slicing manipulation creates a snug fit that does not flex the outer walls.

3.4 SYSTEM IMPLEMENTATION

Each stage of our system would be difficult or time consuming to reproduce using existing tools or with manual effort. For example, populating a scrap library without generators would require resorting to expensive, time consuming, and error prone 3D scanning. Determining the best scrap object that fits into a given target shape is an intractable problem to solve with a standard 3D modelling tool. Small changes in the orientation and position of the inner object can cause the scrap to pierce the target shape. Finding a configuration where the scrap fits feasibly inside the target shape is not even a sufficient condition. It must be possible to *insert* the scrap object without being blocked by already printed parts of the target shape and it must be possible for the print head to move around to continue printing the remaining target

shape after insertion. Given a feasible placement of the scrap, the 3D printer's instructions must be adapted to leave a hollow space affording insertion: again, not just the hollow space of the inserted scrap, but a hollow *path* for the scrap to move along. Naively sending an outer surface and inner surface to a standard 3D printing slicing algorithm could actually result in *longer* printing times and increased material usage. This is because standard slicers will reinforce *all* surfaces with shells to ensure a clean surface finish. Our modified slicing routine removes these material and time intensive print lines for the inner (non visible) surface. The synthesis of these user interface, algorithmic and design contributions combine into a time, energy and material saving system.

3.4.1 *Fusion 360 add-in*

Our system is integrated into the 3D modelling process in the form of a Python Fusion 360 add-in. Using an add-in allows users to benefit from the full modelling capabilities of popular CAD software, while also seamlessly integrating our technique. The add-in creates a new side panel within the Fusion 360 GUI. The panel shows previews of potential scrap able to fit inside the model edited. To gather possible scrap inserts in the first place, the add-in handles exporting the open 3D model, simplifying and potentially repairing it, and sending these files to the server side of our system. Upon receiving scrap insertion options from the server, the add-in creates preview images and displays those to the user.

The user may load any of the scrap meshes that were discovered to fit the current model directly into the active modelling scene to inspect it further, or to use it as reference while continuing the modelling process. After making a decision which scrap to include, the user can request the hollowed meshes from the server, if they were not already provided automatically in a background process. As a final step within Fusion 360, the user may then export the hollowed meshes, including the information when to pause during the printing process to allow inserting the scrap. While exporting, our add-in also provides guidance on how to locate stored scrap, as well as a link to the scrap library for the specific scrap item. The scrap database is then updated to record that the scrap items was used.

Most server communication and related computations are handled in asynchronous background threads to make sure that Fusion 360 remains responsive while scrap inserts are evaluated, or while the hollowed meshes are generated. Since most of the add-in functionality runs concurrently, all scrap operations are associated with time stamps and revision numbers to ensure that the add-in never presents outdated data to the user. Some of the functionality implemented by the add-in makes use of external open-source software. Preview images are rendered by invoking OpenSCAD [52] via the command line. Similarly, a Python script invokes Blender [29] to simplify and repair meshes. The add-in stores local copies of scrap models to minimize network traffic, only transmitting newly added files or files that have been

modified. It also keeps a persistent storage of scrap metadata using JSON files to minimize start-up times of the add-in after the first use.

3.4.2 *Insertion engine*

Given a library of scrap models, our goal is to present the user with an ordered list of scraps that can be safely inserted into the target shape during printing. A valid scrap must not only fit within the target shape geometrically, there must also exist an insertion path for the scrap without crossing the target shape's surface. Furthermore, we must identify the time of insertion during the print so that the part of the scrap "sticking out" after insertion is so minimal that it doesn't collide with the print head when printing resumes.

Although quite distinct in motivation, this geometric problem bares many resemblances to the "Generalized Matryoshka" algorithm by Jacobson [41]. In that work, the goal is to create Russian nesting dolls for arbitrary input shapes: the outer object is split in half and the inner object is inserted within it. Like our scenario, their algorithm considers not just feasible containment, but also feasibility of the insertion path. Distinct from our problem, Jacobson's method seeks to *maximize* the scale of the inner object to achieve tight nesting, while our inner object has a fixed scale as it already exists in the physical world. Jacobson's method also has no requirement that the inserted object must not stick out too much.

We propose several modifications to Jacobson's algorithm so that we can leverage their overall modelling of the problem and optimization technique. For readability, we give a brief overview of Jacobson's method, but refer the reader to that paper [41] for details.

Jacobson's algorithm first assumes that the inner object will be inserted along a straight trajectory (pure translation). Since the path of each point on the inner object follows a straight line, the problem can be recast as a visibility problem and Jacobson's method implements the following algorithm on the GPU for maximal parallelization. The placement of the inner object is parameterized by the 3D position of its center of mass and its 3D rotational orientation. The insertion action is parameterized by the 3D cut plane (in Jacobson's case, literally where the outer object is split in two; in our case, the height at which printing is paused) and 3D direction vectors describing insertion into the top and bottom halves. Given specific values for these altogether 16 parameters, Jacobson's visibility based algorithm returns the maximal scale at which feasibility occurs (returning zero if no solution exists). In an outer loop, Jacobson's method searches the 16-dimensional space using the particle swarm optimization method [49] to maximize the scale parameter. Particle swam optimization is a global optimization method that does not require the objective function (maximum feasible scale) to be differentiable. While many other global optimization techniques exist, Jacobson reports this method works well for the low-dimensional matryoshka problem. The core subroutine that is needed must take as input a candidate configuration

and output the maximal feasible scale. The optimization returns the largest encountered feasible scale after a fixed number of iterations.

Our problem can be understood as a modification of the Generalized Matryoshka problem; surprisingly, the extra constraints and assumptions made in our problem will in many ways simplify the problem and improve performance.

We assume that the user has determined the print direction for the outer object. This is reasonable and welcome as 3D printing surface quality, material strength and use of support material depend on the print direction. Since 3D printers proceed in layers perpendicular to the printing direction, this assumption implies that the "cut plane" must also be perpendicular to the known printing direction. Reducing the four parameters of the cut plane to a single degree of freedom. Similarly, we assume that the object will be inserted along the printing direction (i.e., straight down), eliminating the insertion direction parameters.

We must prevent too much of the inserted scrap from sticking out above the cut plane, so we short circuit the optimization for any configurations where the maximum distance of the inserted object is too far (0.6 mm for our setup) and return zero. This quick check not only avoids the more expensive visibility check, but will discourage the particle swarm optimization from exploring the vast regions of the configuration space where printer-head collisions would occur.

Unlike Jacobson's problem, we ultimately do not want to *maximize* the scale of the inner object. Rather we want to find *a* feasible configuration where the scrap object can be inserted at 100% scale. Figure 3.5 demonstrates the feasibility of configurations. For a given configuration, Jacobson's proposed method conducts a binary search to determine the largest scale possible that passes the visibility test. If we ever identify that a configuration is feasible at 100% scale, then we terminate the optimization early and return this solution. If the particle swarm optimization concludes without finding a configuration at 100% scale, then we declare that this scrap is not feasible and do not present it as an option to the user.

Since we must determine feasibility for each scrap in the library, we first conduct two inexpensive culling procedures before invoking the modified-Matryoshka optimization. If the scrap's total volume is larger than the total volume of the target shape, we can reject immediately. For both shapes we construct a tight fitting oriented bounding box, and then exhaustively check whether any of 10000 random rotation of the scrap's box would fit inside the target's box with a 10% and 20% margin of error for the longer and shortest dimensions, respectively. While not strictly conservative, this simple test is extremely fast to compute and culls away most of the "obviously" infeasible scraps.

As proposed by Jacobson, we also pre-dilate the inner shape to account for 3D printing accuracy tolerances and pre-erode the target geometry to account for minimal wall thickness. When a scrap shape is selected by the user, we generate the interior surface geometry by translating the swept volume of the inflated scrap geometry along the insertion direction. Figure 3.6 demonstrates

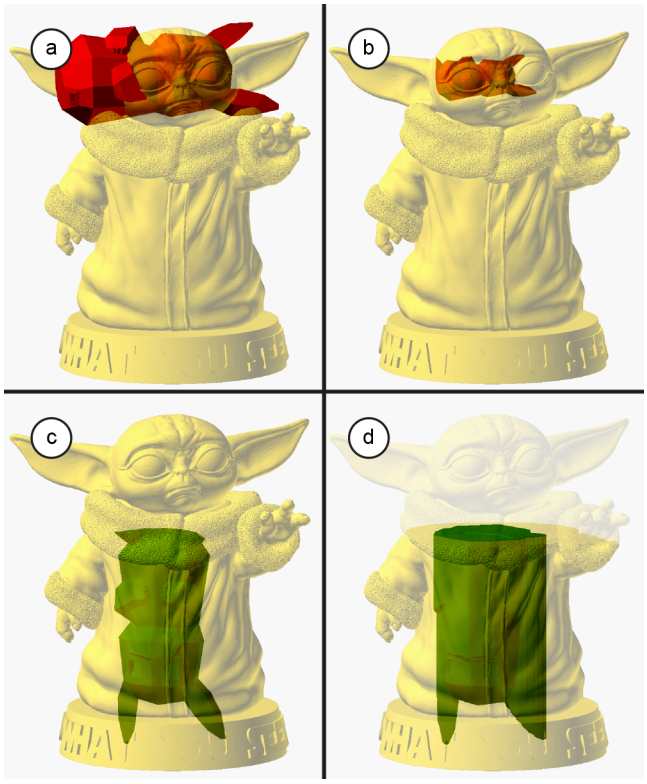


Figure 3.5: The optimization process: (a) This random initial alignment is not feasible for including the Pikachu scrap mesh in the Baby Yoda model, since the scrap mesh protrudes the model mesh. (b) At this alignment, it can only be scaled up to 45% of its initial size before colliding with the model mesh. (c) The optimization concludes after finding an alignment for which the scrap fits the model at its original size and can be inserted without collision. (d) Additional volume has to be removed from the model mesh to achieve an unobstructed insertion path through this insertion plane.

this process. This swept volume's surface is tagged as an "inner shell" so that our modified slicing algorithm can process it distinctly from the outer shell of the target shape.

3.4.3 *Custom slicer*

Our customized slicer is a modification of the Kiri:Moto slicer [6]. Typically, this and all other standard slicers will proceed layer-by-layer determining all intersections of the layer plane with the input surface geometry. This intersection curve is traced with multiple high-density "shell" paths and then the interior between shells is filled with a low-density, fast-to-print pattern (e.g., honeycomb). Shells ensure quality surface finish, but we do not care about this for the internal boundary between the fill pattern and the yet-to-be inserted scrap. Since we have tracked which surface parts correspond to the target shape and the inserted scrap's swept volume surface, we post-process

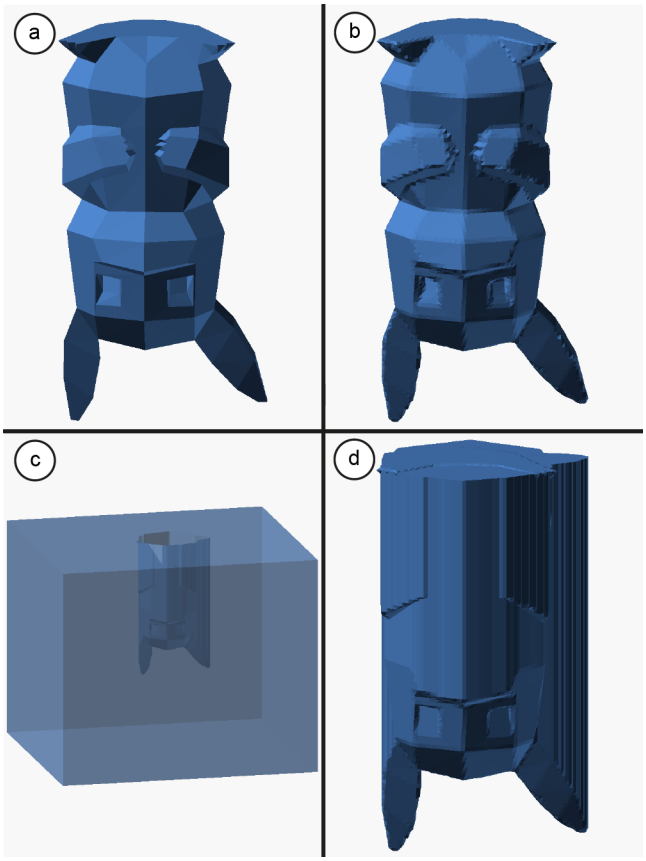


Figure 3.6: Hollowed meshes generation: (a) The aligned scrap model. (b) The scrap model is inflated to account for printer inaccuracies. (c) The scrap mesh is projected towards the insertion plane and the resulting volume is cut from the surrounding material halves. (d) A mesh of the cavity that has to be removed from the model mesh to allow the scrap insertion.

the layer paths from Kiri:Moto and remove the paths corresponding to shells of the inner surface (see Fig. 3.7).

The slicer generates a G-Code file that serves as instructions to the 3D printer. The "cut plane" determined during the modified-Matryoshka optimization above corresponds to a specific height along the printing direction. We identify G-Code corresponding to the layer closest to this height and insert commands to stop printing, move the printer bed to an easily accessible position, move the print head out of the way, and play a beep melody to alert the user that it is time to insert the object. The LCD screen on the printer is triggered to display customized instructions and prompts the user to confirm when the scrap is inserted so that printing may resume.

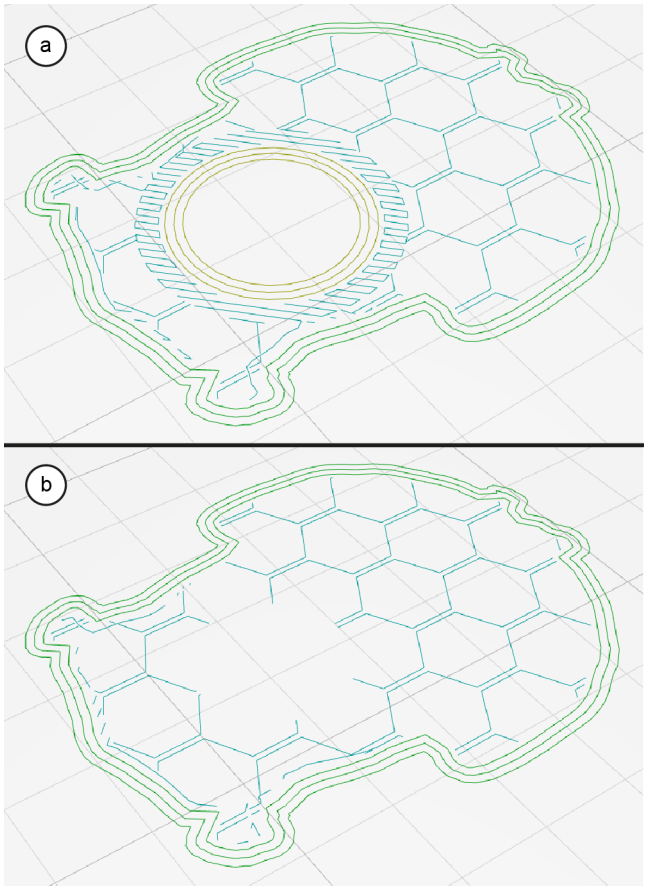


Figure 3.7: Comparison of the slicing result of Kiri:Moto and our customized slicer: (a) The toolpaths generated by the original slicer contain internal walls and supports. The color of the internal walls signify that these lines will be printed slowly. (b) The toolpaths generated by our slicer do not contain internal walls or supports surrounding scrap.

3.4.4 *Scrap library*

The usefulness of our system hinges on whether a user can find well-fitting scrap for their planned 3D prints. The chances of this happening can be increased by having a large and varied collection of possible scrap objects. Users must, however, also be able to locate specific scrap objects. We provide a number of web-based tools that together form our scrap library to support maintaining and extending a collection and to identify and find specific scrap again later.

Not every user will have access to the same types of scrap objects, and even for a single user their access to certain types of scrap will vary over time. One of the maintenance use cases for the scrap library is therefore to keep it up to date and aligned with the user's current scrap availability. To do so quickly, we include an overview page that shows images of the scrap objects and grants direct access to the availability status of any scrap object. This list

can be filtered, for example by the method with which the scrap was entered into the data base. It can also be ordered, for example by scrap availability or volume. This way, users do not have to click through detail pages of every single scrap object, but can make adjustments in bulk.

We provide scrap generators that can be used to generate scrap meshes, instead of uploading a mesh directly. While entering parameters, the generator website already displays the scrap ID that will be assigned to this new library item. This can be useful to mark objects that might be hard to identify again later, such as wooden off-cuts. Users may also enter their own scrap name, a description of where they store this kind of scrap, as well as a picture of the storage location, to help identify and locate the scrap again later. All of this information can be found on a scrap details page. If this scrap was recently exported to be included in a 3D print, this page also contains an animation of how to insert the scrap. This animation shows a partially transparent version of the surrounding model up to the insertion layer. It also shows an animation of the scrap object being inserted into the model along its insertion path.

3.5 EVALUATION

To test our system, we created a set of seven example objects. Each of them was printed with and without scrap inclusions to compare printing time and material usage. All but one of the models were downloaded from Thingiverse [61]. The phone stand was modelled by one of the authors. Some of the models were altered slightly before printing, such as scaling the model with the scrap preview in mind.

The scrap objects that were added to the scrap library are based on unused items that some of the authors collected randomly over time, and common household waste items that were stored instead of being thrown out during the time the paper was written. The household waste items were added to the library using our generators. The collectibles were added by downloading appropriate 3D models from online databases like Thingiverse or by modelling them by hand. Some items in the scrap library are models previously printed by the authors. These items were added by uploading the mesh to fabricate them originally.

All models were printed in PLA at 185 °C using the default slicer settings for infill and perimeters. We chose not to alter these settings as novice users are likely to rely on predefined print settings. The benefits of our system would be more or less pronounced based on higher or lower numbers respectively for infill percentage and number of perimeters. Even at 0% infill, our system can still provide some benefit, since it can replace parts of inner perimeters. The default infill setting of the Kiri:Moto slicer is a 0.5 percentage hex pattern. This equals around 20% infill in the PrusaSlicer [84] software and between 15 to 20% in the Cura [108] slicer, both of which are among their default settings as well. It is not possible to provide a more detailed comparison to the Cura slicer, as it does not offer a hexagonal infill pattern. The default number of

Table 3.1: Printing time and material weight of our example objects. Blue bars denote percentages of time or material saved when using our system.

Model	Scrap	Print time without scrap	Scrappy print time savings	Material weight without scrap	Scrappy material weight savings
Tardis	Salt dispenser	1730 min		285 g	
Gear	Jar lid	150 min		25 g	
Fossil token	Rock	55 min		10 g	
Large XYZ-cube	XYZ-cube	33 min		7 g	
Baby Yoda	Pikachu	472 min		52 g	
Phone stand	I-Pod	272 min		58 g	
Scraping tool	Broken handle	160 min		33 g	

Table 3.2: Average run times and standard deviations for validating insertion options for 40 scrap meshes and the time it took to generate the hollowed mesh for printing the chosen insertion option.

Model	mean validation time	standard deviation	mesh generation time
Tardis	8.6 s	10.9 s	19 s
Gear	3.9 s	10.7 s	14 s
Fossil token	1.7 s	3.4 s	126 s
Large XYZ-cube	1.0 s	1.7 s	17 s
Baby Yoda	4.1 s	6.1 s	84 s
Phone stand	12.1 s	23.1 s	32 s
Scraping tool	2.2 s	4.8 s	114 s

perimeters in Kiri:Moto is three, which is again among the default settings in PrusaSlicer and Cura.

Printing times in Table 3.1 include the time required to insert the scrap while printing. Printing these models with scrap objects inserted during fabrication saved on average 29.4% (SD = 13.9%) material. Printing times decreased on average by 26.4% (SD = 15.6%).

We evaluated the time that our algorithm requires to validate a scrap mesh. This evaluation was performed on a library of 40 scrap meshes. It was performed while running six parallel threads validating scrap meshes. The mean validation and mesh generation times for each of the seven example models are shown in Table 3.2. Validating the whole library of 40 scrap meshes took 32 s on average (SD = 4.1 *s*) among all examples while running on six parallel threads. Generating the hollowed meshes for our seven example objects took 58.0 s on average (SD = 48.7 *s*). Note that this process only has to

be performed once when the user is ready to print. It replaces a traditional export step at the end of the design phase and is quite short compared to savings in printing times in the order of hours.

3.6 DISCUSSION

Though our system provides speed and cost benefits to users, it might also cause some detrimental effects in certain situations. For example, the mechanical properties of the resulting 3D prints can vary widely based on the type of scrap material that was integrated. Inclusions made from wood or metal are likely to increase compression and bending resistance of the model, but the tensile strength of the part may still be reduced due to the inclusion of scrap. This is mostly due to the way the scrap is held in position on the inside of the model. For scrap made from similar thermoplastics as the printing filament, chemical bonding can occur. This bond is limited to one side of the scrap and most other materials will not bond chemically to the printing filament. Since the scrap object is surrounded narrowly on all sides, it still fits tightly inside its cavity. Also, the common FDM quality issue of stringing actually benefits our system, as the strings inside the cavity provide a tense cushion once scrap has been inserted, tightening the fit further. Adding a drop of glue during scrap insertion could further alleviate potential issues with stability of fit.

Our system prevents collisions of the print head with the inserted scrap object by restricting the insertion plane to the top of the scrap object. Knowledge about the geometry of the printer and the print head in particular could be used to generate the reachable volume after scrap insertion. Though having the scrap object reach high above the last printed layer would make many geometries impossible to complete, having more flexibility in choosing the insertion plane would enable the fabrication of others.

Since a scrap might spend a long time inside the model object, not all waste items may be suitable for insertion. Perishable items could decompose inside the model, putting the part under internal pressure, or slowly melting it from the inside. Adding breakable scrap, such as glass jars, could lead to a more dangerous failure mode of parts that come under mechanical stress.

Including multiple scrap objects in a single model would enable replacing even more infill, but would require additional user interactions or additional computation. Short of using a commercial cloud-based solution for the computation, this computational effort of trying to fit two or even more scrap objects into a model at the same time would likely increase the run time of the search too much to integrate this feature into an interactive modelling session.

The usefulness of some objects to be used as scrap may be dramatically improved by breaking them apart. For example, most of the volume of the Pikachu model is located in its body. It is much easier to find suitable scrap placements for the two separate parts of a Pikachu whose tail had already snapped off. We would like our system to be able to recognize

such circumstances and instruct the user to cut or break off specific parts of otherwise unwieldy scrap models on demand.

3.7 CONCLUSION

Based on a formative interview study with fabrication experts, we verified that material is often wasted on failed prints and early prototypes, print time is a limited resource, and makers care about sustainability. With this motivation, we built a system for FDM printing that makes use of these and other waste materials by using them to replace infill material. We extended an existing algorithm to determine possible scrap inclusions, and integrate this information into an existing CAD modelling workflow using our Fusion 360 add-in. Our method includes custom slicer adjustments to avoid printing unnecessary internal walls. We created an extensible web-based tool for makers to maintain a scrap library which provides instructions for locating scrap and integrating it during fabrication.

In future work, we plan to compare different methods of recruiting people nearby to insert scrap, such as printing a QR code that links to a website with instructions, or printing a virtual post-it note. We will also compare different fidelity levels of instructions, such as a written message, an instructive image, or an instruction animation, and we will contrast their performance when used by people nearby to the maker who started the print. Furthermore, we plan to extend our system to replace external support material using ad-hoc supporting structures. It would also be interesting to extend our system to evaluate the added benefit and feasibility of insertions that perform additional functions beyond saving infill material, such as inserting electronics or materials with mechanical benefits.

3D printing has the potential to decentralize the production of many tools and consumables, which are traditionally packaged and shipped around the globe. This decentralization has an important sustainability outcome, the reduction of packing materials and lowering emissions from transportation. However, 3D printing itself can generate significant waste. Our system contributes to make the 3D printing process itself more sustainable.

3.7.1 *Summary*

The system presented in this chapter builds a foundation for getting both machines and people to handle and work with arbitrary physical objects and their digital counterparts. After enabling a single interaction, we extend our exploration of the user involvement spectrum of intermittent interactions to a variable range of interactions. In the next chapter, we contribute a second technical solution to reduce print time and material with unmodified fused deposition modelling printers. The approach uses ad hoc objects inserted by a user during printing as a replacement for printed support of overhanging structures. Examples of objects include household items like books, toy bricks, and custom mechanisms like a screw jack. A software-only system is inte-

grated into existing slicing software to analyze generated support print paths, search a library of objects to find suitable replacements, optimize combinations of replacement objects, and make necessary adjustments to impacted printing layers and paths. During printing, the user is prompted to insert objects with the help of lightweight printed holders to guide placement and prevent movement. Instructions printed on the build-plate help identify and position objects. A technical evaluation measures performance and benefits with different sets of ad hoc objects and different levels of user involvement.

4

SUBSTIPORTS - OPTIMIZING MULTIPLE INTERACTIONS

3D printers that employ fused deposition modelling (FDM) use printable support structures to reliably manufacture models with overhangs, bridges, and other geometry extending more than 45° from the Z-axis (we collectively refer to all of these as "overhangs"). Disposable support structures are pure waste, costing additional material and time, so it is desirable to minimize them.

One strategy is to optimize the support structure itself, for example, using lightweight structures that branch out from lower parts of the model to support overhangs [26]. However, this introduces additional connections between support material and the model, marring the surface finish. In general, optimizing support geometry increases the chance of failure: thin tall structures are more severely affected by thermal deformation and their low stiffness can result in accidental displacement. For these and similar reasons, support structures typically generated by slicers are sturdy with a substantial connection to the build plate and filling much of the volume beneath overhangs [4].

Another approach is to replace support structures with something else. For example, Hongyao, Xiaoxiang, and Jianzhong [37] and Xu et al. [121] replaced a printer build plate with a mechanically actuated grid of supporting surfaces and Yiğit, Isa, and Lazoglu [123] propose using a robot arm to place a set of uniform blocks. These both replace support material with other kinds of sturdy structures, but they would add cost and complexity due to additional hardware and major printer alterations. Our approach achieves the same goal by enabling users to manually insert ad hoc objects during the printing process. This requires no special hardware or printer modifications, and there is no need to adhere to a grid or certain object type for temporary support structures.

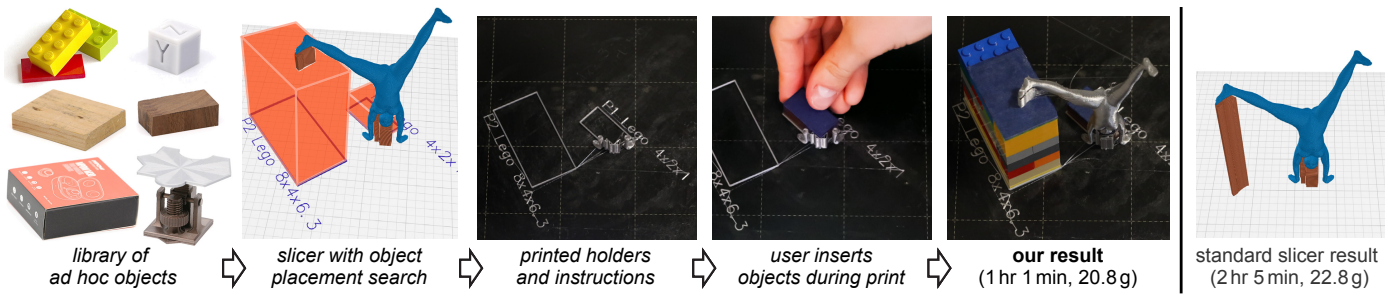


Figure 4.1: Our approach uses a modified slicer to select reusable objects to use as ad hoc structural support. Object "holders" and instructions are printed, then the user inserts objects during one or more pauses during the print. In this example, our approach reduced print time by 1 hour and saved 87% of support (2.0g out of 2.3g): even a small savings in support material can result in pronounced time savings due to reduced printer movements.

Our system is implemented primarily as modifications to an open-source slicer. After default print paths are generated for support, our system performs a geometric analysis and optimization to select and position objects from a user-maintained library (Figure 4.1a). Users can tune solutions to consider the number and complexity of object insertions, and a scheduler can further constrain insertions to convenient times. Once a solution is selected, customized G-code incorporates the replacement objects as support during the print, and adds minimal 3D printed guides to position and stabilize each object with object identification codes printed on the build plate (Figure 4.1b). By leveraging software-based optimization and human dexterity instead of a complex mechanical system, our system is compatible with unmodified FDM 3D printers and it is able to use a broad definition of ad hoc objects as support structures. An object is usable as long as it reliably provides an elevated horizontal surface to print on. Different sizes and shapes of objects may be placed at any position in the print volume, increasing compatibility with model geometry that conform to a fixed-size actuated grid or axis-aligned blocks.

A technical evaluation with a 15-model dataset achieved an average savings of 52.0% of support material and 50.4% of support print time (34.8% of overall printing time). The best performing model replaced 97.0% of support material and reduced total print time by 77.8%. Using Xu et al.'s "gymnast" model [121], our system saves as much as 93.5% compared to 78.1% for their actuated print bed approach. We make two contributions: (1) a software-only system for reducing support for a popular form of 3D printing; and (2) a technical evaluation showing the utility for different levels of user interaction. Code and data are available[1].

4.1 RELATED WORK

Trying to limit the amount of support required to achieve a successful print is a key topic in FDM 3D printing. Methods that mainly adjust the geometry of the support to reduce material usage, such as merging supports into smaller columns, forming truss, scaffolding, or tree structures [23, 26, 38, 43, 44, 56, 81, 93, 110], or filling the support volume with material efficient micro structures [101] can be used in principle with our method.

4.1.1 *Printing Adjustments to Reduce Support*

In some cases printing support material underneath overhangs can be avoided entirely. Print on Air [11, 25] creates horizontal overhangs by carefully attaching material sideways onto existing geometry by making use of reduced printing speeds and strong cooling. This method was shown to be successful for overhangs of 90° relative to the printing direction. A similar effect of depositing material sideways onto existing geometry can be achieved by adjusting the slicing process to divide the printed model into non-planar

1 https://github.com/exii-uw/substiports

slices [72]. When following these non-planar tool paths, the print head moves slightly up and down relative to the build plate. This up and down motion aids in the sideways deposition of material and can enable printing overhangs slightly above 90°, limited by print head geometry. Our system can be used to support arbitrary overhangs like the printed supports that it replaces. It furthermore does not necessitate slowing down printing speeds.

Reorienting models can be an effective tool to minimize required support material [5, 19, 24, 36], for example by bringing parts of the model that require support closer to the build plate, or by allowing parts of the model to be better supported by the model itself. Some support material may still be required after adjusting the model orientation. Users may also prefer using specific orientations to optimize part strength or the visibility of layer lines instead. Our method applies in these cases.

4.1.2 *Model Adjustments to Reduce Support*

By accepting design limitations during modelling, such as keeping overhangs below 45° and bridging less than 10mm, support can be avoided or reduced. Reiner and Lefebvre [82] developed an interactive system to apply these types of limitations to generate support-free models. However, in practice there is a large class of models that cannot adhere to these limitations. For these kinds of models, PackMerger [111] is an approach to divide a model into segments arranged on the print bed to minimize support. Since the segments must be assembled to form the original model, this method is not applicable when there are mechanical requirements and additional user effort is needed. Our method also requires additional user effort but it does not alter the input model. When printing multiple models at once, Jiang et. al [45] propose placing smaller objects underneath overhangs of larger models. This way, a smaller model acts as an elevated platform to print on, replacing some of the support material required by a larger model. Our approach is more general since it works with a single model and uses a large class of re-usable objects instead of assuming other models will be printed.

4.1.3 *Mechanical Methods to Replace Support*

Shen et al. [37] subdivide the build plate into 6 x 6 square segments, each 50 by 50 mm, which can be raised using motors to create elevated print surfaces. They report saving 35.5% of material for two models, but the proportion of support material is not specified. Xu et al. [121] simplify the complex hardware requirements for this approach by allowing all segments to be raised by a single motor after a user manually inserts lifting rods of varying heights. This also enabled a larger 11 x 9 grid with smaller 12.7 by 12.7 mm segments. Average support material savings of 64.7% are reported for five test models of simple to moderate complexity, including the "gymnast" model we include in our test set. Our approach pursues the same goal by involving users to manually provide elevated printing surfaces. Unlike our approach, Xu et al.

require a single up-front user interaction, but one that appears time intensive and complex (though user effort and time are not discussed). In addition to requiring highly specialized printer hardware, a grid-based subdivision is discrete, limiting how support can be replaced, and gaps between segments can cause surface artifacts and potential problems for first layer adhesion.

Yiğit et al. [123] propose using reusable support replacement blocks placed by a robot arm. The system uses a discretized 1 cm grid and a sliding window algorithm to place axis-aligned 4x4 cm blocks. An average of 75.5% support material savings are reported for two test models, including the Stanford bunny that is also in our test set. However, this was only a software simulation where virtual blocks were placed and stacked using a virtual robot arm during a simulated print. Practical issues like adhesion, stability, and mechanical precision are not considered. Our approach pursues the same goal, but with a fully implemented system that handles practical details.

In comparison to these works, our approach requires no additional hardware by leveraging simple human abilities. A fully automated approach introduces points of failure. For example, small debris could alter a placed block position by a few degrees, enough to cause collisions or topple towers of blocks later. Detecting and removing such debris is trivial for a person when placing a block. In addition, we use a more advanced and general optimization method to find optimal ways to place and combine heterogeneous replacement objects at arbitrary positions and orientations.

4.1.4 *Printing onto Objects*

Integrating existing objects into the design and fabrication of 3D models has been explored previously in various ways, including printing on top of other objects. Encore [18] is a method to print on top of, or through holes of existing objects to enhance them. The user places the object onto a partially printed structure midway through the print. RoMA [78] is a prototyping tool to design and print directly on existing objects. It requires extensive hardware, such as a 3D scanner and robot arm. Patching Physical Objects [104] also enables printing onto existing objects with the purpose of making physical adjustments to the model or performing repairs. Under the right circumstances this can avoid printing support material entirely by avoiding printing a new model, but also requires additional hardware in the form of a depth camera, a mill, and a 5-axis rotating platform. RevoMaker [31] prints onto a rotating laser cut base cube to avoid some types of support at the cost of design limitations and additional hardware requirements. Rivera et al. [85] suggest securing flexible objects like fabric pieces in place with double sided tape, which also works for our replacement objects.

In Chapter 3, Scrappy reduces infill material by creating a hollow pocket into which a user inserts a scrap object during a print. Only a single object insertion is supported, objects are primarily simple cuboids and cylinders, and the method requires extensive 3D geometry processing on a server to select and orient the object. The Substiports system supports multiple objects

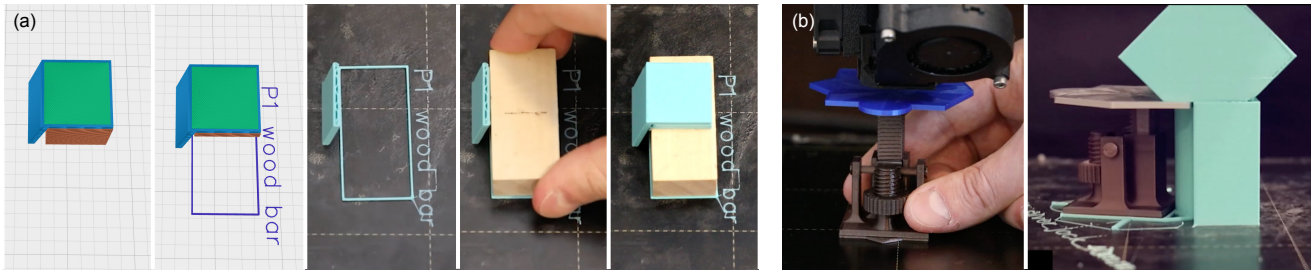


Figure 4.2: Slicing and insertion examples: (a) Angle Bracket model sliced to use a wood off-cut as a replacement object, the wood is inserted into a "holder" marked with instructions during the print; (b) the screw-jack replacement object is adjusted to match the nozzle height of the parked print head, a screw-jack used as support for the Angled Cylinder model.

of varying complexity, and it is able to run in a web browser. We also expand the object library implementation here to additionally differentiate between distinct types of objects, such as replacement objects that have a variable height. Note that infill and structure support are independent parts of a printed model, so both systems are compatible.

4.2 WALKTHROUGH

This section introduces the three main parts of the system from the user's perspective and highlights a scheduling feature. Figure 4.1 illustrates the main user steps and Figure 4.2 provides additional slicing and insertion examples.

Object Library

A library defines what objects are available to use as support replacements. A base library of standard objects includes items that people may have already, like office supplies and standard packaging, and items often generated when tuning FDM 3D printers, like calibration cubes and certain test models. The user can delete, modify, or add objects to match what they have at hand. This is done primarily using web-based generator forms, which use caliper measurements or uploaded 2D line drawings to define the key dimensions. This largely follows the process used in Scrappy. Unlike the Scrappy library, generating a 3D model mesh file is optional for most support replacement objects. For example, cuboids are the most basic type, and only their length, width, and height are needed by our system to use them to replace support. Similarly, for irregular prism shapes, it is sufficient to know the shape of the base, shape of the top, and the total height. This version of the library also supports variable objects, like a miniature screw-jack with adjustable height and stackable objects like toy building blocks.

Slicing

To prepare a print with support replacement objects, the user imports a 3D model into our customized slicer. They select usual slicing settings, with the

addition of a new setting for their desired level of involvement during printing. During slicing, the system searches the library for one or more objects to use as support replacements. The slicing result is directly modified to accommodate the replacement objects and rendered showing object positions for the user to inspect.

Inserting Objects

When printing, exported G-code includes commands to generate instructions and "object holders". When the print reaches the predetermined height where an object must be inserted, the printer pauses, parks the print-head away from the model, and beeps to prompt the user with instructions. Each object is identified by a name and ID printed on the build plate. The user locates the object, then inserts it into the custom printed holder which ensures it is aligned and secured in place. Depending on object type, the insertion process might differ. For example, a stack of toy bricks may need to be assembled to match a height printed on the build plate (i.e. number of bricks). Similarly, an adjustable model, like a screw jack, may need to be adjusted until it lightly touches the print nozzle (Figure 4.2b), which our custom G-code positions to the required support height. After inserting one or more objects, the user resumes the print, and this process of automatic pausing, object insertion, and resuming repeats as needed.

4.2.1 *Scheduling User Interactions*

The system includes a scheduler to search for solutions that avoid user interactions during certain periods of the day, for example during the night. Figure 4.3 provides illustrative examples for the Arch Bridge model shown in Appendix B B.1. Example (c) is a worst case without scheduling where the printer idles for a full eight hours before printing can continue. Example d shows a solution using the scheduler to avoid interactions between 11 pm and 7 am. This results in a much shorter print time than the worst case without scheduling, but does increase time and material from example b, which is the

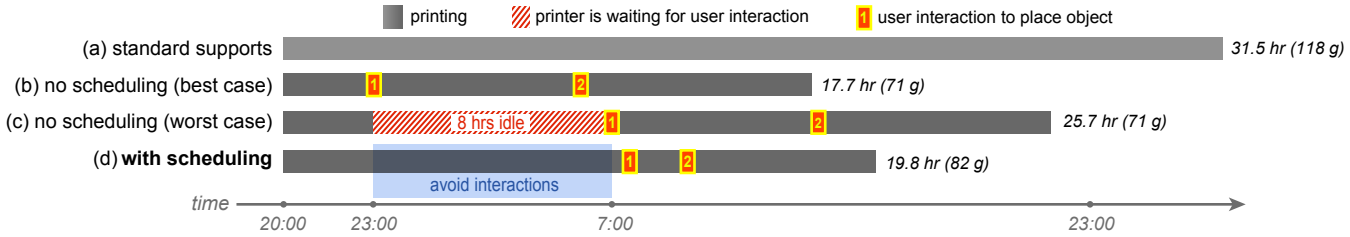


Figure 4.3: Example print times for the Arch Bridge model with print parameters held constant, with and without scheduling: (a) without our system all supports are printed; (b) an optimal solution using our system without scheduling constraints, and the user places objects during the night; (c) the same optimal solution without scheduling, but the printer idles until the user places the first object at 7am; (d) an optimal solution using the scheduler to avoid object insertions during the night.

optimal solution without any schedule constraints where the user is available for interactions regardless of time. Note that all three examples with our system, even the worst case, are still faster than example a, which is a default print with standard supports.

Scheduling restrictions are built into the system by marking solutions with interactions during unavailable hours as invalid, and by severely reducing their fitness value. To operationalize the scheduling constraint during search, the user can enter a print start time, with the current time used as default.

4.3 AD HOC REPLACEMENT OBJECTS

Our system supports replacement objects with two parallel flat surfaces, these can be gathered from a variety of sources. Found objects include books, household waste like cardboard boxes and other packaging material, or workshop scraps such as wooden off-cuts. 3D printing scrap like calibration cubes or support towers from previous prints can be reused as support replacements as well. Users may opt to create custom reusable objects that aid in replacing support material further. For example, a custom object perfectly suited to replace the support of a recurring print job that acts as a jig, or an adjustable screw jack that can vary in height.

We group replacement objects into three types based on their height variability: Objects with a fixed height, such as books or wooden off-cuts; standardized objects that can be easily stacked to form variable discrete heights, such as toy building bricks, stacks of DVD cases, or poker chips; and objects where the height is continuously variable, such as a 3D printed screw jack that adjusts in height by turning a knob.

4.3.1 *Technical Tests of Object Properties*

We conducted technical tests of various object and filament properties. The general results are discussed below, please see Appendix B.2 for test descriptions and more specific results.

Test B.2.1 Object surface adhesion

During empirical tests of 17 materials (Table B.1), only low density foam blocks proved problematic. Materials should have a hardness similar to medium density foam or harder. Materials that are softer might have insufficient physical resistance to reliably receive deposited printing material since FDM printing uses some downward pressure for layer adhesion. Materials should also remain stable at the temperature of the heated build plate and the filament when extruded. We found even materials with a melting point below 200°C can be used, albeit with some marring and deformation to the object surface. Such materials can be reused multiple times, but not indefinitely. Adding a layer of masking tape to the top surface of an object will prevent most surface marring. In practice, the adhesive properties of the replacement

object material in relation to the printing material is unimportant as long as the top object surface is covered with masking tape, a thin layer of paper glue, or a similar coating.

Test B.2.2 Adhesion with different filaments

Different types of support printing material like PLA, PETG, ABS, and TPU can be used. Depending on the printing material, tape or glue may have to be applied to a different set of replacement object materials.

Test B.2.3 Object measurement precision

Objects should be measured with an accuracy of at least ± 0.25 mm to ensure a secure fit inside printed holders and to ensure ideal print adhesion. Measurements can be error-prone [51], but cuboids should be simple to measure and using standardized objects like toy bricks requires no measuring. The top surfaces of objects should be mostly planar. However, adhesion remains reliable for height irregularities or measurement errors of up to 0.3 mm. Additionally, irregular objects beyond that range, such as a slightly arched softcover book, may still be used if the print head can compress them to be within tolerance.

Test B.2.4 Stability with build plate movement

The sizes, shapes, and densities of replacement objects should be such that the object is stable and unlikely to tip. In general, heavy objects with a low centre of gravity are better suited than lightweight objects. The object holders that our system prints onto the build plate provide significant additional lateral stability, which prevents movement and tipping even for light and top-heavy objects. Objects remain secure even if the build plate moves at maximum speed. Holders can be secured to the build plate further by printing a single layer high brim. Brim width is adjustable as a slicing parameter.

Test B.2.5 Finish quality after long pauses

When printing with PLA, neither pausing nor extended idle periods cause noticeable deformations or surface finish degradation. Changes in geometry may change the printing direction of perimeters, which might be visible at certain reflection angles.

Test B.2.6 Compatibility with support structure types

The geometry or layout of support structures seems to have no effect on adhesion, making our approach likely compatible with any type.

4.3.2 *Interaction Density and Complexity*

There are a variety of ways that one or more replacement objects can be inserted into a print. We consider user effort to insert objects as the *density* and *complexity* of interactions. Our technical implementation attempts to balance this expected user effort with material and time savings.

Interaction Density

There can be one or more object insertions each time the printer pauses. Density refers to how frequently the user is interrupted during a print, and how many insertion steps are required during each interruption. The rows in Figure 4.4 illustrate this dimension.

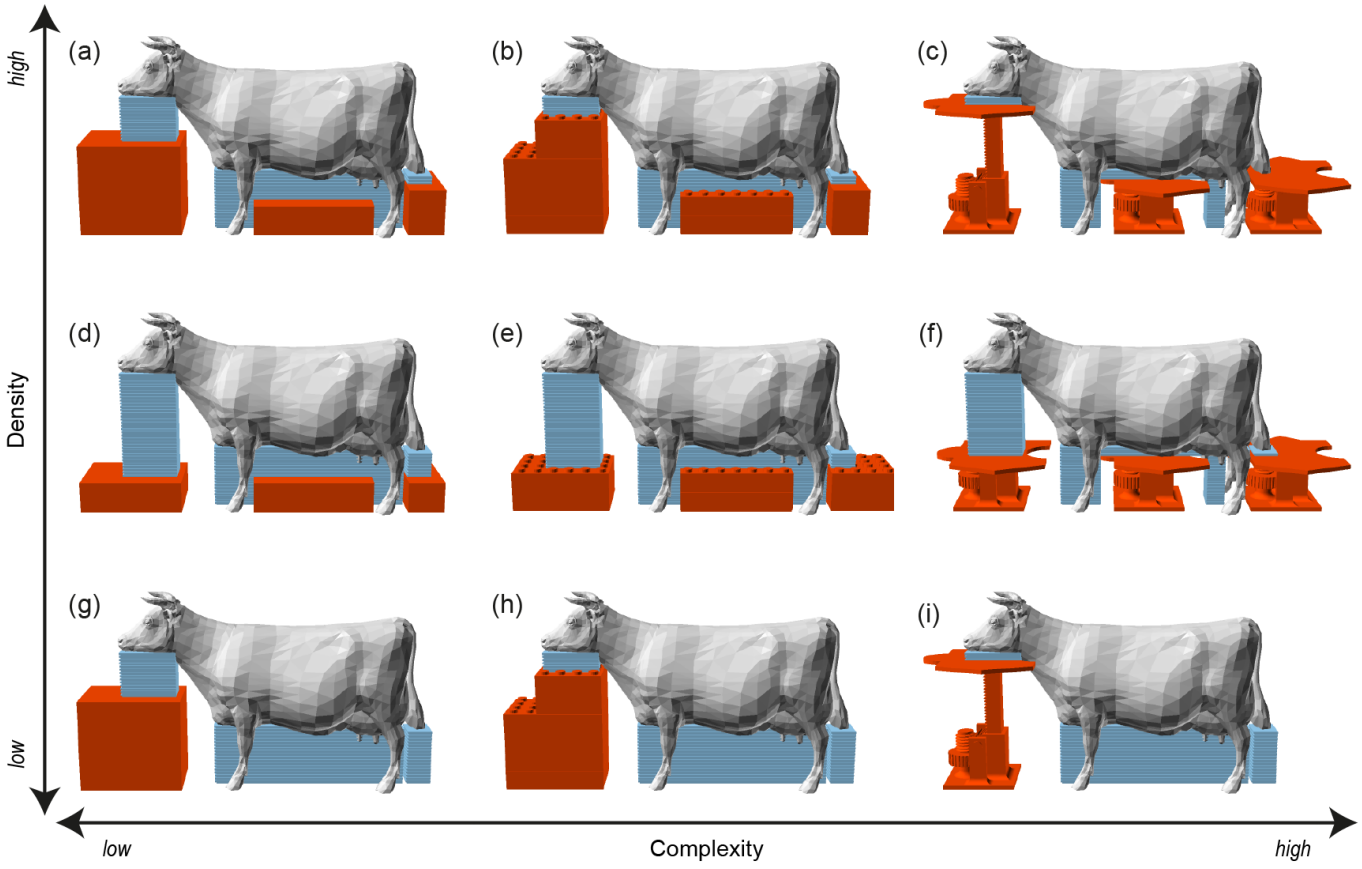


Figure 4.4: Interaction density and complexity: (g) a single cuboid replacement object requires only a single simple user interaction for insertion; (e) multiple more complex replacement objects with the same height also creates only a single interruption, but it requires more time to locate, assemble and insert multiple stacks of objects; (c) high density and complexity result in multiple user interactions throughout the print process and require additional precise adjustment steps at insertion time.

If the print requires only a single replacement object, then there is only one interruption with a single insertion, making the interaction density low (Figure 4.4 bottom row). If multiple objects are used as support replacements, and all objects have the same height, there is still a single interruption but with multiple interactions to insert each object (Figure 4.4 middle row). This increases the density of a single interaction. Common types of uniform ad hoc objects exist, such as wooden off-cuts sharing the same standard thickness. When there are multiple replacement objects with different heights, there will be multiple interruptions creating a high density of interactions during the print (Figure 4.4 top row). There are other timing characteristics that affect interaction density. If multiple interruptions occur near each other, perhaps separated by a few minutes, the perceived interaction density is likely lower than if the same number of interruptions were spaced out during the print. The actual time of the interruption also likely affects perceptions of interaction density: consider object insertions during working hours versus in the middle of the night.

Interaction Complexity

Complexity is an orthogonal dimension to interaction density that captures how difficult, time consuming, or error-prone a single interaction during a single object insertion step is. The columns in Figure 4.4 illustrate the range of this dimension with three examples.

Examples of low complexity interactions are inserting fixed height replacement objects (Figure 4.4 left column). Objects only need to be located and inserted into the printed holder. A medium complexity interaction example is stacking multiple objects to achieve a desired support height (Figure 4.4 middle column). This requires more effort to gather the stacked objects, stack them, and check the height. It is more error-prone since it is possible to misunderstand stacking instructions, or introduce unexpected dimensional height variations due to how well the objects stack. An example of a high complexity interaction is using an adjustable replacement object like a screw jack (Figure 4.4 right column). There is increased complexity to fabricate the object itself, though this is a one-time cost. Setting a precise height requires some additional time and could be error-prone, so our system parks the print head to form a gauge for the user to "dial in" an adjustable height. The top of the screw jack shields the user's hand from a potentially hot nozzle.

In practice, complexity and density could achieve similar results in terms of support savings. For example, Figure 4.4a has three pauses each with a single simple insertion, while Figure 4.4i has a single pause but requires a more complex interaction to set a screw jack height; both achieve similar results. Likewise, Figure 4.4b and f use varying mixtures of complexity and density to achieve comparable results. This means the user has multiple options to choose based on their goals, constraints, and preferences.

4.3.3 *Insertion Feasibility Test*

The fundamental action of inserting an object into a holder is simple, but unfamiliar in the context of 3D printing. To evaluate this crucial step, we recruited 6 people with 3D printer experience ranging from none to expert. We explained how to assemble toy bricks securely and how to adjust a screw jack to match the print nozzle height. Using a separate build plate, we demonstrated how to align an object with the holder, insert it, then wiggle it to test stability.

After this introduction, the participant inserted three objects of varying complexity during a single printing pause: a simple cuboid, a 4x4 stack of toy bricks that required assembly to match the correct height, and a screw jack that had to be height adjusted. They were asked to test stability after insertion, and comment about their confidence in its placement. We recorded the time from when they began the task until they told us they were done inserting all three objects. Then, the 3D printer was restarted, and 10 layers were printed on all inserted objects to objectively test the placement.

For these first time users, the average time to assemble/adjust, insert, and test stability of all objects was 54 s (median 46, min 34, max 98, sd 22). All participants expressed confidence in their final insertion, and all objects were printed on successfully. A participant with little 3D printing experience described the task as "easy." We believe these times would be much faster with experience. For example, two participants did not initially understand how much pressure was needed to snap objects into holders, and one participant chose to re-insert objects multiple times, "just to be sure."

4.4 IMPLEMENTATION DETAILS

The system is implemented into the open source Kiri:Moto slicer [6]. In the modified user interface, the desired level of interaction density and complexity are adjustable, there are optional scheduling constraints expressed as time periods to avoid inserting objects (see above), and a search quality setting adjusts the optimization stopping condition and how similar object placements are pruned to speed up the process. When the user initiates slicing, our system searches for solutions given these user parameters and the available object library.

Our method uses the sliced 2D polygons representing the 3D model and support areas to select and position support replacement objects. To optimize computation, a copy of these 2D polygons are simplified using a heuristic method to conservatively remove excess precision. Then, our system proceeds in three steps: (1) separating the support into area clusters; (2) searching for candidate replacement objects and positioning them within each cluster; and (3) combining compatible solutions from each cluster into a list of global solutions. After determining the best global solution, the slicing process continues as normal, generating printing paths for the modified set of 2D polygons to accommodate the replacement objects. Our system further modifies G-code

to add object holders, insert pauses with brief messages to alert the user to insert objects, and to make layer adjustments to accommodate arbitrary object heights. Working directly with slices and G-code means our technique is compatible with support generation strategies other than standard support fill, such as cones or trees.

4.4.1 *Separate Support into Clusters*

To help the subsequent search step find solutions focused on areas with large volumes of contiguous support, K-Means [59] is used to separate the total support material volume into sets of clusters. For efficient parallel computation, the maximum k is chosen so each cluster can be explored in parallel. As an example, 5 cluster sets for $k = 1$ up to $k = 5$ generate 15 unique clusters (i.e. $1 + 2 + 3 + 4 + 5 = 15$) which can all fit within a 16 core processor.

Clustering is performed on a height map of support. Support polygons from all slices are discretized onto a $1x1$ mm grid, with each cell set to the number of slices that contributed. Considering height is important since adjacent support areas in X/Y should not be in the same cluster if they extend to vastly different heights in Z. After clustering, a concave hull (concavity factor 100) is generated for each cluster. Figure 4.5 shows an example of $k = 5$ clusters.

4.4.2 *Find Solutions for Each Support Cluster*

We use Particle Swarm Optimization (PSO) [50] to explore the solution space since it offers robust control parameters, is sufficiently efficient, and used in related geometry applications [42, 57, 115]. In parallel, PSO searches are performed within each support cluster hull generated by the sets of clusters. Each particle describes a target cuboid volume with: bottom X/Y corner position; length, width and height; and 2D rotations around the X/Y corner and object centre. Including two rotation parameters enables the particle volume to swivel around cylindrical model features while maintaining alignment with other particles. At each optimization step, a candidate replacement object is selected from the library that best approximates the particle's target length, width, and height. If available, this can be an assembly constructed from a set of toy bricks, or a variable height object like a screw-jack.

When the PSO search starts, each particle target cuboid volume is initialized randomly to span the smallest and largest replacement object in the library. The particle positions and orientations are initialized so 90% of the particles have one edge aligned with part of the cluster hull outline (Figure 4.5). An object placed beyond the hull is unlikely to overlap additional support from that cluster, and the hull typically defines a border between the support and the model, suggesting a non-colliding position with large support area coverage. The remaining 10% of particles have entirely random positions and orientations to provide a way for the PSO to escape early local maxima. Then,

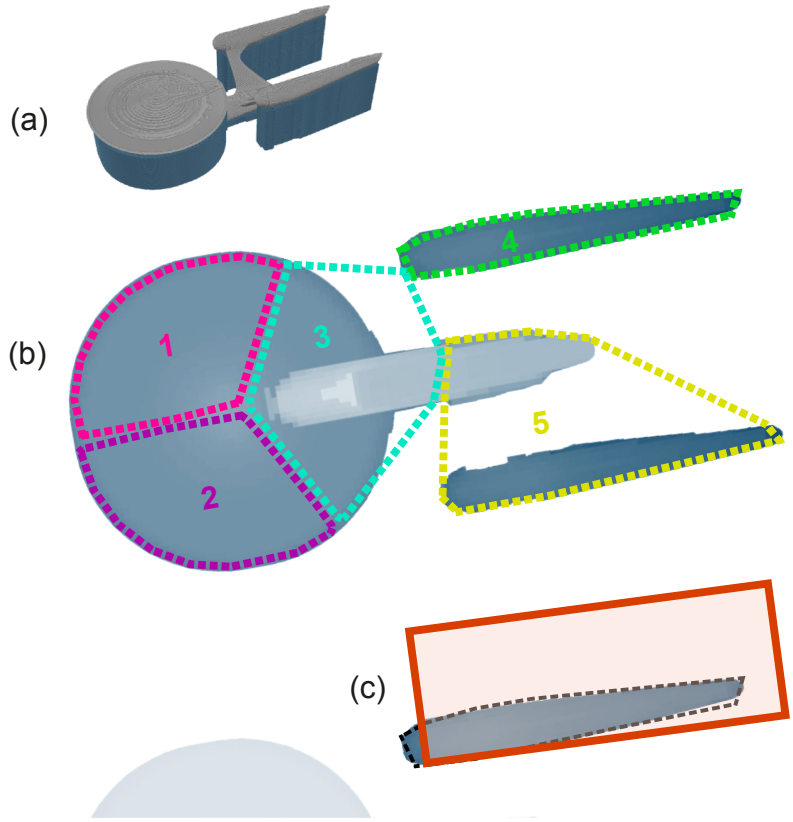


Figure 4.5: Example support material clustering and search: (a) sliced model with support shown in blue; (b) support material height map (darker means more support) with $k = 5$ clustering and convex hulls; (c) each cluster is searched independently, here cluster 4 is shown with a random initial object position (orange rectangle) aligned to part of the hull outline. In practice, many different replacement objects and positions are found for each cluster.

the PSO search iterates in a loop: adjusting particle positions, orientations, and sizes; selecting matching objects for each particle; then calculating their fitness (see "Fitness Function" below). As is typical for PSO, the particles initially move at high speeds to explore a wide range of possibilities, then gradually reduce speeds so each settles into a local fitness maximum. The search stops when it no longer yields a significant improvement in fitness. This is defined as multiple subsequent steps where an accumulated and decayed fitness improvement is below a relative threshold. The search quality setting in the user interface indirectly adjusts the decay and relative threshold to make the stopping condition more or less lenient. On completion, each support cluster PSO returns a list of candidate objects that do not overlap the model.

4.4.2.1 *Fitness Function*

Each particle fitness is a product:

$$f_i = S \times o \times c \times b \tag{4.1}$$

where:

S is the *volume of support replaced* by the candidate object (mm^3).

o is an *overlap penalty* defined using $\omega \in [0, 1]$, the proportion of the object that *does not* intersect with the model:

$$o = \begin{cases} (1-\omega)/10, & if \omega < 1 \text{ (there is overlap)} \\ 1, & \text{otherwise (no overlap)} \end{cases}$$

Note that any overlap invalidates the candidate from being used in a final solution, but scaling o by a factor of $1/10$ when there is any overlap creates a soft constraint, resulting in a smoother fitness space for the PSO to explore.

c is an *object complexity penalty* defined using an object type complexity factor α and the level of interaction complexity selected by the user $I_c \in [0, 1]$:

$$c = 1/\alpha^{(1-I_c)}$$

Fixed size, simple objects have $\alpha = 1$; more complex objects that must be assembled prior to insertion, such as stacking floppy discs or assembling toy bricks, have $\alpha = 2$; the screw jack, which requires manual height adjustment has $\alpha = 3$. The interaction complexity factor is used to condition the object complexity (e.g. $I_c = 0$ means keep complexity low, $I_c = 1$ means high complexity is acceptable).

$b \in (1.0, 1.05]$ is a *small footprint bonus*. Objects with smaller contact area on the baseplate are less likely to interfere with other replacement objects when combining local solutions into global solutions. b is mapped to the sizes of the smallest and largest replacement objects, such that the smallest object receives full 5% bonus and the largest receives almost none.

4.4.2.2 *Constructing Ad Hoc Towers*

If ad hoc towers are enabled in the library, the system also attempts to recursively place objects on top of other candidate objects, using those returned by the main PSO as a base. This tower creation uses a PSO with fewer particles, earlier stopping criteria, and a search area restricted to the base particle area. The fitness function is the same, except that b is reversed to assign a bonus to larger objects: this increases the potential for additional objects in the tower by providing more surface area for the next object.

4.4.2.3 *Run-time Optimizations*

Strategically chosen subsets of layers are used to approximate support material savings S and for early detection of model collisions. Initially, the calculation uses the lowest and highest layer according to the particle height, along with 5% of the layers between. If the approximation of S is above a threshold (based on maximum material savings of previous candidates) and no collision is detected, the remaining layers are added to the calculation in random order, each time checking for the same stopping conditions. This method significantly decreases objective function computation since particles with collisions or poor support savings are not fully calculated.

Replacement objects with variable heights, such as toy construction bricks and screw-jacks, are explored in an optimal way. They are first evaluated at their minimum height (e.g. a fully retracted screw jack) to confirm base placement validity. Then, the adjustable height is gradually increased within the possible range until the most support material is replaced without causing a collision. This way, all possible height variations for a candidate object are evaluated at once to find an optimal configuration.

4.4.2.4 *Final List of Candidate Objects*

When the main PSO search and ad hoc tower searches have completed, all candidate object placements that do not collide with the model and replace more than a minimum amount of support material are saved. The minimum threshold ϵ (in mm^3) is based on the search quality setting $q_\epsilon \in [0.2, 0.3]$ and the replaced volume of the current best candidate S_b, where $\epsilon = q_\epsilon \times S_b$. This list can be very large, so it is further pruned heuristically to select a diverse set of options with a focus on the most beneficial ones. Duplicates of equivalent results are removed. Results are sorted by support material saved, then every ith result is selected where $i = \lfloor 1.5(i-1) \rceil$. This indexing method selects the very best results, some good results, and a few poor ones. This selection process is performed separately for lists of individual objects and lists for each tower size (i.e. 2 object towers, 3 object towers, etc.). This selection process ensures a relatively stable number of outputs, even though the number of valid results from the search can vary wildly. A diverse set of candidates is useful when creating global solutions across objects selected for each support cluster.

4.4.3 *Combine into Global Solutions*

The system now combines different objects across support clusters into global solutions of replacement objects. Each global solution can use a specific physical object at most once, and all objects in a solution must not collide. Global solutions are constructed using each single cluster set (e.g. considering only the solutions from 4 clusters in the $k = 4$ set) and across all cluster sets. Sets of global solutions are generated in parallel on separate threads.

Finding the best combination of candidate objects to form a global solution is equivalent to determining the solution to a *maximum weighted clique problem* on a graph describing object compatibility. A graph is constructed where each vertex represents a candidate object (or tower) at the solution position, with the vertex weight set to the objective function value for the associated particle (f_i in Equation 4.1). Vertices are connected by edges if the objects they represent are not the same replacement object and they do not collide. A custom solver is used to find the maximum weighted clique for clique sizes ranging from 1 to 8 (we consider a single vertex a clique of size 1). Larger cliques translate to solutions with more objects which require more user interactions, so finding global solutions with smaller cliques (representing fewer objects) is important to provide options to the user.

The conventional weight of a clique is the sum of included vertices, but we set the clique weight using a global objective function that also considers the user's desired level of interaction density:

$$f_g = \frac{\sum_{x=i}^{N} f_i}{0.75P^{(1-I_d)} + 0.25N^{(1-I_d)}}$$

where:

f_i is the *fitness of a candidate replacement object* (Equation 4.1);

N is the *number of objects to be placed*;

P is the *number of printing pauses* (i.e., number of "interruptions");

$I_d \in [0, 1]$ is the user's *interaction density setting*.

Note the number of printing pauses is weighted 3 times greater than the number of objects to be placed (i.e. $0.75P$ vs. $0.25N$). This codifies an assumption that more pauses are more disruptive than placing more objects in a single pause. However, these weights can easily be adjusted, in the future they could be included in the user interface.

4.4.3.1 *Run-time Optimizations*

We prune some vertices using heuristics to decrease computation for the maximum weight clique solver. Since maximum cliques are evaluated in order of increasing clique size, vertices with fewer edges than the target clique size minus one are pruned to speed up following iterations. For example, when searching for a clique of size 4, a vertex with only 2 neighbours can be pruned. If multiple vertices are not connected, but each has identical neighbours, then they likely replace a similar area of support and are interchangeable within a solution. Among sets of interchangeable vertices, only the highest weight vertices representing a single object and a tower are kept in the graph. If more than 150,000 combinations of vertices remain after the previous two steps, additional vertices are pruned randomly to avoid excessive computation times.

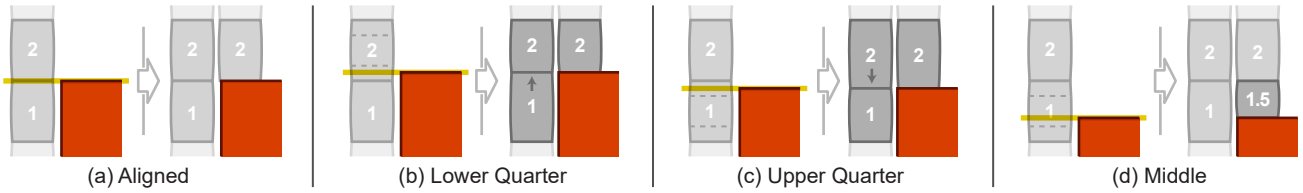


Figure 4.6: Side views of layer height adjustments cases showing two adjacent boundary layers (grey shapes) before and after adjustment to object (orange shape): (a) object height and layer boundary are aligned, no adjustment necessary; (b) object top is within lower quarter of layer 2, layer boundary adjusted upwards; (c) object height is within upper quarter of layer 1, layer boundary is adjusted downwards; (d) object top is within middle 50% of layer, new intermediate layer inserted between boundary layers.

4.4.4 *G-code Modifications*

Once the most desirable global solution is found, all support material print paths that are replaced by objects are removed from the slicing result. Our system then modifies the heights of layer boundaries near the top surfaces of objects, adds printed object holders, printed text instructions on the baseplate, and inserts printing pauses.

The layer boundary is the height at which filament from a new layer touches the previous layer, and it rarely matches the height of a replacement object precisely. Without adjustment, these differences can cause visual imperfections and structural weaknesses. Our system adjusts adjacent layers to match the height of inserted objects by considering the four cases illustrated in Figure 4.6. When the top surfaces of two or more inserted objects fall within the same layer boundary, each is handled separately.

Custom holders and instructions are created by inserting parameterized G-code generated by our system. Each object holder is composed of 6 layers of filament extruded in a single outline path offset from the object by half the nozzle thickness (e.g. 0.2 mm for a 0.4mm nozzle). A brim for increased adhesion can be added using a user setting. Brief instructions are printed on the baseplate near the object as a single-layer vector line font with a user-defined letter height in mm. Instructions provide a human-readable replacement object id (also searchable in the library) with dimensions for simple objects or manual adjustment for variable objects, such as screw jack height or toy brick assembly size. The G-code that pauses the print for the user to insert an object can be customized, such as allowing time for a heated enclosure to reach a desired temperature. The printing material and time required to print the holders and instructions are included in the measurements throughout the paper.

4.5 EVALUATION

To evaluate our approach, we conducted a technical evaluation using the system with a dataset of diverse models. The primary measures are proportions of printing time and support material saved. We also explore variations relat-

ing to the interaction density and complexity "usage space" factors discussed earlier. These are operationalized as slicer parameters and what types of ad hoc objects are available in the library.

4.5.1 *Test Dataset of 3D Models*

We assembled 15 3D models to test various conditions. Nine are models identified in previous research related to 3D printed support structures and related topics. Some were reconstructed from descriptions within the related work. The remaining six are canonical 3D models used in computer graphics research. Note that not all previous works identify the models used, and few provide mesh files or references. We also limited our selection to models that can be accessed freely for replication. Some models were scaled to fit into the build volume of the printer preset used in the evaluation. Models were printed in their original orientation, which in some cases is non-optimal to showcase extreme support cases. Our goal was to gather an assortment of objects with reasonable diversity, replicability, and comparability, and using objects from previous work mitigated selection biases. The complete dataset is shown in Figure B.1 of Appendix B.

4.5.2 *Ad Hoc Objects*

The ad hoc objects used for the evaluation were primarily collected from the authors' households. This included 22 objects with fixed heights, such as a book, woodworking off-cuts, 3D printing calibration cubes, and small cardboard packaging boxes of various sizes. Object dimensions ranged from $73 \times 10 \times 9.5$ mm to to $172 \times 144 \times 37$ mm. Nine floppy discs were used as one type of stackable object. A toy brick construction kit was included as an option in the library, which essentially enables an arbitrary variety of ad hoc objects by assembling blocks into a required size as needed. Finally, there were 3 adjustable objects with continuously variable height in the form of 3D-printed screw-jacks with different height ranges (29-46 mm, 49-66 mm, and 78-95 mm). All three screw jacks used the same prism shaped cap. In our experiments, we use these objects with different levels of complexity to form different libraries as a dependent variable to evaluate the impact of interaction complexity.

4.5.3 *Procedure*

We first generated machine printable G-code for all dataset models with the unmodified Kiri:Moto slicer (version 3.3.D12) [6]. We used default slicing settings, with the exception of enabling automatic support and using a support pillar size of 8 instead of 6. Increasing the support pillar size ensures that continuous areas of support are merged into a single structure, rather than forming numerous small disjoint towers. This reduces execution time by reducing the number of support polygons. These default slicing results

are baselines for material weight and printing time. All models were also sliced without any support to obtain the model material weight to compute proportional savings in material and printing time.

We then generated machine printable G-code for all models in the dataset using our system. In these tests we use the default print volume (300 × 175 × 150 mm) with all models centred on the build plate. Since replacement objects must be placed fully within the build plate area, the position of model affects the possible solution space. All tests were run on a high end consumer Windows PC with 24 available processing threads using the Chrome browser.

4.5.4 *Independent Variables*

There are three independent variables: two settings offered by our customized slicer (search quality and interaction level) and which library of replacement objects is available.

- *Search Quality – Fast or Thorough.* This setting adjusts details of the PSO search, such as the number of search particles; how persistent the search should be, as well as thresholds for early exit conditions during the search.
- *Interaction Level – Low, Medium, or High.* This evaluates increasing levels of user interaction as a combination of interaction density and complexity, where I_d and I_c are 0 for *Low*, 0.5 for *Medium*, and 1.0 for *High*.
- *Library – Simple, Towers, or Variable.* The tests are run using three libraries containing progressively more versatile and adjustable ad hoc objects and arrangements. The *Simple* condition is a library with 22 fixed height objects without stacking; the *Towers* condition adds the easily stackable floppy discs and permits object stacking to form ad hoc towers; and the *Variable* condition adds 3 height-variable screw-jacks with different height ranges, and enables assembling objects of varying sizes using a toy brick construction kit.

4.5.5 *Dependent Variables*

For each combination of independent variables, we conducted a search with each test model and calculated three metrics:

- *Material Saved.* This is the proportion of printed support material weight generated by our system to the support material weight generated with the unmodified slicer. We calculate support material weight based on the printing weight estimates provided by the slicer itself, which is defined as the total length of extruded filament times the area of a cross section of the filament, times the average density of PLA.
- *Time Saved.* We use printing time estimates provided by the slicer itself for the total time to print the model including all support material. We define time saved as a proportional savings of print time using the ratio of time

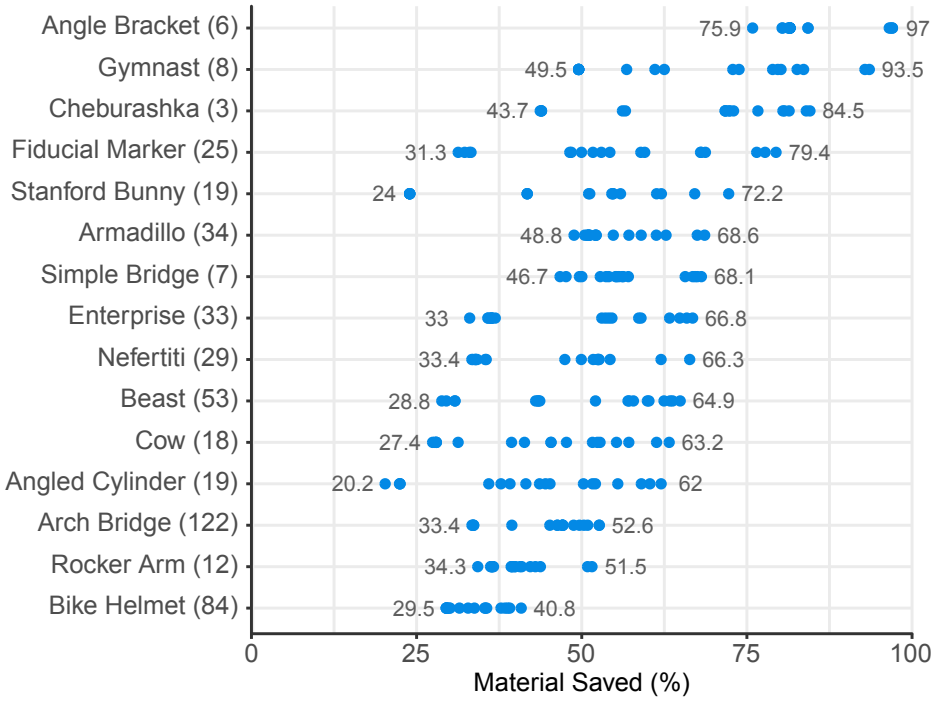


Figure 4.7: *Material Saved* by model for 12 test runs spanning all conditions. The default support material mass for each model is shown in parentheses (in grams).

to print G-code generated with replacements to the time to print G-code generated with the unmodified slicer.

- *Search Time.* The run time of our system was measured during the ad hoc object search process and during the entire slicing process to generate all G-code.

4.5.6 *Results*

For each of the 15 models, we ran the system for 12 test runs spanning all conditions: 2 *Search Qualities* (fast, thorough) $\times$ 3 *Interaction Levels* (low, medium, high) $\times$ 3 *Libraries* (simple, towers, variable). This produced 270 data points for each dependent variable.

Each dependent variable is analyzed using a $2 \times 3 \times 3$ repeated-measures ANOVA with test model as a random blocking variable. Visual inspection of Q-Q plots revealed no severe violations of normality; Greenhouse-Geisser corrections are applied to p-values when Mauchly's test for sphericity was significant; significant differences are reported at a 5% level. All post hoc tests use Bonferroni-corrected pairwise t-tests.

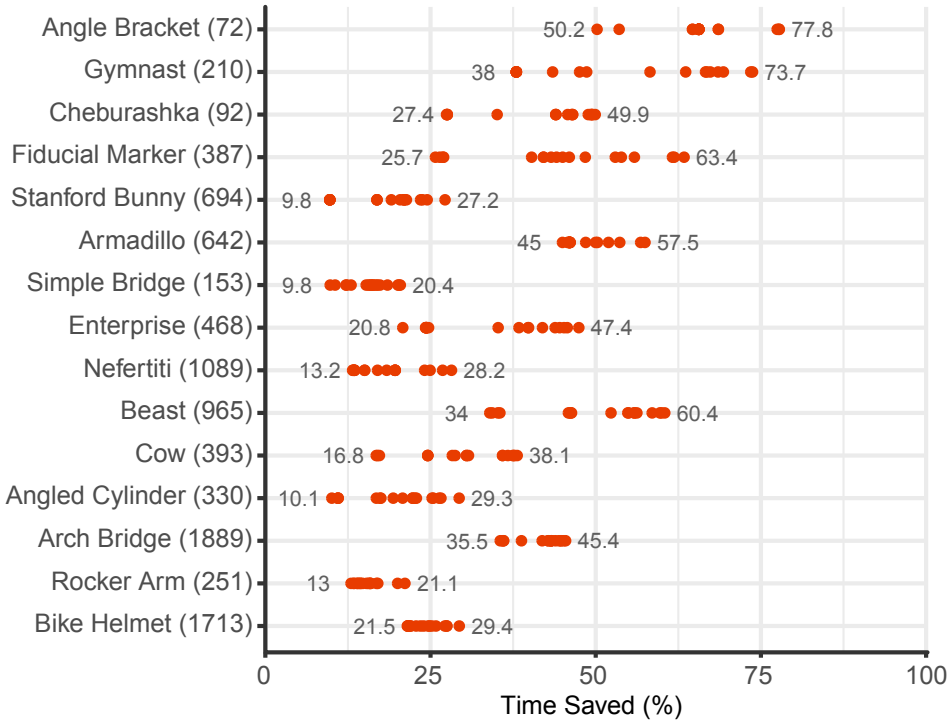


Figure 4.8: *Time Saved* by model for 12 test runs spanning all conditions. The proportion of print time saved by ad hoc objects varied depending on model and how much default printing time is spent creating support. The default print time for each model is shown in parentheses (in minutes).

4.5.6.1 *Support Material Saved*

Overall, replacement objects reduced support material by 52.0% (95CI [50.0, 54.1]). *Material Saved* ranged from 20.2% (worst result for the Angled Cylinder model) to 97.0% (best result for the Angle Bracket) (see Figure 4.7).

Increasing levels of user interaction saved more material, libraries providing more versatile and adjustable ad hoc objects saved more material, and using a more thorough PSO search saved more material (see Figure 4.9). There was a main effect for *Interaction Level* ($F_{2,28} = 29.1, p < .001$), *Library* ($F_{2,28} = 42.2, p < .001$), and *Search Quality* ($F_{1,14} = 65.2, p < .001$). Post hoc tests found significant increases in material savings for *Low* (44.0%), *Medium* (53.8%), and *High* (58.5%) *Interaction Level* (all $p < .05$) and significant increases for the *Simple* (45.1%), *Towers* (50.6%), and *Variable* (60.6%) *Libraries* (all $p < .05$). *Search Quality* had less of an effect, with 50.7% support saved for *Fast* versus 53.5% saved for *Thorough*.

With the *Towers* library, moving from a *Low* to *Medium Interaction Level* made the most pronounced increase (14.6%), unsurprising as only *Medium* and *High Interaction* permit the additional pauses required to construct towers, with little or no additional savings moving to the highest level of interaction. For the other libraries, there was a steady increase in average material savings ranging from 4.7% to 7.9% when moving to the next higher *Interaction Level*. There

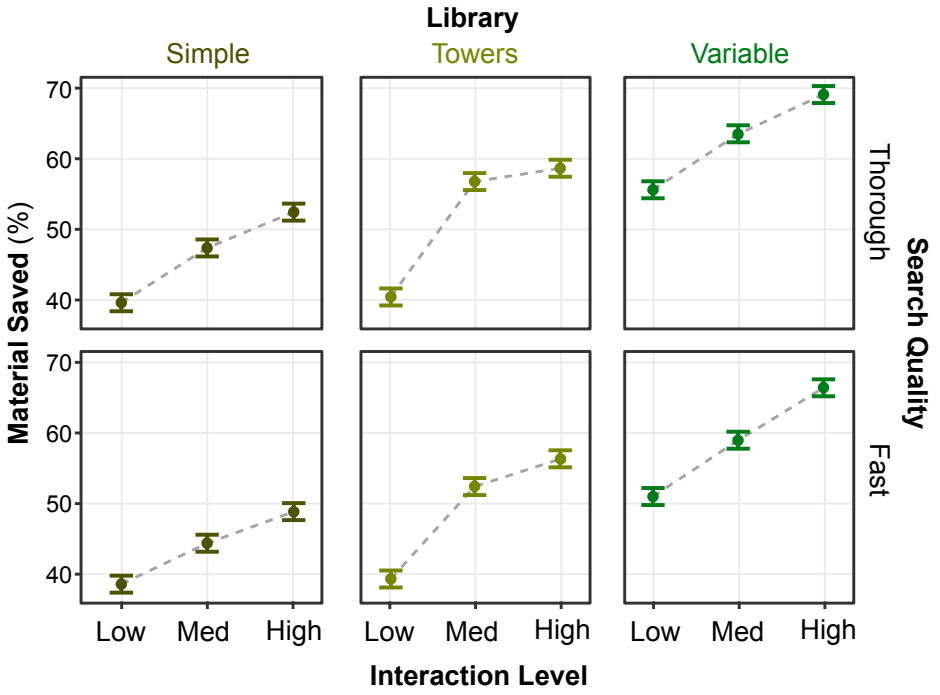


Figure 4.9: Percentage of *Material Saved* for each combination of *Interaction Level*, *Library*, and *Search Speed*. Error bars are 95% confidence intervals.

was a *Interaction Level* × *Library* interaction ($F_{4,56} = 3.17, p = 0.048$). Post hoc tests examining *Interaction Levels* for each *Library* found significant differences in all cases (all $p < .05$) except between *Medium* and *High Interaction Level* with the *Towers Library*.

4.5.6.2 *Time Saved*

Overall, replacing support with objects reduced print time by 34.8% (95CI [32.8, 36.9]). *Time Saved* ranged from 9.8% (worst performance for the Stanford Bunny) to 77.8% (best result for the Angle Bracket) (see Figure 4.8).

Additional print time was saved with increasing levels of user interaction, with libraries providing more versatile and adjustable ad hoc objects, and with a thorough PSO search. There were main effects for *Search Quality* ($F_{1,14} = 70.0, p < .001$), *Interaction Level* ($F_{2,28} = 17.3, p < .001$), and *Library* ($F_{2,28} = 22.5, p < .001$). Post-hoc tests found *Time Saved* increased with each *Interaction Level*: *Low* (29.7%), *Medium* (36.1%), and *High* (38.6%). The same trend was found with *Library*: *Simple* (30.8%), *Towers* (34.4%), and *Variable* (39.2%). Though significant, *Search Quality* had less of an effect, with 34.4% time saved for *Fast* versus 35.2% time saved for *Thorough*.

	Simple	*Towers*	*Variable*
Fast	9.2 (8.6)	14.5 (15.0)	12.7 (10.9)
Thorough	20.6 (20.4)	39.9 (36.8)	28.8 (25.0)

Table 4.1: *Search Time* by *Library* and *Search Quality* (mean time in seconds, stdev in parenthesis).

4.5.6.3 *Search time*

On average, *Search Time* took 20.9s (sd 23.5 s, (95CI [18.1, 23.7])). *Search Time* ranged from 0.9 s (Angle Bracket, fast, simple, low interaction) to 124 s (Bike Helmet, thorough, towers, low interaction), 18 searches took longer than a minute. For comparison, the average slicing time without using our system is 16.7 s. Table 4.1 summarizes *Search Time* results.

Searching the *Simple* library was consistently fastest and, as expected, a *Thorough* search takes more time. There were main effects for *Library* ($F_{2,28} = 6.32, p = .016$) and *Search Quality* ($F_{1,14} = 19.5, p < .001$). Post hoc tests found *Simple* (14.9s) faster than *Towers* (27.2s) and *Variable* (20.8 s) (all $p < .05$). *Search Quality* was unsurprisingly faster on average with *Fast* (12.1s) than with *Thorough* (29.7s).

Interestingly, the most versatile *Variable* library actually reduced search time compared to *Towers* with ad hoc object stacking. This makes sense, as objects with variable height may already reach up to the desired height, avoiding the need to construct towers recursively in many cases. Within each library, *Fast* quality search times are approximately half those of *Thorough* quality search times. There was a two-way interaction effect for Search Quality and Library ($F_{2,28} = 10.5, p < .001$). Post-hoc tests found that for *Fast* search, *Tower* (14.5s) and *Variable* (12.7s) were significantly longer than *Simple* (9.2s) ($p < .05$) but not different from each other. However, for *Thorough* search, all three values were different: *Towers* (39.9s) was the longest, followed by *Variable* (28.8s), then *Simple* (20.6s) (all $p < .05$). For all libraries, *Thorough* search took longer than the equivalent *Fast* search.

4.6 DISCUSSION

Replacing support material by involving the user in the fabrication process rather than using automated motion controlled mechanisms provides additional flexibility. Unlike previous solutions, this approach is not limited to a rectangular grid. This enables custom object shapes and arbitrary positions for support replacement, which can reach and replace additional support material. The flexibility comes at a cost of increased computational complexity. By applying domain knowledge and placement heuristics to streamline the optimization, and careful pruning to shrink the solution space to a manageable size, our system overcomes the complexity to provide the sustainability benefits of reusing support to anyone who owns a FDM 3D printer.

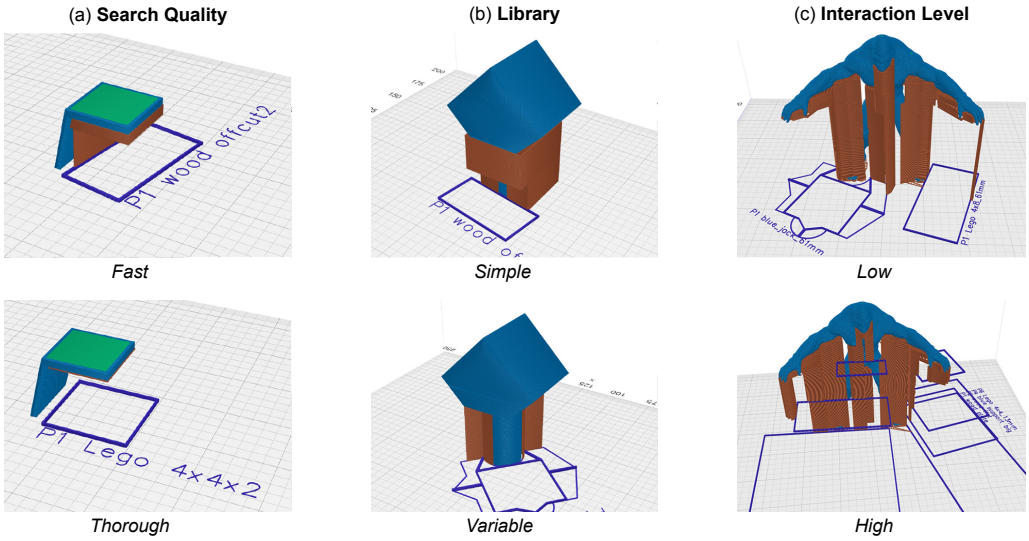


Figure 4.10: Examples of solution variations: (a) a fast search solution for Angle Bracket places a single wood replacement with sub-optimal height; a thorough search solution finds a more optimized object constructed with toy bricks; (b) using a library of simple objects, a single block of wood is used to save some support; using the variable object library, non-rectangular objects are available to better fit the curved surface of the Angled Cylinder; (c) a solution restricted to low interaction can still replace a large volume of support with multiple objects placed during a single pause; a solution that permits a high level of interactions can replace even more support, but requiring 7 print pauses for this example.

Our system must consider the importance of a user's time. With decreasing size of print models, the benefits of using the system can become small enough that they might not justify any additional user interaction. The system avoids interactions that only save a token amount of support and exports the unaltered slicing result if no worthwhile user interactions can be made. Though not being able to save support for any model is an undesirable outcome, the affected small prints produce less waste from support material in the first place. In contrast, the system is most effective for large models that would generate excessive amounts of waste. Overall, the system provides the most benefit per user interaction for models that require large amounts of continuous support.

4.6.1 *Ad Hoc Objects for Sustainability*

From our 15 diverse models, using ad hoc objects reduced support material by 52.0% and print time by 34.8% on average. The best cases for all models allowed replacing more than two thirds of support material on average. Print time has a lower effect generally, since the print time depends on both support and model material. Other techniques, such as in-fill optimization [120] or objects being inserted inside the model could further reduce print time, as shown in Chapter 3.

The efficacy of using ad hoc objects as support replacements depends on the specific arrangement of support material. In some cases, the effects can be dramatic. For models with large, planar overhangs like the Angle Bracket, using replacement objects can remove almost all support material (97.0% savings), and reduce printing time by more than half (77.8%). However, in other cases, results are limited. In some models much of the support material is build on top of model geometry and unreachable from the build plate. Given the proposed "usage space", extensions can make it possible for ad hoc objects to also replace support in these situations, but likely with some added interaction complexity. Figure 4.10 illustrates more examples of solution variations.

4.6.2 *Comparison with Previous Approaches*

Comparing our average savings to previous work is problematic, as each work uses a very different set of test models. Xu et al. [121] test 5 models and Yiğit et al. [123] test only 2. We use a larger dataset of 15 models and many are more complex in terms of support requirements. This is an argument to standardize test model datasets.

Perhaps most instructive is to compare using a common model. From the models tested by Xu et al., we were able to source the "gymnast". With this, our system achieves maximum support material savings of 93.5% compared to their 78.1%. From models tested by Yiğit et al., we were able to source the "bunny". Here, our system achieves 72.2% savings compared to their 85%. However, even these more direct model comparisons are limited by system evaluation conditions, with different slicers and different slicing settings. For example, more than 19% of support is located above the body and below the ears of the bunny in the slicing result used in our evaluation. Those upper support regions are unreachable for both our system and the Yiğit et al. system, so it is unclear how more than 81% of material could be saved.

In terms of run time, Xu et al. report an average optimization time of 27.4 s, which is longer than our average run time of 20.9 s, despite our more complex set of test objects. For simple test models, our system can deliver results orders of magnitude faster: our lowest search time is less than 1 second, compared to their 25.6 s.

4.6.3 *Alternative types of support structures*

We used the default kiri:moto supports in our evaluation, which generates support towers that are shaped to match the outline of model parts that require support. This resembles "Snug" supports in the Prusa slicer [83]. Other types of support structures, such as Prusa Organic or Cura [108] Tree supports can further reduce the amount of support material. The slicer estimates for printing time and material used by kiri:moto, Prusa, and Cura and each of their respective support types can be found in Appendix B. The data suggests that there is no single optimal type of supports. One model

failed to slice with Cura Tree supports. Printing Prusa Organic supports requires additional printing time, and for some of the models, they also required more material than Prusa Snug supports.

Comparing "between-slicers" is less controlled than the "within-slicer" comparisons above and in our submission, as slicers may use different approaches to calculate the material and time cost estimates. The slicing results alone also provide little insight whether or how much the supports impact printing reliability and visual print quality. By correcting for relative differences in support weight using a per-model ratio of material used by Prusa Snug to kiri:moto supports, we can estimate material savings of our system over Prusa Organic supports. The estimated maximum material savings of our system are 63% on average, ranging from 37% to 95%. Using the worst case material savings of our system, the average improvement drops to 26%. Interestingly, Prusa Organic support is estimated to outperform the worst case of our system for the Nefertiti and Angled Cylinder models. Our method is in principle compatible with any software support generation method. Based on the Prusa and Cura slicing data, we estimate an additional 3-8% of all support beyond our presented results can be saved by combining advanced support generation methods with support replacements.

4.6.4 *Cost-Benefit of Mid-Print Interactions*

Correlated with support material saved, using replacement objects can have dramatic results on printing time. For example, reducing print time from 31.5 h to 14.3 h with the Arch Bridge model (45.4% savings) or from 16.1 h to 9.7 h with the Beast model (60.4% savings). The lowest time saved in the test set was 15 minutes. User insertion time is not included in these values, but since insertions take less than a minute to complete, time savings remain relevant even for small models. Our system prints the names of all replacement objects on the first layer, so the user can search for all objects during printing and place them nearby ready to be inserted. The dimensions of stacks of objects are printed on the build plate as well, so they can also be assembled ahead of time. A specific assembly order is not required. For example, a stack of toy bricks can have multiple holes and cavities without issue, as long as the stack meets the overall dimensions and has a solid top surface. In addition, the effort to manage a library of ad hoc objects is more of an up-front setup cost. We use the library management interface from Scrappy (see Chapter 3) which has a relatively easy way to enter and locate objects, so in practice this is not a time-consuming aspect.

The main cost is not the time to complete the interaction, but when the interaction must be performed. With our approach, someone has to manually insert objects at one or more intervals during a long print job. We note that similar kinds of interruptions exist for manual tasks with appliances: for example when loading and unloading clothes into household washing and drying machines. The scheduler in our system can ensure that interruptions occur at more convenient times, such as during daylight hours. One can

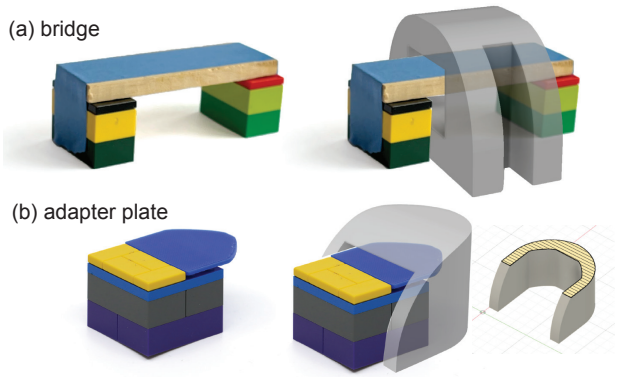


Figure 4.11: Mock-ups of more complex replacement objects: (a) combinations of objects to reach internal support structures like through-holes; (b) a custom adapter printed to perfectly fit into irregular object profiles.

imagine design solutions for further reducing the disruptive nature of the notification: setting up timers or calendar alerts to facilitate scheduling, or even recruiting passersby to insert the objects in a public area or shared workplace. Finally, the adaptability of our fitness function allows minimizing this cost, even restricting solutions to a single mid-print interaction if desired, while still providing worthwhile material and time savings.

Future work could more closely examine the associated cost-benefit trade off. We imagine two complementary approaches: releasing the software to log actual users as they choose different schedules to trade-off interaction complexity; and a longitudinal study using a generic interaction task that can be controlled for density and complexity.

4.6.5 *More Complex Replacement Objects*

Further increasing the complexity of interactions could replace support material at places that are otherwise unreachable. For example, an ad hoc bridge could be constructed from three objects, such as a thin piece of wood taped onto two construction brick stacks (Figure 4.11a). This could enable supporting a model with large openings. To save even more support, a thin plate could be custom printed to perfectly fit into irregular object profiles (Figure 4.11b). Depending on the size and height of the supported area, the overhead to print this kind of custom object before the main print could still save overall time and material, especially when manufacturing multiple copies of the same model.

4.6.6 *Improving Mechanized Approaches*

Our more advanced optimization method system is versatile enough to find object placement solutions to work in tandem with automated approaches from prior work. For example, Yiğit et al. [123] propose using standardized

cubes placed over a 1 cm grid resolution by a robot arm. This major simplification would reduce the search time for solutions, but our results show that ad hoc objects that are not restricted to a grid or a standardized shape are more flexible in placement and can save more material. Using a robot arm (or even the printer itself [47]) to place heterogeneous objects would be challenging, and robot actuators likely lack the dexterity to use our printed holders for object stability. However, our approach would enable more variation in block size and height, and eliminate the limitation of a 1 cm grid resolution. Our optimization algorithm could also be modified to find the optimal positioning of a model on an actuated build plate [37, 121] by treating each grid cell as an adjustable height object at a constrained location. Our approach of relying on users to insert replacement objects could enable a lower-cost alternative to an actuated build platform. Instead, users could place a collection of standardized raised platforms on top of the build plate. Systems that rotate the build platform [31] to avoid printing support may still require some support structures to print complex models. As our printed holders can hold objects even when the build plate is flipped upside down, our approach could work at steep angles in conjunction with such systems.

4.6.7 *Practical Considerations*

Although pausing a print is common in certain applications, such as printing ranges of layers in different colours, or may be required to continue a print when the filament spool runs out, some users may be unfamiliar with the process. To make best use of printing pauses, they should be dialed in like any other slicing parameter. For example, our system includes machine commands to prime the nozzle after a pause, but users may wish to adjust the details of this process based on the type of extruder used. The primed material may be loosely attached to the print. Although easily removed and typically attached to a support tower, custom adjustments to the pausing instructions could prevent such minor blemishes. Our system inserts pauses prior to printing support material. Should a pause cause blemishes, they will typically be constrained to the support material itself. If the build plate is movable, inserting objects may push the build plate out of position. After a pause, the build plate axis is homed to reset its location and avoid issues from accidental movements. Using printing materials that require a heated build chamber might necessitate custom pause handling for prolonged pauses. Maintaining a minimum temperature during the pause or adding delays to allow the temperature within the enclosure to equalize slowly before resuming the print could prevent deformations. The system includes a basic scheduler to help avoid long idle periods.

Using support replacement objects offers additional modes of printing failure recovery. If a user notices a lack of adhesion on the object after a pause, they may still be able to adjust the height of a screw-jack or add layers of blue tape to match the current printing height to restore adhesion after skipping some layers.

Printed supports sometimes stick thoroughly to the supported part of the model. Replacement objects can in some cases essentially allow supporting the area with blue tape rather than filament. The tape is easier to remove sequentially which may prevent damaging fragile models. The smooth finish of the supported area may also be desirable. Support material can typically be removed cleanly from the replacement objects by hand or with a knife.

4.6.8 *Limitations*

Depending on the build plate and filament colour, the printed instructions can be difficult to read under certain lighting conditions. Users may require hand tools such as calipers to measure replacement objects with sufficient accuracy.

The system slightly adjusts the thickness of some layers, which are usually imperceptible, but may still be undesirable when printing mechanical objects. Instead of adjusting layer thicknesses, it would be possible to print a layer of material underneath inserted objects to raise them to the desired height. This solution however would come at the cost of using some additional support material.

Our findings, especially trends within our dataset, may not generalize to all models. Prior work often used very small sets of test models and models chosen or built to demonstrate benefits. We intentionally chose a diverse but representative set of 15 models, and we were able to observe a wide range of outcomes. The current system would be unable to save any support material for pathological examples like a hollow egg model, which only includes internal support. In the worst case scenario where the system fails to find an opportunity to replace support material, the unaltered slicing result is exported instead.

4.7 CONCLUSION

We introduced a practical software-only solution where people manually insert ad hoc objects to replace support material during a FDM 3D print. A technical evaluation shows support material is reduced by an average of 52.0% across 15 models (ranging from 20% to 97%) and revealed how model characteristics influence support savings. Possible future improvements include using a print head depth probe to work with non-planar replacement objects and to measure replacement objects with the printer itself, perhaps circumventing the need for an object library. Regarding user interaction, a single basic mid-print interaction saved 39.1% of support material on average, but with increased density and complexity of interactions, average savings increased to 60.1%. By involving users even a little bit during the print job, 3D printing can be faster and more sustainable.

4.7.1 *Summary*

The Substiports system demonstrates that the benefits of intermittent interaction can scale with user involvement, and offers options for users to find a trade-off between their involvement and the benefits that they prefer. However, a handful of interactions typically suffice to achieve the maximum possible benefits as they stem from preventing a specific source of waste, and additional involvement will not provide further benefits beyond a handful of interactions. As a result, the next chapter explores a different use case that does require high levels of user involvement. We present FabricAtion, a method to extend the capabilities of FDM 3D printers to include sewing capabilities like crafting clothing, performing alterations, and repairing textiles. We introduce 3D printed needles and thread to create stitches, and pattern-based seams that can recreate all common classes of traditional hand and machine sewing patterns and can produce novel seam types. This software-only system assists in translating common traditional sewing patterns, or 'flats', into a fabrication workflow that crafts the same items without the need for a sewing machine or sewing competency. Fabrication steps are ordered and merged to maximize automation while following traditional sewing sequencing that simplify handling the fabric. During fabrication, the system requests user assistance to place and align pieces of fabric, and directs the process using registration marks and 3D printed guides. We present a set of artifacts created with the system that demonstrate crafting, alteration and repair of clothing and present a usability study to discuss the practicability of the manual user interactions.

5 FABRICATION - EXTENDING MACHINE CAPABILITIES

Tools such as a chisel or a sewing machine have very specific purposes and their usage is typically limited to this purpose. A chisel removes a plane of wood when pushed by a user. A sewing machine joins pieces of fabric together with thread or decorates fabric with thread patterns. In contrast, a 3D printer can produce almost any three dimensional geometry. This breadth is achieved through mechanical precision and continuous iteration. The output is however still limited to producing 3D objects.

In combination with a hammer, a chisel can be used as a slot-punch. Or in combination with a lathe, the chisel can carve out curved features rather than straight planes. For digital fabrication tools, it can be sufficient to simply provide new software in lieu of another tool to support different use cases.

We propose extending the capabilities of commodity 3D printers to sewing through software and manual user assistance. We provide a software tool that allows translating sewing patterns called 'flats' into 3D print jobs and assembly instructions. Rather than modifying machine hardware, we propose extending the capabilities of the machine as needed by recreating physical end-effectors on the fly. Through custom G-Code, our system 3D prints freestanding sharp needles. The printed needles replace the role of the end-effector of a sewing machine. Instead of re-using the same metal needle, our system generates a 3D printed needle on demand for each stitch. Similar to traditional sewing, the user provides pieces of fabric during the assembly. The user pushes the needles through the fabric, or aligns them with pre-cut holes. By remelting the needle after it has been pushed through pieces of fabric, the needle itself becomes the thread that joins the fabric layers together.

When interactions are intermittent, the user is afforded downtime during the process. Our software system combines and merges compatible steps of the fabrication process to maximize user downtime. Since our system uses digital fabrication tools and integrates some sewing domain knowledge, it allows fabricating highly precise seams and garments without sewing proficiency.

Beyond creating basic seams between two pieces of fabric, our system aims to comprehensively translate sewing into digital fabrication. We provide a modular pattern for printed seams that is able to recreate many classes of traditional sewing seams. Just like with a sewing machine, the seam width and length (i.e. "stitches-per-inch") of these patterns can be adjusted to fit the use case. The provided patterns allow printing temporary basting stitches, flexible seams for stretchy fabrics, and even ornamental patterns. The modularity of the patterns allows extending their use cases beyond traditional seam patterns. For example, it allows creating stiff seams to shape and reinforce sections of a garment or to include as boning.

We evaluate the strength of printed seams in comparison to traditional thread, providing a "thread strength" value that is commonly used to approximate seam strengths. Our results show that stitch for stitch, printed thread can be stronger than real thread used in commodity sewing machines. Unlike a sewing machine, the distance between stitches is not limited for printed seams, which for example allows creating wider ornamental patterns along a seam.

5.1 RELATED WORK

There are examples of prior research and commercial products that allow using fabrication machines for additional purposes. While this may match the high-level approach of our system, prior solutions tend to require extensive setup, often with specialized hardware accessories, to enable or switch between functionalities. Prior work also has made use of sheets of fabric during 3D printing, or has reproduced textiles by printing it directly. While approaches may in specific use cases circumvent the need to work with fabric entirely, they do not reproduce a sewing style of fabrication as our system does.

5.1.1 *Multipurpose Fabrication*

Prior work has transformed personal fabrication machines like 3D printers and laser cutters to allow them to automate tasks beyond typical use cases. Jubilee [112] introduces automatic tool changing to re-use the 3D gantry of a 3D printer to perform other tasks, such as plotting or fluid manipulation. Commercial products, such as SnapMaker [99] use the same approach to perform 3D printing, laser cutting, and CNC machining in one gantry, but require changing the different end effector tool heads manually. Through complex additions, a 3D printer can be used to hook and move objects similar to a robotic manipulator [48], and a laser cutter can be used to perform simple automated soldering and assembly tasks [73].

Similar to our system, these projects allow fabrication machines to perform additional purposes. However, unlike our approach, they require extensive changes to the existing hardware, or they limit the original functionality of the fabrication machine, such as limiting the build volume to avoid collisions with the additional hardware, or by reserving areas to store or operate the added mechanisms.

5.1.2 *Working with Layers and Fabric*

Conceptually related to the user interactions in our approach, Lamifold [58] allows users to assemble functional objects by inserting layers of wood and paper into the build volume of a laser cutter. Lamifold however uses intermittent interactions to simplify and streamline the assembly of laser

cut mechanisms, rather than enabling the fabrication of a new category of outputs.

Other prior work directly 3D prints onto fabric, either to supply novel textures to the exterior of 3D printed objects [113], to add printed mechanisms to prototype actuating fabrics [71], or to join the outputs of 3D printing with the outputs of other devices, such as an embroidery machine [87]. In these projects, users insert a layer of fabric to enhance the final product, but traditional sewing devices would still be required to craft fabric items such as clothing.

Finally, prior work has shown that textiles can be 3D printed, either by printing a weave of disconnected strands of filament [103], or by strategically forcing gaps via under-extrusion when printing a sheet of material [28]. Unlike working with traditional fabric, the outputs can be limited in flexibility, durability and feel of the textiles, and they still require further assembly or sewing steps to create shapes other than simple strips or cylinders. 3D printed textiles also could combine well with our system, for example by printing patches for repair applications directly, or for alteration applications by extending garments, or to join sections of fabric with strips of printed fabric that are included as stylistic features.

5.2 APPROACH AND DESIGN SPACE

The FabricAtion system assists users that have basic competency in using a 3D printer in performing sewing tasks without access to a sewing machine. Our system frequently prints multiple small features, so users should be able to identify and rectify basic printing issues, such as improper first layer adhesion. The system allows the creation of cloth items based on sewing instruction "patterns", or "flats", which contain the shapes that need to be cut out of cloth in preparation for sewing. Sewing then creates seams, that is, areas where two or more pieces of fabric are sewn or stitched together. An FDM 3D printer is used to create printed seams that connect pieces of fabric, and a laser cutter or scissors can be used to cut the fabric into pieces. During fabrication, users assist in the process by manually placing pieces of fabric onto half-finished printed seams. Sewing domain knowledge is integrated into the system to better assist users that lack sewing experience. For example, the system can automatically determine a beneficial order of fabricating the seams based on their purpose, such as handling internal features like darts and pleats before joining the outer shell of a garment. Many more sewing specific details are handled through the system, such as ensuring seams are "tied off" at the start and at the end. The printed seams need to be able to follow and fit arbitrary shapes and sizes to match sewing patterns, and machine code has to be generated for each printed seam. Different stitch types may have to be used for specific seams, for example temporary basting seams that are removed partway through the crafting process, or highly flexible stitches for garments that are designed to stretch. Stitch patterns may need to be adjusted, for example by using a tighter pattern with a higher number of stitches per

inch (SPI), for high-strength applications. Our software system makes sewing with unmodified commodity FDM 3D printers possible.

5.2.1 *Overview*

The system supports three common use cases of conventional sewing. It can be used to create cloth items from start to finish, such as a T-shirt or a plush toy shown in Figure 5.1. It can also be used to repair damaged fabrics, for example by printing iron-on patches to match and surround the damaged area. Finally, it can be used to tailor existing clothing. This is enabled by setting up certain types of printed seams in existing garments, such as incorporating darts and tucks to change the shape to adjust the fit of a garment.

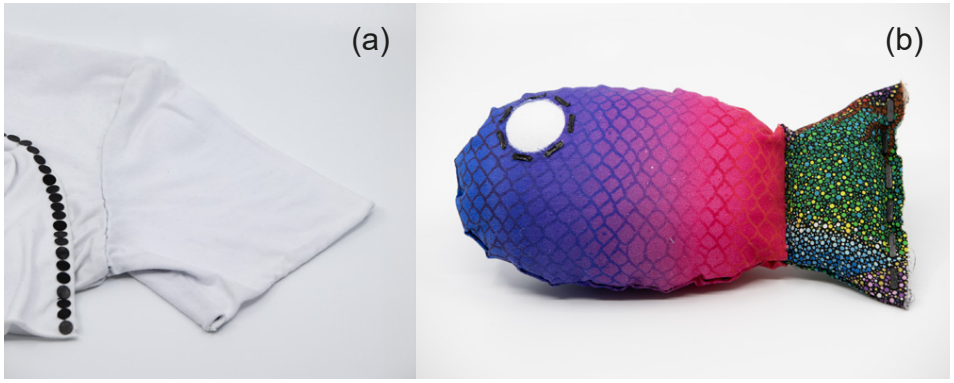


Figure 5.1: Fully fabricated fabrics, such as a (a) T-shirt with printed seams revealed, or a (b) fish plushie.

To prepare for fabrication of a cloth item, the system ensures overlapping seams do not obstruct one another and renders previews for the fabrication steps. During fabrication, the system provides a control page to help the user keep track of the progress, export cutting and printing jobs, or connect with fabrication machines to start jobs directly from the control page. Fabric can be cut either by hand using scissors and paper print-outs of the desired shapes, or by using a laser cutter. Laser-cut fabric pieces include small pre-cut holes for each printed needle to simplify aligning the fabric, and engraved markers to help identify and align pieces. When exporting or running fabrication jobs, cuts and prints are combined into groups based on the available work area of the chosen machine to minimize the amount of necessary user interactions and maximize the pause duration between interactions. On the control page, users may also override the recommended procedure, for example to change the ordering of fabrication steps on the fly, or to repeat steps when necessary.

5.2.2 *Sewing with a 3D Printer*

To make performing sewing tasks with a 3D printer possible in the first place, the printer needs to be able to fulfill the primary purpose of a sewing machine,

which is to join two or more pieces of fabric together. Sewing with thread uses one or more continuous strands of thread which are woven through fabric to join them together. A needle is used to pass the thread through the fabric, where it is then looped around itself or another strand of thread. In a lock stitch, which is common for home sewing machines, one strand of thread on top of the fabric is looped around a second strand of thread below the fabric so that they interlock at each stitch location where the needle pierces the fabric.

3D printers are not equipped with needles. During regular operation, printing starts and stops regularly instead of continuously extruding filament. Continuous extrusion is only used for printing parts specifically designed for this "vase" or "spiral" printing mode, where only a single outline of the shape is printed to craft a one-body hollow object. In short, 3D printers do not offer a direct equivalent to the threads and needles used in sewing. We introduce the concept of 3D printed needles and thread, which are fabricated directly at the location where they are needed. Printed needles and thread allow FDM 3D printers to take the place of a sewing machine in tailoring tasks and other crafting tasks that use fabric.

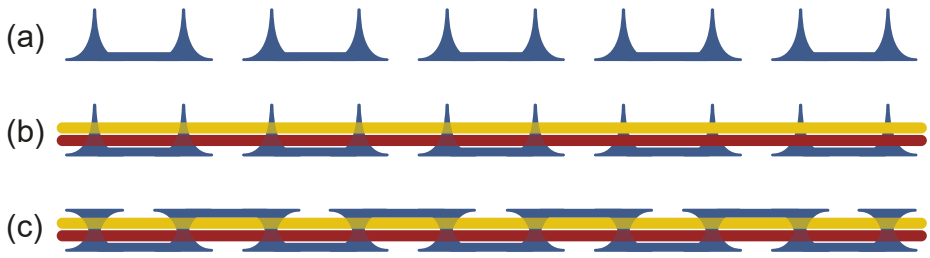


Figure 5.2: Layout of printed needles and thread with (a) needles and bottom half of the seam printed, (b) layers of fabric inserted by a user, (c) Fully printed seam joining the fabric layers together permanently. Base flare of needles is exaggerated for visual clarity.

5.2.3 *3D Printing Needle and Thread*

Printed threads are essentially just thin traditionally printed features that are placed in precise locations. In the simplest form, printed thread can be created by extruding one to a few parallel lines, one to a few layers high. Initially, printed thread for a seam is placed wherever traditional thread would be visible when looking at the bottom of the fabric. Printed needles are created using custom G-code that uses relative movement commands instead of absolute locations. We provide the specific commands for a sharp needle in Appendix B.3. Printing a needle starts by anchoring it to printed thread. The hot printing nozzle is pushed slightly into the existing filament and digs through it in a grid pattern. Then, decreasing amounts of filament are extruded while the printing nozzle moves upwards to form the shape of a needle that comes to a sharp point. Retractions are included in the G-code to

avoid stringing at the tip of the needle. Throughout the process, the movement speed of the nozzle is adjusted and brief dwell commands are inserted to provide time to either heat up the existing thread filament, or to allow parts of the needle to cool and avoid drooping.

Modular options for printed needles can be used for different usage scenarios. For example, based on the print head geometry, the maximum height of a printed needle can be limited before they may collide with the print head. For needles made from flexible filament like TPU, collisions are typically not a concern, as they simply flex out of the way. Shorter needles can avoid collision issues altogether, but taller needles can simplify aligning fabric pieces onto the seam and can help keep the fabric in place once fabrication is resumed. The printed needles can also be adjusted in other ways. For example, including hooks or blobs as part of the body of the needle can further secure fabric pieces in place after placement by the user. A hook shape can be created by briefly moving the print head down sideways and back while printing the needle, and blobs can be formed by slowing the needle extrusion movement down briefly to gather excess filament into a blob.

One printed needle is created for each stitch location where thread would normally be woven through the fabric. While printing the needles, printed thread is also created in between each stitch location to connect the needles as shown in Figure 5.2 (a). This printed thread is placed wherever traditional thread would be visible when looking at the bottom of the fabric. With the help of a user, layers of fabric are then placed on top of the needles. The fabric is aligned to ensure each needle mates with a mark or cutout in the fabric, and the fabric is pushed down below the needles as shown in Figure 5.2 (b). To complete the seam, printed thread is created on top of the fabric wherever traditional thread would be visible when looking at the top of the fabric, as shown in Figure 5.2 (c). The existing printed needles are re-melted to fuse the printed thread and printed needles permanently. The printed needles remain in place extending through the fabric, forming the part of traditional thread that passes from one side of the fabric to the other. At the end of this process, the pieces of fabric are joined together permanently with printed filament weaving through the fabric in place of traditional thread. By following this process, FDM 3D printers can create structures that fulfill the same purpose as traditional seams sewn from thread.

5.3 WALKTHROUGH

Making something from fabric using our technique follows two overall stages, preparation and fabrication. Preparation means making the information from the sewing instructions available within the system, and fabrication means turning the digital layout into a physical object.

5.3.1 *Starting from flats*

The first step to craft fabric items is the same as it is for traditional sewing, to make or acquire a set of sewing instructions or fabric flats. The essential part of the sewing instructions are the outlines of the fabric shapes that need to be cut, as well as the seam allowance or hem allowance included in the outlines, if any. The seam allowance describes how far away from the edge of the fabric seams are placed when sewing this pattern. We will walk through creating a a simple fish plushie using the system.

The web-based FabricAtion software system and interface is an extension and conceptual repurposing of the Joinery system introduced in prior work [126]. All the original functionality of the Joinery system remains available, such as loading line drawings, selecting pairs of lines to create a "joint" that connects them, saving the current state as a project file for re-use, and exporting an adjusted line drawing including the planned joints. The system has been extended to allow the fabrication of 3D printed seams, default to recommended sewing practices, and guide users through batched assembly steps of larger sewing projects, among other changes.

5.3.2 *Defining seams*

To start the making process, a user loads line drawings that contain the fabric shapes into the system. For example, for the fish plushie example, there is one flat provided for the body and one for the eyes as they are intended to be cut from two different types of fabric. The shapes can then be arranged manually to organize them.

Before fabrication, all seams that need to be fabricated need to be defined. To do so, the user selects two compatible lines of the loaded line drawings, or selects one line and clicks a button to connect a line to itself. This is common when cutting a shape "on fold", where the shape is placed along the folded edge of a piece of fabric to cut two copies at once. A stitch type can be assigned to each connection afterwards. For the fish plushie we use the running stitch from the set of default stitches, and assign this stitch type to each seam. If necessary, parameters for the stitch type can be adjusted, just like changing the length or width of a stitch when using a sewing machine.

Defining seams can be done by setting up each seam as they are mentioned in the sewing instructions, which ensures a correct order of operations when making the seams, as the default setting follows the order of definition. For the simple fish plushie, the eyes need to be connected to the sides, the long seam around the body needs to be done afterwards, and the small seam at the tail fin needs to be done last to close the plushie after filling it with stuffing.

5.3.3 *Printing the lower "threads" and the "needles"*

After setting up all the seams, the project is ready for fabrication. Clicking the fabricate button opens a control panel in a new browser window that shows

previews for all planned fabrication steps, including fabric cuts to be made and seams to be printed. Fabrication steps are laid out in the recommended order of operations. Printing seams requires access to the fabric cut-outs, so the printing operations are laid out in groups underneath the cutting operation that enables them. For the fish plushie, both types of fabric for the body and for the eyes need to be cut before the eyes can be attached to the body. The control panel always recommends both a cutting and a printing operation, as printing takes time and is largely automated.

The user starts the first printing operation by clicking the "print now" button next to the first highlighted printing step, which sends a print job directly to a 3D printer using a connected OctoPrint server [39]. For the fish plushie, the first two recommended printed seams fit on the build plate of the user's 3D printer, which are the seams used to connect the eyes to the fish body. The machine commands for both printed seams are combined into a batch, such that the 3D printer completes as much of the print job as possible before the first user interaction is required. In particular, all printed thread, printed needles, and all markers for all seams that are part of current batch are fabricated autonomously. This setup maximizes the length of the break between starting the print job and the first user interaction.

5.3.4 *Cutting fabric shapes*

While the 3D printer is working, the user handles cutting out the fabric shapes. The shapes can be exported for paper printing to cut them out using scissors as in traditional sewing, or as line drawings for laser cutting. The exported shapes contain additional line features to simplify the assembly, such as pre-cut holes for each stitch needle location and cutouts along the fabric edge to match 3D printed markers. The markers and also the needle holes ensure the correct selection and alignment when placing fabric pieces, especially when assembly requires handling multiple similar fabric pieces. When using scissors to cut out fabric shapes, a hole punch can be used to cut out holes at needle locations. The user can also skip this step and instead push the sharp 3D printed needles through the fabric during assembly, which requires minimal effort for many types of fabric, but the fabric alignment may be less precise when done this way. When using a laser cutter, the system optionally also exports engraved markers to avoid fabric cut-outs along the edge. Placing painters tape on top of the fabric at marker locations can prevent marring the surface of the fabric.

With the fabric pieces for the body and the eyes in hand, the user is free to take a break until a 3D printer requests user assistance. The time until the next planned user interaction is constantly updated on the printer status display. The progress percentage status is set to show the percentage of how much of the print job as a whole has fabricated already, as usual.

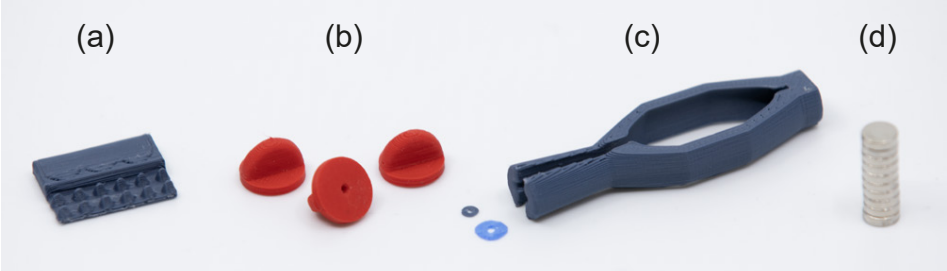


Figure 5.3: Layers of fabric may be secured in place using (a) magnetic clamps, (b) rubber pin backs attached to needles, (c) platelets that merge with the printed seam, or (d) common tools such as magnets or tape.

5.3.5 *Printing the upper thread onto the needles*

With a message and a beep, the 3D printer will request user assistance after printing the bottom seams, needles, markers, and ID written on the build plate. Given the ID of the seam, the user can check the control page to see a preview of the fabricated seam to simplify picking the right fabric pieces. The user then places the fabric pieces onto the build plate, making sure that the printed markers match the fabric cut-outs, and that the printed needles slot into the pre-cut holes. The printed needles help keep the fabric in the correct location, but depending on the seam and type of fabric, the fabric layers may need to be secured in place further. To do so, the user can apply rubber pin backs shown in Figure 5.3 (b) to printed needles to lock fabric in place underneath temporarily, but the pin backs need to be removed before the print head reaches the places where the pin backs have been applied, as they would physically disrupt the printing process. Alternatively, fabric may be secured with magnets or clamps, shown in Figure 5.3 (a) and (d), or by applying painters tape as demonstrated in prior work [71, 85, 113]. We also provide options to lock the fabric in place with thin discs made from the fabrication filament itself, which merge into the printed seams during the process, or to hold the fabric in place using magnetic clamps, shown in Figure 5.3 (c). After clicking a button on the printer, fabrication continues to "pick up" the printed needles from the top side of the fabric, which can be done by just re-printing the needle shape again, as well as printing the remaining seams on top of the fabric.

5.3.6 *Subsequent Interactions*

For the fish plushie, completing the first print job attaches both eyes to the body halves of the fish. The next recommended fabrication step is printing the seam that forms the outer shell of the plushie surrounding the fish body. The user repeats the process of starting a print, taking a break, and placing the fabric onto the build plate during an intermittent interaction. Since this seam forms the outer shell of the plushie, it will be turned inside-out later, so the outsides of both layers of fabric should be touching as the two layers

of fabric are placed onto the needles on the build plate. Once this seam is finished, the fish plushie can be turned inside-out, stuffed with filling, and fabricating the last seam closes the last opening, which traps the filling and finishes the plushie, which can be seen in Figure 5.1 (b).

5.3.7 *Repair and Alteration using Making Tools*

It is also possible to use the system for alterations or to repair damaged fabric items. The tools and stitch patterns offered can be used to create iron-on or sewn patches to reinforce damaged areas, or to make garment shape alterations such as darts, tucks or pleats. These operations will typically require making a specific line drawing appropriate for the specific use case. For alterations, this is typically just a straight line of a specific length. For repair, an outline of the damaged area of the clothing can be used.

5.4 STITCH PATTERN DESIGN SPACE

There are many different stitch type options available when using a sewing machine and when sewing by hand. The different stitch types provide a variety of aesthetic and mechanical options, as well as special use cases, such as providing reinforcement surrounding a button hole. We briefly describe how sewn seams are created and then explain how our pattern-based approach can recreate existing seam types.

5.4.1 *Seam types*

3D printed seams can recreate the majority of class 100 to class 600 ISO standard stitch types. This includes hand stitches and chain machine stitches, which use a single thread and are typically fairly flexible, but prone to unraveling after a single breakage of the seam. Figure 5.2 (c) demonstrates an example of recreating a simple running stitch with a single continuous printed thread. Unlike traditional thread, printed seams are less prone to unravel after a single breakage as printed seams can adhere to the fabric, making them resistant to slipping through stitch holes after breaking.

Lock stitches use two or more continuous strands of thread which are interlocked regularly. Lock stitches can be recreated as printed seams by simply adding additional connections between printed needles. In particular, a lock stitch is formed if a closed loop exists that starts at a printed needle, visits both sides of the fabric by following along printed seams or needles, and returns to the starting printed needle. Lock stitches using traditional thread are resistant to unraveling from a single thread breakage, but the breakage may still propagate with agitation over time. A local breakage in a printed lock stitch can never propagate globally, but printed lock stitches tend to be stiffer than traditional ones.

5.4.2 *Parameterized stitch pattern*

The FabricAtion system can be used to define a variety of different stitch types using a parametric pattern approach that places, orients and scales modular low-level machine code "primitives" for printed needles and thread. Each stitch pattern is primarily defined through three main factors, which are the number and placement of "stitch lines" parallel to the fabric edge, the placement of needle holes along those lines, and connections between those needle holes. Each pattern also defines additional parameters, such as the average distance between two stitches, which implicitly sets the SPI, or the default width of the pattern.

On household sewing machines, the distance between stitches and the width of the pattern can be adjusted to customize the predefined patterns. The parameters of FabricAtion stitch patterns can be adjusted just the same to alter the appearance and functionality of each pattern. As an example, the class 304 machine lock stitch, or "zig-zag stitch", alternates back and forth between two parallel lines that are equidistant from the center of the presser foot. To recreate this seam shown in Figure 5.4, two parallel lines that are some distance apart have to be defined in the pattern, which is done by providing a list of distances from the center, for example the list `[2,-2]` to achieve a default width of 4 mm.

By default, needle holes will be placed along the first line based on the set SPI. To match the zig-zag pattern, each odd-numbered needle hole should be located on the second parallel line instead, which can be done by providing a list of integers that selects which of the stitch lines should be used for that hole: `[1,2]`. Patterns provide the option to slide needle holes back or forth along their stitch line. Since stitches are placed at regular distance intervals in the zig-zag pattern, doing this is not necessary, so this offset should be set to `[0,0]`. Offsets are provided in units of the default distance between stitches, so setting the offsets to `[0,-1]` would make sure the two needle holes are right next to each other, perpendicular to the fabric edge.

Finally, connections between needle holes need to be defined to place printed threads. Connections are provided as integers by targeting the n-th next needle hole to connect to, by using 0 to denote a skipped connection, or by denoting 0.5 to "tie off" this needle hole without connecting it to another one. Each stitch is connected to both nearest neighbors in the zig-zag pattern, which can be recreated by providing the connection list `[1,1]`. The pattern input definitions do not need to match in length, so simply providing the connection list `[1]` achieves an identical result. Providing definitions with mismatched length can also be used to create irregular patterns that only repeat after several iterations.

A straightforward way to define patterns with multiple connections between needle holes, akin to a sewing machine going back and forth during operation, is to duplicate and optionally adjust the provided pattern details lists, and halve the average length between two stitches.

The far edge of a stitch pattern is typically located a given "seam allowance" away from the fabric edge. To accommodate class 500 over-edge stitch patterns

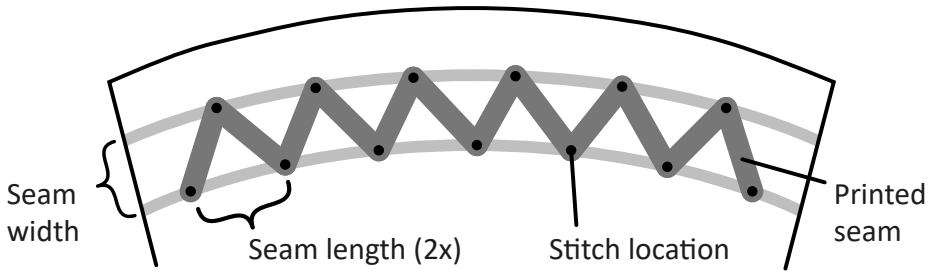


Figure 5.4: Printed stitch pattern overview and basic parameters to adjust their look and functionality.

that extend beyond the fabric edge, the locations of stitch lines can optionally be defined not based on the hem distance, but directly from the fabric edge. For example, the list `[0,"o0"]` would place one stitch line directly on the hem line, and a second stitch line directly on the fabric edge, where the letter "o" denotes targeting from the outside edge of the fabric.

Fabric may also be aligned and folded differently in preparation for sewing. Printed seams can directly create superimposed and flat seams, and can fashion lapped and bound seams by recreating them in multiple steps, for example first creating a hem before finishing the full lapped seam.

5.5 IMPLEMENTATION DETAILS

Our system is based on joinery, a parametric joint generation tool for laser cut assemblies [126]. Joinery provides a fixed set of joint profiles that are specifically designed for use with a laser cutter. The tool allows designers to define connections between parts of an assembly and then automatically generates the joints. We provide an extension to the system to automatically generate 3D printed seams, handle many classes of stitches through an extendable pattern based approach, and allow the system to guide users through the assembly process.

Within the software system, seams are assigned to lines or paths of an input line drawing. To ensure compatibility with other seams, the specific layout of all seams is re-evaluated whenever a new seam is added.

5.5.1 *Planning seam locations*

In traditional sewing, sewing seams that intersect are trivial, but for printed seams, this could potentially cause problems during fabrication. For example, when multiple printed needles occupy the same location on a piece of fabric, this may make it difficult or impossible to properly insert and align the fabric while the printer is paused, as previously printed needles may be physically in the way. To avoid issues during assembly, the system scans for possible collisions between printed areas and can skip affected stitch locations. The

neighboring locations will be connected instead to still maintain the integrity of the seam.

To plan printing a seam, the system first determines the location of stitch holes along the fabric edge using the selected stitch pattern. A printed needle will be placed in each stitch location. As many stitches as possible are fit into the seam, while maintaining at least the length between two stitches distance from the corners, to ensure stitches don't cross past the fabric edge. The group of stitches is centered, so the seam starts and ends the same distance apart from the corners.

5.5.2 *Machine code generation*

Starting with a group of stitch locations, printed seams are then placed based on the selected stitch pattern. Overall, the system performs custom low level machine code generation rather than using a traditional slicing approach to fabricate printed seams. Modular snippets of machine code to print the seams are placed based on combinations of a start and a target stitch location. As a convention, the code snippets are provided using relative distances, rather than absolute locations. The origin of this set of machine code should match the placement of the starting needle location, with the end needle location placed straight to the right of the origin. An origin and rotation offset can be provided to use code that does not follow these conventions, and we provide a Visual Studio Code [63] plugin that can be used to automatically turn machine code with absolute positions into relative coordinates.

To add printed seam machine code to the fabrication plan, the code snippet is adjusted as needed. First, the print head is moved to the proper start location, with optional filament retractions placed around the move to avoid filament drooping during travel. Then, each line of the machine code snippet is parsed to extract movement and extrude commands, separated into axes and extrusion distance. Each machine code command is rotated around the starting location and scaled as necessary to connect the starting and ending stitch location through the printed seam, and optional extrusion and speed modifiers can be factored in. The modular machine code snippets can be provided with multiple variations that each create the same type of seam, but along a different seam length. The system selects the variant with the default length closest to the target length to avoid excessive scaling of machine code, as advanced slicing features such as wipes at the end of a seam may not work as well at a different scale.

5.5.3 *Printed guides*

Text guides and arrow markers on the build plate can be added in a similar manner, printing lines that trace characters or shapes, which are repositioned and scaled to place them next to the printed seams. Fabric cutout markers are added directly to the line drawings that are either printed to be cut out with scissors, or sent to a laser cutter for automated cutting. The simplest form of

these markers is created by adding a circle to the fabric edge, which creates a half circle shaped cutout. The 3D printed markers that fit into the cutouts are generated on the fly to match the line data directly. The line data is sent to a local web server that generates machine code to print a small prism that fits perfectly into the cutout. In detail, the server uses the line data to create a matching 2D polygon in Open-SCAD [52], which is extruded linearly, and the resulting prism mesh is stored briefly. The server generates machine code by slicing this mesh, and returns only the part of the code responsible for printing the prism.

Typically, two simple circle markers placed at random locations along the outline of the fabric piece are sufficient to match and align pieces deterministically. For larger projects, additional markers can be placed, or the system could place variations of cutout shapes, like wedges, squares, or assemblies of cutouts. This generation process is performed for each seam that is selected as part of a combined print job in preparation for fabrication.

5.6 TECHNICAL EVALUATION

We have demonstrated how 3D printers can join pieces of fabric in a manner similar to traditional sewing, and that the 3D printed stitch types of that approach are varied, versatile, and adjustable, much like machine sewing stitches. However, 3D printed seams are likely only of interest to users if they can also achieve similar durability compared to traditional sewing. We provide a technical evaluation of the ultimate tensile strength (UTS) of 3D printed seams compared to machine sewn seams. Overall, our evaluation shows that 3D printed seams can provide comparable durability, as they can be strong enough that the fabric itself may rip before the seam does, and they can match the strength of the tightest available 301 lock stitch setting on the sewing machine used for testing. We found no visible damage to printed seams after washing and drying clothing with fabricated seams using residential laundry appliances.

5.6.1 *Apparatus*

We created a device to measure the ultimate tensile strength of seams connecting one inch strips of fabric modelled after the ISO 13935-1:2014 and ASTM D1683 standards for testing seams until failure. During the test, samples are pulled apart using slowly increasing static tension. Figure 5.5 shows the layout of the test apparatus, which is operated by hand using a clutch clamp. The fabric strips are gripped along the full width of the strip and to prevent any slipping, strips are wrapped 180° around the grip bars, and sandpaper has been applied to the gripping surface. The gripper that is being pulled is attached to a sled that restricts its movement to a straight line perpendicular to the seams. The second gripper is held in place by a tensile force gauge set to measure the peak tensile force applied. We used a ZMF-500N digital

force gauge, which measures up to 500 N in increments of 0.1 N, to measure ultimate tensile strength.

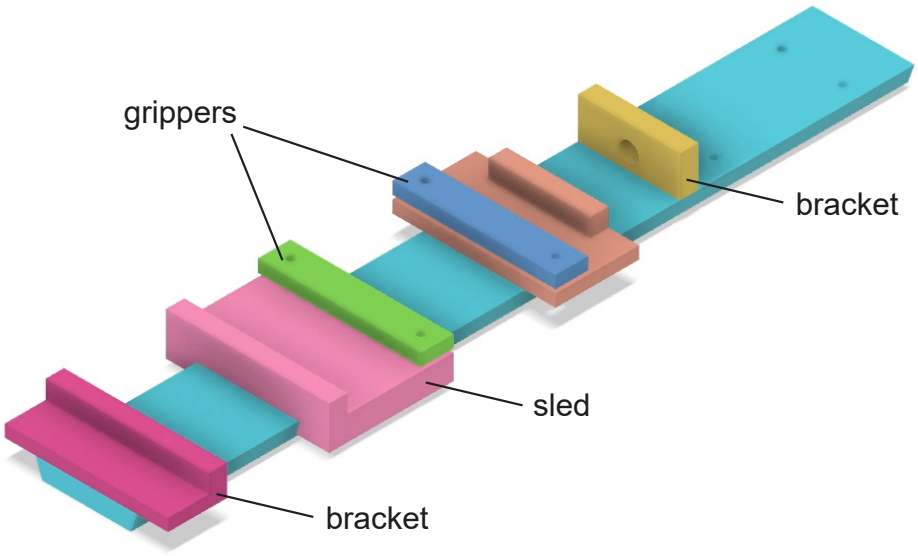


Figure 5.5: Test apparatus to measure ultimate tensile strength. The green and blue grippers hold strips of fabric, and the pink sled is moved away in a straight line from the strength tester behind the yellow retainer bracket.

For all following tests, we used a 0.8 mm thick 100 % woven cotton fabric that was laser cut into 35.4 mm wide, which is one inch plus 5 mm free space on both sides, and 130 mm long test strips cut at a bias of 45 °. This rotation of the fabric before cutting the test strips avoids incidentally aligning the pull direction with the fabric weave, which could otherwise considerably compromise the performance of the fabric in tensile strength tests based on small differences in the rotation.

We validated our test apparatus with the help of a dedicated mechanical engineering lab and their professional grade material testing equipment. Members of the engineering lab confirmed that the design and operation of our test apparatus is appropriate for the intended UTS tests. We also performed three sets of three tensile strength tests both using their Shimana SHPMFG177 1000N and our test apparatus to compare the results. For all three test sets among the two conditions, the averages placed within the 95 % confidence interval of the test set of the other condition.

For all following experiments, five repetitions were performed per condition. For each test, a 1-inch long seam centered along the width of the strip was used to connect two strips, using either traditional or 3D printed thread. All following error bars show the standard error.

5.6.2 *Experiment 1: Seam Strength by Material*

We compare the UTS of 1-inch long seams when using different 3D printing filaments, as well as two types of traditional brand name thread as a baseline. Each seam was made with an SPI of 7, which means that exactly seven stitches form each of the seams, for both 3D printed and traditional thread. We test a variety of 3D printing filaments, including different brands and types for the most commonly used types of filament. As a baseline, thread with TEX values between 25 and 35 are typically recommended for sewing garments, and most stitches per inch values between 7 and 14 are recommended for most types of clothing [40]. We used Double Duty 30 TEX polyester thread and Coats and Clark 33 TEX cotton thread to test traditional seams, which is the highest TEX that could reliably be sewn with the Janome C30 sewing machine used during testing. The polyester thread caused regular jams while sewing, so the cotton thread has been used in all further testing of traditional seams. The 301 straight line lock stitch, which is the default on the used sewing machine, and its 3D printed equivalent were used during these tests. 3D printed seams were fabricated on a Prusa MK3s 3D printer with a 0.4 mm nozzle, using 4 layers of 0.2 mm filament to print the seams, for a total height of 0.8 mm, at a width of 1.6 mm. Nozzle and bed temperatures have been adjusted to a recommended value for each filament.

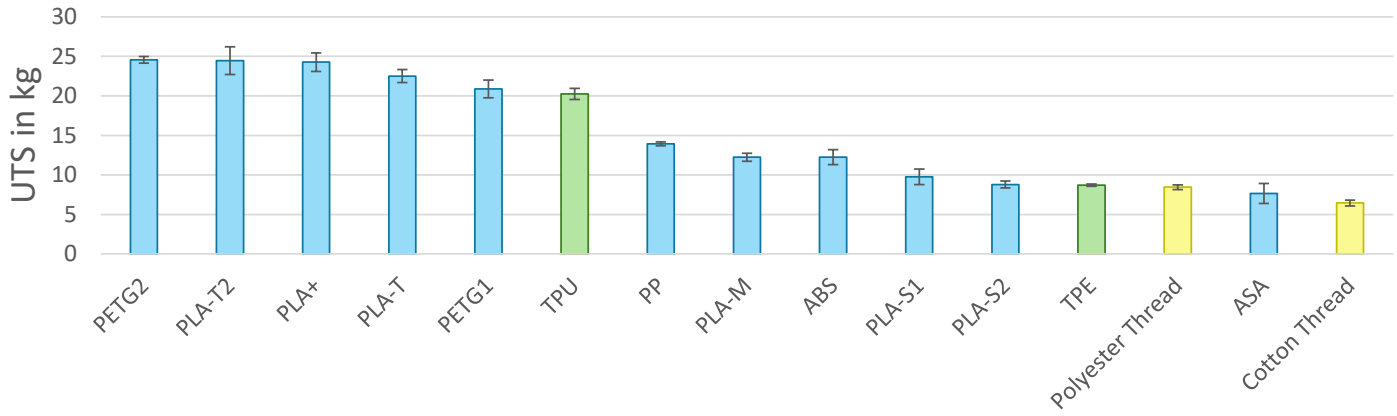


Figure 5.6: Experiment 1: Ultimate tensile strength of a 1 inch seam by material type. Green bars denote flexible 3D printing filaments. Yellow bars denote traditional thread.

The results show that stitch-by-stitch, popular 3D printing filaments like PLA+, TPU and PETG achieve a much higher UTS compared to traditional thread commonly used in home sewing machines, as shown in Figure 5.6. The filaments that achieved the highest UTS, like PLA+ and PETG, are ductile enough to bend with the fabric before breaking and have great layer adhesion. TPU provides a high UTS and is flexible, which can be desirable for the flexibility of a seam. Materials that are brittle (PLA), have weak layer adhesion (ASA), or are too ductile (TPE) are poor choices for 3D printing seams.

5.6.3 *Experiment 2: Seam Strength by Hole Size*

3D printed needles made from stiff filaments like PLA+ or PETG can easily penetrate most types of fabric. Printed needles made from flexible filament like TPU can reliably only be pushed through fabrics that are very stretchy, or woven or knitted sparsely. Pre-cut holes in the fabric pieces allow using any combination of fabric and printing filament and they simplify aligning the fabric to the printed needles during fabrication. If the holes are small, it can still be challenging to match them exactly. If they are too big, the spacing around the printed needles offers a lot of play, which can make the final alignment less precise. We found the ideal hole size range for both ease of placement and precision to be between 1 mm and 1.5 mm.

The pre-cut holes also reduce the overall strength of the fabric itself, especially for fabrics that do not melt when heated, like cotton. We provide UTS tests based on the diameter of the pre-cut holes in Figure 5.7. For the tests, we used TPU filament with a 0.8 mm thick tightly woven cotton fabric, where the printed needles can not be pushed through the fabric outside of the pre-cut holes.

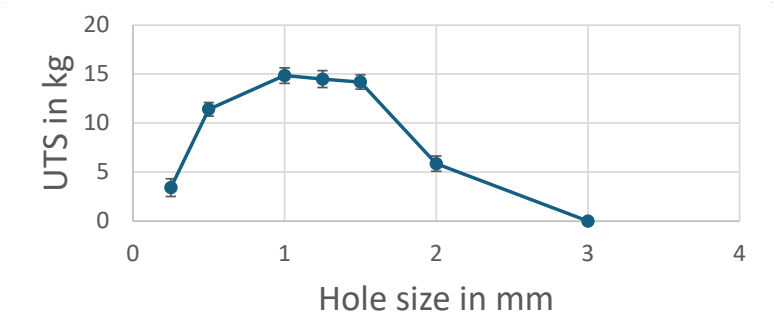


Figure 5.7: Experiment 2: Ultimate tensile strength of flexible printed seams based on the size of pre-cut holes.

The results show that the tensile strength of the seams drop off considerably when the holes are too small or too large. The 3 mm diameter holes essentially fully break the weave of the fabric, as the center to center distance between two holes at seven stitches per inch is only 4 mm. At 0.5 mm hole diameter and below, the holes are too restrictive to reliably push the flexible needles through to the other side. Functional seams can however still be created when using very small holes or using no holes at all. Especially when using flexible or thin fabric, both the printed needle and the printing nozzle can create and widen the hole as needed.

5.6.4 *Experiment 3: Seam Strength by Stitch Pattern Parameters*

3D printed stitch patterns can be adjusted in width and length just like machine stitches. We test the UTS of seams with varying pattern parameters to show the effect of these adjustments on seam strength.

In the simplest possible setup of a 301 lock stitch, where the stitches per inch can be varied by changing only the distance between each stitch location on a straight line, both printed and traditional seams show a linear progression of UTS with varying SPI, as shown in Figure 5.8.

Sewing machines limit the width and length of stitch patterns based on the available hardware. The maximum pattern width on the machine used in the tests is 5 mm. For 3D printed seams, there are no hard limitations. When using pre-cut holes however, the minimum distance between stitch locations is effectively limited by the hole size. More generally, the relationship between the number of traditional stitches in a unit length and seam strength is known to be "complex and non-linear" [92]. For 3D printed seams, this likely also holds true, but with different complex relationships.

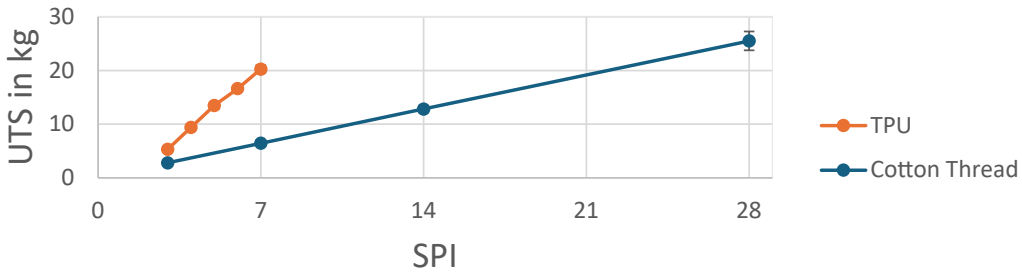


Figure 5.8: Experiment 3: Ultimate tensile strength shows linear progression for both printed and traditional seams.

The results show that printed seams can provide compression resistance, especially with fully connected members like in the 3D printed zigzag lock stitch used in the test. The stitches placed on the second zigzag stitch line further away from the fabric edge contribute to the overall strength of the seam, but for the 3D printed lock stitch, the UTS approaches the baseline of only printing the stitches on the first line closer to the fabric edge with increasing pattern width. In contrast, when a 304 zig-zag lock stitch becomes wide enough, the stitches can almost fully collapse back onto a single line by folding the fabric 180° within the width of the seam, as can be seen in the non-linear progression of the cotton thread UTS in Figure 5.9. These complex behaviours will differ with stitch types and the type of fabric used.

5.6.5 *Additional Test: Needle Size*

We also conducted a smaller test to examine whether the size of the initially printed needle has a significant impact on the strength of a printed seam. We tested small needles that are about 2 mm high and tall needles about 6 mm high and found no noticeable impact, with almost identical average UTS results of 25.2 kg and 25.0 kg respectively.

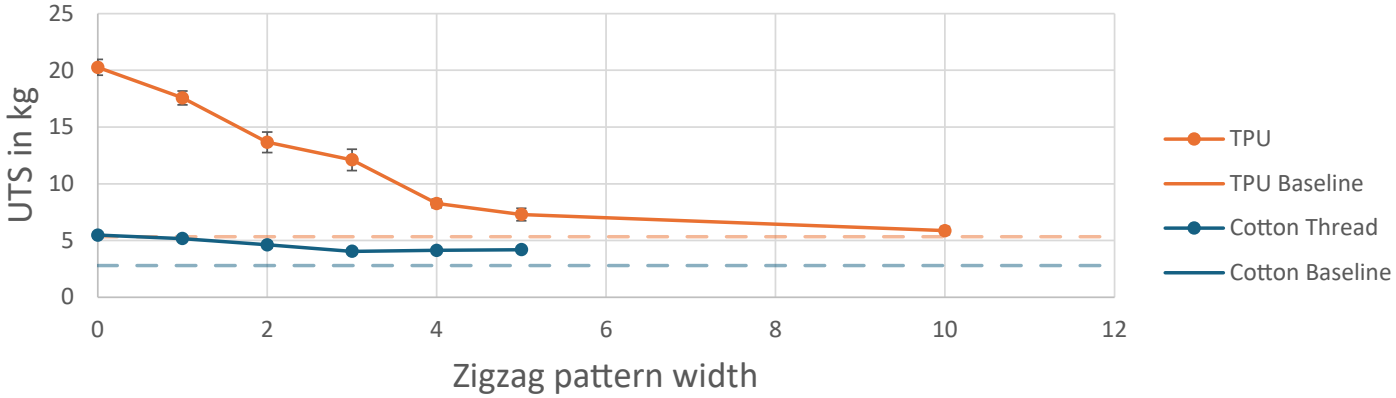


Figure 5.9: Ultimate tensile strength shows non-linear progression for varying stitch pattern widths.

5.7 DISCUSSION

We introduced a software system that provides a direct mapping of the majority of traditional sewing techniques afforded by a desktop sewing machine. The system's main focus is enabling a workflow that allows reusing traditional sewing patterns and fabric flats, as well as following typical sewing instructions. Our technical evaluation situates printed seams well next to traditional seams regarding material choice and pattern parameters.

However, our approach and evaluation have limitations. Our evaluations use a single baseline physical thread thickness. In traditional sewing, the choice of thread can considerably influence the strength of a seam beyond which material it is made from. The TEX, or "weight" of the thread, which describes its density and thickness, can significantly alter the strength of a seam, but using a higher TEX thread is not always possible. For example, thread that is too thick may not fit through the eye of a needle, or a thread that is too heavy can cause jams when using a sewing machine. We chose a thread thickness that is commonly used in home sewing which is also the maximum thread size that can be used with our given sewing machine without making modifications.

Much like limits to thread thickness on a sewing machine, we chose a "thickness" for the printed needle and thread that is well suited to commodity 3D printers equipped with a common 0.4 mm nozzle. Changing the TEX equivalent of 3D printed seams requires both adjusting the width or thickness of printed threads, and adjusting the speed and extrusion amount while printing needles.

Varying the density of printed seams would be interesting for both high strength applications or thin, lightweight garments, but may be challenging to fabricate based on the available hardware. For example, significantly thinner or thicker needles may be difficult to print with a 0.4 mm nozzle.

Overall, our results suggest that the combinations of printed string and printed needles in our technical evaluation were adequate, as failures occurred throughout all parts of the seam as well as the fabric within separate test sets.

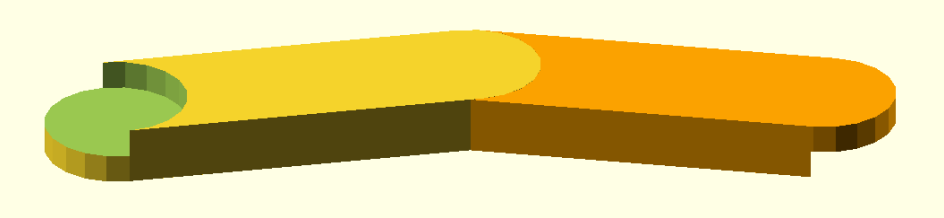


Figure 5.10: Printed seams can be designed to overlap and merge seamlessly to form sturdy bars.

Although our system description and technical evaluation focus on recreating and comparing to machine sewn seams, our system offers many options that extend beyond traditional sewing, alone, such as adding boning structures or apply embellishments that would typically require an embroidery machine.

5.7.1 *Additional variety in seam properties*

By using thicker printed seams, the printed thread can be tuned to mechanically behave more like a bar than a rope, resisting compression and bending in addition to tension. Overlapping stiff printed thread can create a bar stitch, which allows stiffening specific parts of clothing, as shown in Figure 5.10. This allows for example to create features like shoulder pads or to stiffen sections of a suit or to add boning to a corset directly through the choice of stitch pattern.

In contrast, some applications and fabrics require particularly flexible seams, to allow the cloth to stretch, compress or bend extensively. Printed seams can add additional flexibility and extendability by connecting stitch holes through curved or angled lines, instead of the straight lines necessitated by traditional sewing. For example, printing thread in an S-curve pattern offers a longer maximal extension without sacrificing ultimate yield strength. Printed seams can also be crafted from materials such as TPU that naturally stretch under load, further increasing the flexibility of a seam.

5.7.2 *Printing beyond the seams*

More generally, printed seams can follow arbitrary paths to connect stitches, use patterns that fill areas rather than using lines, and add features that are not directly connected to the stitch holes.

By making use of such options, the design space of printed stitches further extends beyond recreating existing stitch types. For example, adding printed thread far beyond the space between the actual stitch hole locations allow including printed appliqué, embroidery, lace or tassels along the edge of the fabric directly into the creation of a seam.

Finally, breaking up continuous sections of string requires tying off and starting a new seam for every separate section. When using printed stitches,

stitch holes do not need to be connected to any other ones. Fabric can for example be joined together by printed rivets, where a thin disc is placed on top of and below each needle to clamp the fabric in place. Using small discs that can be pulled through the fabric creates a break-away seam, which can be useful to create a temporary basting seam. The force required to break the seam can be tuned by adjusting the size of the rivet discs, for example to attach tear-away tags, or for safety seams that rip when excessive force is applied.

Disconnected point connections can also be used to include 3D printed button snaps or other accessories that are typically sewn in manually.

5.7.3 *Future Work*

Our system successfully recreates sewing tasks using fabrication machines, but since these machines remain unmodified, they also offer the full range of typical fabrication applications. Studying how sewing or fashion experts would make use of the system or which additional functionalities they would request could allow investigating how fabrication may not only recreate, but also further benefit sewing applications.

The FabricAtion system can be used to alter or repair garments that are damaged or do not fit properly by providing simple, custom line drawings for each operation. For example, a simple straight line with the length of the dart seam that will be added to the fabric can be used to set up this kind of alteration. Or the outline of a damaged area of a piece of clothing can be used to fabricate seams to apply a matching patch. In future work, the system could be extended to include tools to directly support alteration and support workflows, for example by allowing users to upload a picture of a garment and draw seam lines directly onto it, to help visualize the intended changes.

Currently, to fashion seams that require folding the fabric requires creating the seams in multiple steps, where the fabric is folded in between printing seams, or for the user to create a fold by pushing the printed needles through two adjacent rows of pre-cut holes at the same time while placing the fabric. It may be possible to include non-printing moves when printing a seam that move past the edge of the fabric, lower the nozzle down to the build plate, and push up against the fabric edge to push and fold it over. The inclusion of such custom moves to manipulate the edge of the fabric could allow printed seams to create additional types of traditional sewn seams, beyond lapped and flat seams, automatically.

A clothing iron can be used to apply heat activated patches and to make a hem using hem tape. Theoretically, any tool that can provide sufficient heat and pressure could be used for the same purpose. By supplying a custom stitch type, the 3D nozzle of a printer may be used to both print a patch of heat activated material, and then later push a layer of fabric onto the patch and heat it to bind them together, replacing the role of the clothing iron.

It may be possible to make layers of fabric stick to each other by "injection molding" a seam in between them, which may be another option to achieve

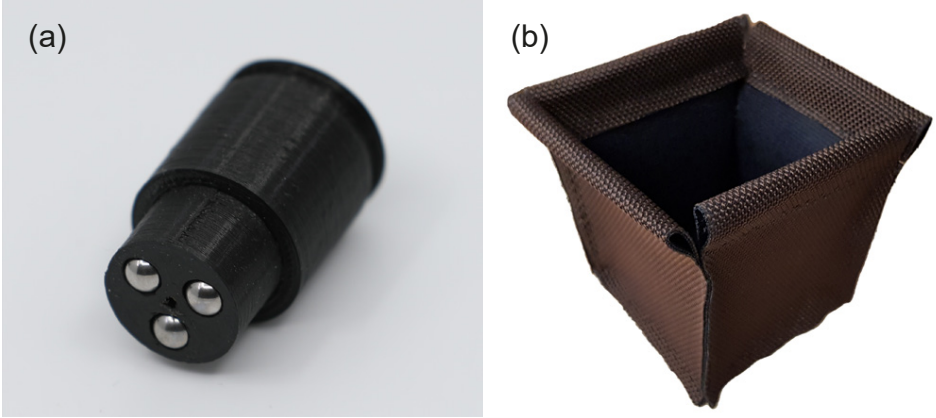


Figure 5.11: Fabric can be joined with a laser cutter, using (a) a passive operation laser tool head attachment, to create (b) a standing bag joined by welding layers of fabric together.

an effect similar to using hem tape. By arranging two layers of fabric where only the top layer has pre-cut holes, the nozzle could be lowered into the hole onto the lower layer. Extruding filament in that location could spread a patch in between the two layers that holds them together after cooling. By using any of the above mentioned methods that use or replace hem tape, it may be possible to craft "invisible" seams that are fully hidden when looking at the clothing item from the outside.

Certain fabrics melt when heated to high enough temperatures. As demonstrated in LaserStacker [109]. This effect can be used to weld layers of material that are stacked on top of each other together at specific locations using a laser cutter. This bond however can only form when the layers touch as they are melted by the laser. A simple, passive attachment to the end effector of a laser cutter can apply pressure to layers of fabric to ensure that they make contact as they are cut. The attachment shown in Figure 5.11 (a) uses spring-loaded ball bearings to gently press down on the fabric surrounding the cutting location during operation, and can be used to join fabrics directly within a laser cutter, as demonstrated by the standing bag in Figure 5.11 (b).

5.8 CONCLUSION

We have introduced a software only system that extends the functionality of a FDM 3D printer to perform sewing tasks by incorporating brief user assistance into the fabrication process. We show how modular pattern based machine code generation can map the majority of the design space of stitch patterns to 3D printed seams, and describe how the code can be dynamically adapted to vary pattern parameters, printer and filament settings, and maintain a continuous timer to track the next planned user interaction. We have compared the mechanical strength of printed seams to traditional seams and shown that printed seams can match or exceed the strength of sewn seams that follow industry recommendations for typical use cases. By enhancing

the capabilities of commodity fabrication hardware, we increase access to new making, alteration and repair options, promoting local alternatives to unsustainably produced and quickly thrown out fast fashion.

5.8.1 *Summary*

With the conclusion of this chapter, we have investigated approaches to utilize intermittent interaction in fabrication across the user involvement spectrum while largely focusing on the outcomes of these processes. During personal fabrication, users are involved in the process of creating the final outcome and the level of participation can vary considerably. Fully automated fabrication can approach almost as little involvement as buying an item from a store. Interactive fabrication that is fully driven by the user on the other hand approaches manual crafting, but with system guidance. Intermittent interaction describes the full participation spectrum, and in the following chapter, we investigate how people perceive manual involvement in semi-automated creation tasks over extended durations across the entire spectrum. A LEGO assembly task with timed lock boxes simulates human involvement with a semi-automated machine process, similar to a 3D printer. This is used in an in situ study with 12 participants over 4-hour sessions with experimental controls for number of interruptions and step complexity. Complex interactions during assembly can amplify the perceived value of the assembled object and increase enjoyment. Participants used either a clustered or evenly distributed strategy to schedule interactions; we show these can be modelled with simple heuristics. We contribute evidence that intermittent interaction is generally acceptable for creation tasks and practical guidelines for integrating intermittent interactions into semi-automated fabrication systems.

6 EVALUATING INTERMITTENT INTERACTIONS

Many digital fabrication processes require a user to periodically intervene to do something that would be impossible or impractical for a fabrication machine. Consider using a fused deposition modeling (FDM) 3D printer. In between starting a print and removing the result from the build plate several hours later, the user may also need to replace empty filament spools, switch between filament colours, and even handle issues like clogged nozzles or collisions at various times during the print. Some of these manual tasks can be automated with additional mechanical complexity and cost, like switching filament colours, but this is often not practical for most printers. Moreover, some tasks are almost impossible for a machine to handle, such as clearing a clogged nozzle. Chapters 3 and 4 have proposed inserting scrap objects into a partially printed model during a printing process, saving overall time and material by using these objects as substitutes for printed infill or support material. To make these approaches feasible with existing printers, the software simply pauses the machine and notifies the user to insert the object. We refer to this kind of turn-taking approach to interacting with semi-automated fabrication processes where interactions can be initiated by the system after a waiting period as *Intermittent Interaction*. In the context of personal fabrication, intermittent interaction describes periodic user interventions during an ongoing process with prolonged breaks between interactions, typically lasting tens of minutes to several hours. The interactions allow accomplishing tasks that would be impractical or impossible for an automated system to perform otherwise.

Beyond personal fabrication, many kinds of making processes share similarities in their user interactions with intermittent interaction as well. In traditional craft, makers have to wait for glue to set, for paint to dry, for tools to heat, or for casts to cool. In fact, researchers have proposed many semi-automated approaches that depend on some form of intermittent interaction for making and fabrication. For example, to improve crafted objects [3, 17, 22, 27, 30, 34, 53, 54, 80, 95, 97, 118, 125, 127], increase creative user agency [102, 117], and enable novel fabrication methods [2, 18, 70, 78, 87, 91, 104]. All these systems rely on the user to complete some part of a semi-automatic process, often over a long duration, but prior work typically does not consider the required user effort to perform these interactions. To close this gap, we report on a controlled study to investigate how people perceive the subjective cost of intermittent interactions, when they prefer the interactions to occur, and what effects varying levels of intermittent interactions have on the subjective value of the object being made.

Our study protocol asks participants to assemble LEGO models at controlled time intervals over several hours. LEGO makes it feasible to conduct a long-duration take-home study, and it makes it possible to strictly control the

complexity of each interaction (i.e. the number of LEGO bricks placed per sub-assembly step) and the *density* of interactions during the making process (i.e. the frequency and distribution of LEGO sub-assembly steps). Not only has LEGO been used before to represent other materials in production tasks [91], but assembling LEGO parts has been used directly for intermittent interaction in Chapter 4. Moreover, the goal of a semi-automated digital fabrication process (and many crafting processes) is to produce a completed tangible item, typically involving manual dexterity and precision, and requiring following a series of steps. Assembling a LEGO model also has these characteristics.

We recruited 12 participants to each assemble three different LEGO models, with the assembly steps of each model distributed over a 4-hour time frame. This was accomplished using 2 to 8 custom "lock-boxes", each containing LEGO pieces and a link to online assembly instructions. SMS notifications were sent to the participant at controlled times with a combination code to unlock the next box in sequence. By manipulating the frequency and schedule of notifications, and the complexity of LEGO assembly steps in each box, we could represent different levels of interaction density and interaction complexity. Density and complexity are design parameters identified in Chapter 4 for a semi-automated 3D printing process. Our system logs how long it took to respond to each step notification and when the step was completed. After all steps for a model were completed, we ask about mental demand, annoyance, enjoyment, and feeling of contribution. A post-study interview asked how the participant valued their completed LEGO models, their general experience for this style of intermittent manual tasks over a long period, and what kind of task schedule they preferred.

Overall, our results identify themes for intermittent interaction: the progression of making an object over time was enjoyable; higher task complexity can increase enjoyment due to greater engagement with the process; and although performing an assembly task as an infrequent sequence of steps can feel rewarding, it can also be annoying, even stressful, depending on how the steps are distributed over time and whether the overall goal was worthwhile. Using a standard methodology, we demonstrate that the "IKEA effect" of self-assembly enhancing object valuation [74] extends to partial assembly over time, even when the maker contributes only a small portion of the assembly. We also identify two main types of scheduling strategies and propose simple prediction heuristics to identify or generate desirable schedules. Finally, we contribute guidelines to consider when designing systems that use intermittent interactions.

6.1 BACKGROUND AND RELATED WORK

Our work relates to the use of AI agents, notifications, slow interactions, and the value of manual participation. We contextualize our contributions within these domains, outlining how our study augments or confirms previous findings. Finally, we examine semi-automated fabrication where our contributions can be used to justify, improve, and extend this family of approaches.

6.1.1 *AI Agents*

Turn-taking style interactions between humans and machines exist in the space of AI agents, such as querying smartphone assistants or large language model chat systems. These systems typically respond in real time, and interaction patterns differ without the need to address waiting periods. In the context of AI, the term intermittent interaction describes a fully user driven turn taking approach to interaction [15]. This differs from our definition in the context of making physical items, where prolonged processes are common and where the interaction can be initiated by the system after a waiting period, for example in the form of requesting manual assistance from the user.

6.1.2 *Notifications and Interruptions*

Our study makes extensive use of phone notifications. Prior work shows that the majority of notifications are either handled immediately, i.e., within 30 seconds, or after a long time, i.e., over eight minutes [90]. Though smaller in scale, our study confirms these findings. The related work further finds that notifications are only considered bothersome if they are deemed superfluous, such as ads, uninteresting content suggestions, or reminders for expected events. Although the notifications in our study were expected, they were typically still deemed important and addressed promptly.

6.1.3 *Slow Interactions at Home*

Technology that acts autonomously can change the behaviour of people within its vicinity. Many makers regularly approach their 3D printer to check the progress or simply to observe the process. Photobox is an interactive piece of furniture designed to reflect on the role of technology at home [75]. The box occasionally prints randomly selected photos from the user's past and stores them within itself. Most users regularly checked the box for new photos. Photobox demonstrates that users will change their routine to include technology without being prompted to if the encounters with it are meaningful. Otherwise, the same technology may become a nuisance or is simply tolerated and fades into the background. Our study also invites participants to live with technology, but it actively prompts interactions following a known schedule. Beyond changes in routine in response to technology, we observed participants structuring their day around the scheduled interactions. Similar to the interactions with Photobox, participants experienced anticipation towards coming study tasks and to see the final outcome.

The Reflexive Printer [107] also regularly prints pictures, but does not allow storing the pictures within the device itself. Handling a large number of pictures can cause clutter and was identified as a source of annoyance. Participants also handle a large number of items in our study, and we designed our study material with storage options to avoid clutter.

The Slow Game by Odom [76] studied slow, intermittent interactions outside of fabrication at home. Interactions that can be delayed for many hours without any downside lack a sense of urgency. That does not mean, however, that they are unimportant. There was no fixed deadline to complete our study, so participants could freely delay tasks. Tasks were typically still considered to be both important and urgent, as delays add up between tasks and participants aimed to complete tasks within self selected time frames, such as before the end of the day. Odom describes intersection (rather than interactions) with technology. In the case of slow fabrication machines, intersections could account for glancing at the machine to detect errors that the machine misses, or simply watching the machine's steady work as a mindful distraction. Our study aims to determine whether prompting interactions alienates users. Cumulative intersections, or "ensembles" as described by Odom, could occur for shared fabrication machines in maker spaces. They may materialize as a form of unseen, crowd-sourced failure recovery. Given sufficiently simple pending user interactions, even passers-by may be able to perform tasks similar to low complexity tasks in our study and assist in turn-taking fabrication processes. The interactions with Slow Game are expected, but don't necessarily align with the users' available time. In our study, notifications were similarly expected, but users had limited control over their timing, which allows us to investigate strategies and challenges in handling slow schedules. Slow Game was designed as "[...] a bridge between intersections and rich, yet highly minimal interactions." Our study aims to determine whether minimal interactions with making processes can achieve similarly rich connections with the made item.

6.1.4 *Labour leads to love*

The IKEA effect [74] is the idea that assembly effort increases the perceived value of an item, and that this increased valuation only persists as long as those items remain complete, that is, have not yet been disassembled again. The study determined a 63 % increased valuation when asking participants that had assembled a piece of furniture to rate and bid on the piece, compared to participants that had only inspected the same furniture before bidding. We study a similar effect, but in a different context and scale. In particular, in our study, the finished models are always complete, but participants only performed a part of the assembly. Similar to the IKEA effect, we perform an incentive compatible value elicitation method where the dominant strategy is bidding the true value associated with the assembled model. We modify the procedure to avoid the need for participants to carry currency to participate.

6.1.5 *Guiding Assembly Processes*

Assembly processes can be error prone and need to be carefully guided to limit the chance of failure. Mobile Fabrication [89] envisions and prototypes portable personal fabrication devices, such as extruding a handheld 3D print-

ing pen directly onto a smartphone that displays model outlines. Similar to Mobile Fabrication [89], we provide visual instructions via smartphone that the user has to match to perform the manual construction parts. Our instructions closely match official LEGO instructions, but deviate in the number of parts per step.

Being the machine [21] also puts the user in the place of the physical actuator of a 3D printer. The system visualizes machine code through a laser pointer, fully relinquishing the physical crafting process to the user. Similar to intermittent interaction, Being the machine relies on human assistance to complete the crafting process, but the system furthermore abandons all automation options.

Exquisite Fabrication [87] and smARt.Assembly [91] guide users to assemble pieces from multiple sources. The smARt.Assembly system uses LEGO bricks as their assembly medium to mimic industrial assembly stations where parts or bricks are assembled one at a time. We also use LEGO bricks to mimic tasks that require some manual dexterity, but deliberately combine multiple assembly steps into one instruction image to adjust the interaction complexity. The Exquisite Fabrication system allows users to coordinate and combine the outputs of multiple fabrication machines. Such a setup requires turn-taking interactions to orchestrate the different machines, demonstrating one of the types of use cases we investigate.

6.1.6 *Semi-Automated Fabrication*

Semi-automated fabrication processes require some form of user interaction to succeed. This broad category spans from interactive fabrication, which requires the user's constant attention, to merely resetting the system at the end of batch processes to enable repeated execution.

Mueller situates how the speed of fabrication allows different kinds of user interaction, from hands-off automated processes, to turn-taking, to direct manipulation [66]. Increasing the speed and interactivity of personal fabrication has been framed as one of the key challenges of the technology [12, 13]. Intermittent user interactions can speed up the fabrication processes that are traditionally fully automated, for example by replacing internal or external support structures with user-placed objects (see Chapters 3 and 4). The replaced structures do not need to be fabricated, which can significantly decrease fabrication time. Substiports requires the assembly of LEGO pieces during fabrication, which closely matches the interactions in our study. User involvement can also allow for more expressive, faster design iteration, as demonstrated by ReForm [117] and 3D Pen + 3D Printer [102]. These systems allow design adjustments by manipulating clay prototypes, or by manually adding elements with a 3D printing pen. Similarly, Encore [18] and Patching Physical Objects [104] allow adjusting existing objects, instead of recreating them entirely, by involving manual interactions. These examples show that this kind of interaction can improve existing processes, open up new fabrication options and enable more creative control.

Involving the user during fabrication can provide further benefits by materially improving the fabricated object. For example, including objects that provide mechanical functionality with Medley [17], or creating custom fits for existing objects with Fitter [3], FusePrint [127] or MixFab [118]. When fabricating fluidic systems, manually incorporating fluids [53, 54], stir bars [95], micro channels [30] or membranes [80] can enhance object functionality. Similarly, Printy3D [125], HotFlex [34] and further systems [22, 27, 97] incorporate electronic components and sensing devices to enhance the functionality of fabricated objects significantly. These and other systems that involve turn-taking user input make use of a limited number of intermittent user interactions. Our study provides in-depth insights into user experiences in the space between fully automated and fully user driven systems.

Systems that enable rapid prototyping are often fully driven by the user, but may still incorporate autonomous fabrication machines, and may involve waiting periods to allow the machines to catch up to the latest user input. Examples of such systems are On-The-Fly Print [79], RoMA [78], Interactive Construction [68], Turn-by-Wire [105], Protopiper [2] and Fabrickation [69]. The shortest wait times included in our study are 3 minutes long, which may still be relevant for such systems. For example, FormFab [70] allows shaping acrylic plates into 3D models by applying either a vacuum or positive pressure to a heated section of the plate. After designating an area that the user intends to reform, they have to wait for a few minutes while the system heats up the area, before they can take direct control of the shaping process.

Our study aims to provide empirical data for a wide variety of systems that use intermittent interactions by distinguishing interactions based on their complexity and rate of occurrence. We identify insights and guidelines that help design turn-taking systems, particularly in the context of fabrication.

6.2 STUDY

This study examines how the density of interactions with extended waiting periods, and the complexity of each interaction, can affect user experience, perceived value of an assembled item, and the effort required to complete the assembly. We also observe trends for what participants prefer when selecting a schedule of interactions distributed over a long duration.

Participants were asked to assemble three LEGO models in three different 4-hour "assembly sessions", each scheduled with 2, 4, or 8 user interactions. This represents low to high *interaction density*. To draw an analogy with 3D printing, starting a session represents starting a print, and the final user interaction represents retrieving and cleaning up the finished print. The final interaction is always at the end, so a schedule of 2 interactions means a single interaction somewhere between the start and end of the session. During each interaction, the participants had to place 1, 4, or 12 bricks (or sub-assemblies of bricks) representing low to high *interaction complexity*.

This study took place over multiple days[1], so we had to create a portable, low-cost system for participants to take to their home or workplace. We used locked and labelled wooden boxes, each containing LEGO pieces and a QR code. SMS notifications with a box ID and lock combination were sent at controlled intervals to prompt an interaction. The QR code inside each box linked to a web page with assembly instructions for the LEGO pieces in the box. Our experiment software recorded when the instructions were retrieved after sending the SMS and when the task was marked as completed on the web page. The participant also had to take a photo of their partially assembled model so we could log when they finished the interaction.

We use a LEGO model assembly task for reasons provided earlier: it makes the study feasible and controllable, and building a LEGO model shares many characteristics of semi-automated digital fabrication processes (and we have used LEGO directly in some Chapter 4). Like a semi-automated 3D printing process, the session progress is paused while waiting for the participant to complete an interaction task. This matches the dependent sequential nature of an intermittent interaction 3D printing process where the printer is paused until the user completes an interaction like changing the spool or inserting a scrap object [115, 116].

6.2.1 *Participants*

We recruited 12 participants without any specific requirements for recruitment. Eight self-reported their gender as men and four as women. One of the men self-reported as transgender, and all others were cisgender. Six participants were between 18-24 years old, and six were between 25-34 years old. We asked about their occupation and to describe and rate their crafting and fabrication experience. Our participants had diverse levels and types of experience in fabrication and crafting, from none to expert for various manual tools and digital fabrication machines. We also asked participants whether they practice sustainability effort and found that the set of participants was also very diverse in this regard, ranging from taking no sustainable action to performing extensive daily effort to act sustainably. Six participants were graduate students, five were undergraduate students, and one was a working professional. Recruitment was executed through email and word-of-mouth. Participants were remunerated with $30 and one of the assembled LEGO models.

6.2.2 *Apparatus*

To control the timing and availability of LEGO pieces, we employ an array of individually locked boxes (Figure 6.1a). Each box is secured with a combination lock and labelled appropriately. Additional boxes contain a reminder letter and replacement pieces. Each individual box is large enough to store

1 The extended duration of this study protocol was approved by our institution's Research Ethics Board.

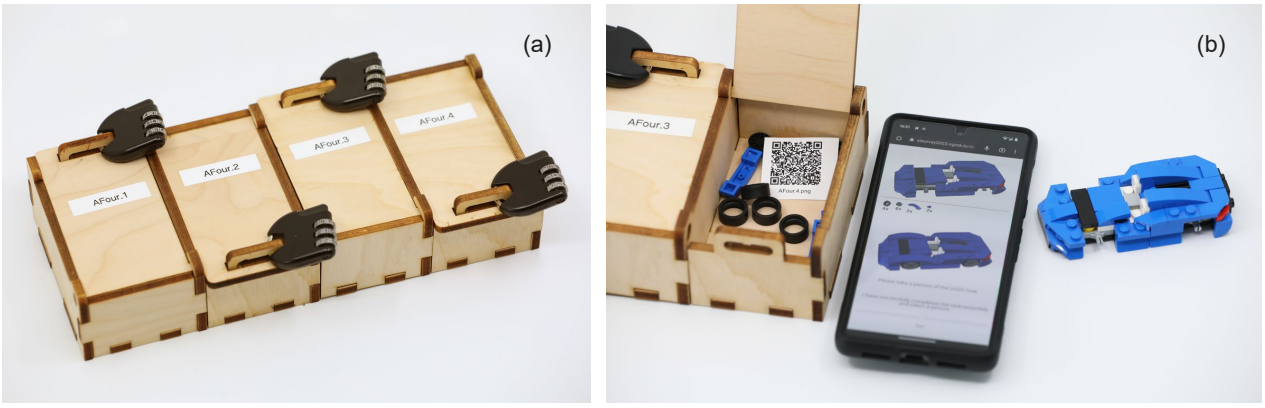


Figure 6.1: Apparatus: (a) example of individually locked boxes, in this case the 4 boxes are used to control 4 interactions during a session with this density level 4; (b) an unlocked box during an interaction task. The QR code leads to a logged webpage with instructions to assemble part of the model using the pieces inside the box.

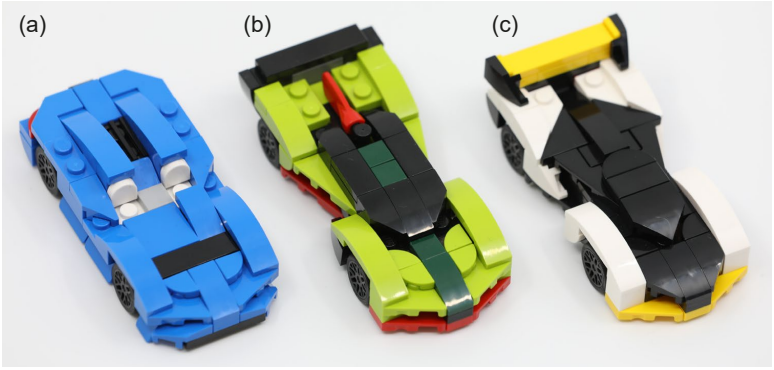


Figure 6.2: LEGO car MODELS used in study, referred to as: (a) blue; (b) green, and (c) black & white.

locks and intermediate assemblies to avoid clutter during the study. A server tracks participant progress throughout the study. The server monitors participant access to web pages to retrieve instructions or once they confirm task completion on the same page after each interaction task, and it sends SMS notifications according to the controlled schedule.

There were three different LEGO models, all race cars, one *green*, one *blue*, and one *black & white* (Figure 6.2). They have between 85 and 96 pieces and each can be assembled within 15 minutes. The maximum number of pieces in a model informed the maximum number of pieces used per assembly session. For some conditions, parts of the models were pre-assembled to control the number and complexity of steps in an interaction.

6.2.3 *Procedure*

The procedure had three parts: initial setup and instructions, completing the three sessions of assembly tasks, and a final interview.

6.2.3.1 *Initial Setup and Instructions*

For the initial setup, participants visited a university lab to pick up the the study boxes. During the meeting, they were also briefed in detail about what to expect during the study, and received training how to use provided QR codes; how to open the number locks; how to log in to the study server with a password; how to interpret and select a notification schedule for a session; and how to change the mobile phone number used during the study. A reminder sheet with relevant information was placed into a study box. A researcher emphasized that it is up to the participant when to start each session; that each session is scheduled to last 4 hours, but that it is fine to delay tasks or to leave the location of the study boxes at their discretion; that they have to complete three sessions to complete the study; and that they can contact the researcher anytime if they have questions. The researcher then noted participant demographics and asked the participant to set up the study boxes at home or at work, and to leave the boxes there until the final interview.

6.2.3.2 *Assembly Task Sessions*

Each participant completed three 4-hour assembly task sessions. Most did one session per day over 1 to 4 days. Four hours was chosen to represent the duration of moderately sized 3D prints. When the participant was ready to start an assembly session, they used a QR code to log into the server and select a schedule. They were given a choice of four pseudo-random notification schedules generated according to the experimentally controlled number of interactions during the session and three conditions for notification distribution: one notification must occur at the very end of the session; the other notifications must not occur within the first 10 minutes or last 10 minutes of the session; and there must be a minimum period of 3 minutes between any two notifications. Enforcing minimum periods between notifications approximates turn-taking with semi-automated machines like 3D printers, and 10 minute boundaries at start and end approximates startup and cool down times of 3D printers.

Once the session started, participants received SMS notifications according to their chosen schedule. Each notification provided a lock code and box label to open a specific box. The box contained LEGO pieces and a QR code that the participant scanned to open a web page with assembly instructions (Figure 6.3). The number of LEGO pieces to assemble determines the complexity of the task. The web page also provided a form for the participant to confirm they completed the assembly task for the current interaction. In addition, the participant was instructed to take a picture of the model after completing each interaction. Picture time stamps were examined to ensure that the assembly task was performed when indicated by the web page form. This verified all participants followed the instructions and the assembly schedule. After confirming the interaction task was completed, the number of minutes until the next scheduled interaction notification was displayed.

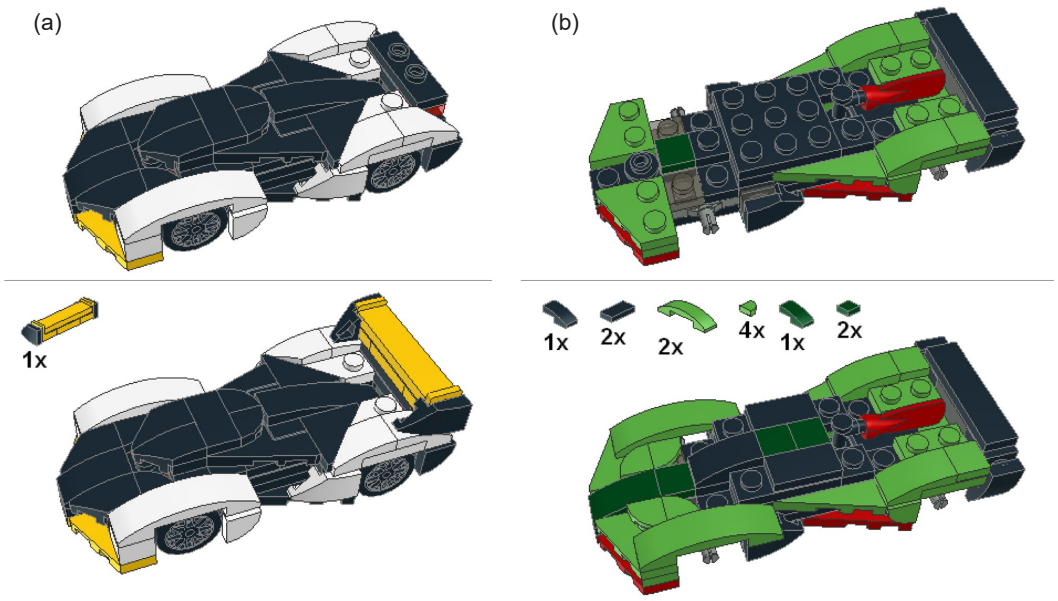


Figure 6.3: Example task instructions showing current model state at top, and the target state with list of pieces at bottom: (a) a low complexity task adding a single sub-assembly; (b) a high complexity task with 12 individual pieces.

This process repeats until the last interaction task is completed with the submission of the web page form. After completing the last task in a session, the participant answered a post-session questionnaire with four subjective measures: mental demand, annoyance, enjoyment, and sense of contribution. Each had seven options presented equidistant on a number line with Likert-type labels ranging from "Strongly Disagree" to "Strongly Agree", with a "Neutral" option in the middle. After submitting the post-session questionnaire, the next session was unlocked for the participant to begin when they wished.

6.2.3.3 *Final Interview*

Post study interviews asked participants about their general experience, their scheduling preferences, and their perceived value of each LEGO model. The latter used a standard method variant [14]. In this variant, the participant is initially gifted the item, allowing them to participate in the following auction without carrying currency. First, the participant is told they can chose one assembled model as a gift to keep. Then, the researcher offers to "buy back" the chosen model in an auction. After the auction process was explained, the participant was asked to value their chosen model in dollars. This was noted as the model worth, which can be higher than market value. The researcher then rolled a 10-sided die: if the result was equal or higher than the model worth, the researcher bought the model back from the participant for the model worth. Regardless of the outcome of the auction, the noted model worth was used to assess their perceived value of the model. After establishing

this baseline, participants were then asked what the remaining two models would be worth to them.

6.2.4 *Design*

This is a mixed design study with two primary independent variables: interaction DENSITY (as 2, 4, or 8 interactions during a 4-hour session) and interaction COMPLEXITY (as 1, 4, or 12 pieces of LEGO per interaction task). In the within-subjects portion of the design, each participant experienced all levels of both DENSITY and COMPLEXITY variables, creating a total of three sessions per participant. During each session, a participant assembles a LEGO model with a total of DENSITY x COMPLEXITY pieces (or sub-assemblies of pieces). In the between-subjects portion of the design, all possible combinations of the DENSITY and COMPLEXITY were explored between three participants. Combinations of the two variables were counter-balanced using a 3 x 3 Latin square (Table 6.1).

Table 6.1: Within and between participants design. The left Latin square counterbalances combinations of DENSITY and COMPLEXITY. The right Latin square is flipped and used to ensure each MODEL type is assigned exactly once per level of DENSITY, COMPLEXITY and per participant.

COMPLEXITY in pieces \ DENSITY in interactions	2	4	8
1	P1	P2	P3
4	P3	P1	P2
12	P2	P3	P1

COMPLEXITY in pieces \ DENSITY in interactions	2	4	8
1	Blue	Green	B&W
4	Green	B&W	Blue
12	B&W	Blue	Green

A secondary variable, MODEL, was introduced through the three different LEGO models. A second flipped Latin square was used to counterbalance MODEL type. This ensures that every participant assembles each of the three different MODEL types, and each model is assembled once with each level of DENSITY and COMPLEXITY between three participants. Using a different model for each assembly session reduces learning effects between sessions. It also maintains the purpose of creating a new item during each session, rather than repeating the assembly of a known item. The inclusion of different models helps participants differentiate the sessions based on unique visual memories of assembling the distinct models. Finally, the inclusion of different models allows gathering additional qualitative insights as participants can clearly delineate between model preferences and process preferences.

Four subjective measures are created by the responses to the post-session questionnaire, each are 7-level Likert-type questions:

mental demand "The set of tasks was mentally demanding"

annoyance "I was annoyed by the tasks"

enjoyment "I enjoyed the tasks"

contribution "I feel like I contributed to the result"

Three objective measures were computed from timestamps in server logs of user interactions:

delay time The interval from SMS notification until the QR code is scanned to view instructions.

assembly time The interval from instruction webpage access until task completion form was submitted.

session time The total duration from session start to completing the last task.

In summary, each participant completed one session with 8 interactions, one session with 4 interactions, and one session with 2 interactions, creating 14 data points for *Assembly Time* and *Delay Time* (168 data points across all 12 participants). In addition, each participant generated 1 data point for *Session Time* and 4 data points for the subjective measures for each of the 3 sessions they completed.

6.3 RESULTS

In this section we present results for objective timing measures from logs and subjective measures from the post-session questionnaires. Emergent themes from the post study interviews, an examination of scheduling preferences, and perceived value of the final models are presented in a following section.

6.3.1 *Mental Demand, Annoyance, Enjoyment, and Contribution*

Complex tasks are more mentally demanding, but they also increase the feeling of contributing to the result (Figure 6.4b,h). Higher task complexity also increases enjoyment (Figure 6.4f). An analysis of variance based on mixed ordinal logistic regression found effects of COMPLEXITY on *Mental Demand* ($\chi^2(2, N=12) = 21.05$, $p < .001$), on *Contribution* ($\chi^2(2, N=12) = 9.27$, $p < .01$) and on *Enjoyment* ($\chi^2(2, N=12) = 6.02$, $p < .05$). Pairwise comparisons using Tukey's HSD test indicated that placing 1 piece provided a lower sense of *Contribution* compared to 12 pieces: ($Z = 2.88$, $p < .05$). Pairwise comparisons also indicated that placing 4 pieces required less *mental demand* than 12 pieces ($Z = 2.36$, $p < .05$). The original LEGO instructions of our test models are simple and intended for children 6 years and older. The 4 pieces condition closely matched the original instructions. In contrast, the 12 pieces condition far exceeded the number of pieces used per instruction step and illustration.

A high number of interruptions can cause users to feel annoyed. Figure 6.4c suggests 8 interruptions was more annoying than 4 or 2 interruptions and three participants noted that 8 notifications was more annoying (P5, P11, P12).

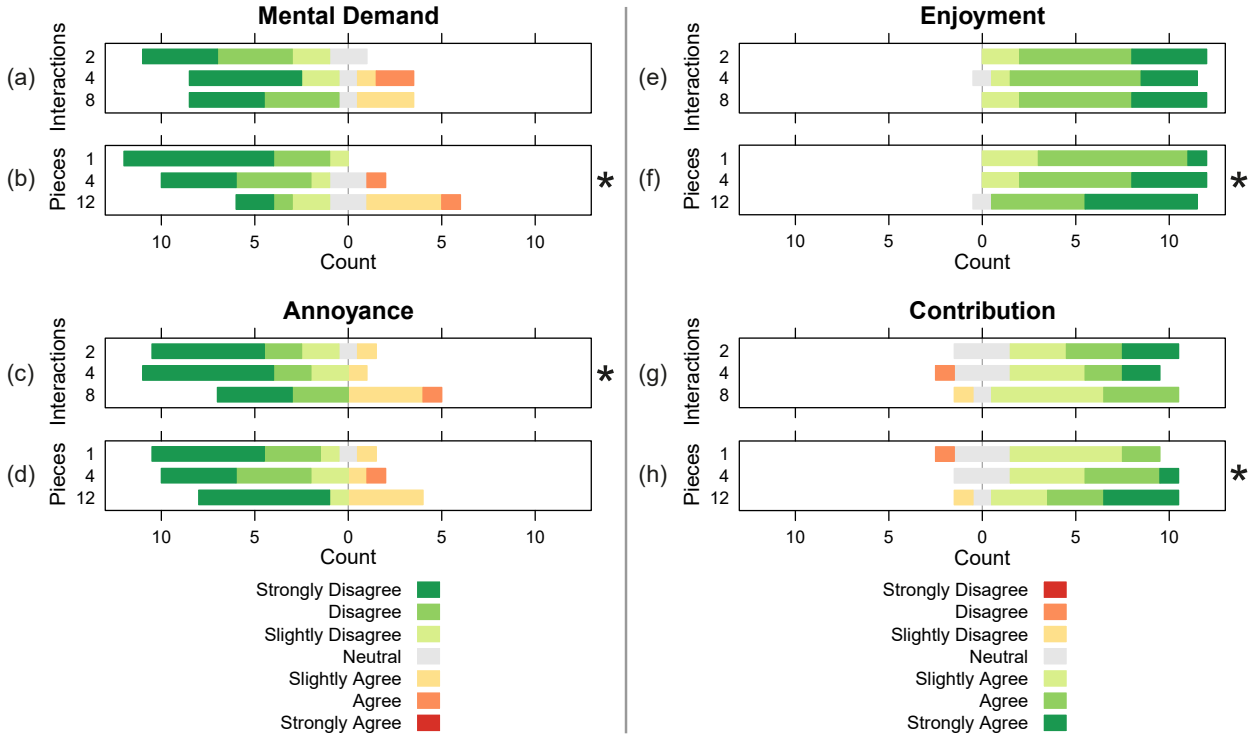


Figure 6.4: *Mental Demand*, *Annoyance*, *Enjoyment* and *Contribution* by interaction DENSITY in interactions, and by interaction COMPLEXITY in number of pieces. Plots with significant effect in ordinal regression are marked with *. Questions for (a - d) are negative assertions, and (e - h) are positive assertions.

Analysis indicated a statistically significant effect of DENSITY on *Annoyance* ($\chi^2(2, N=12) = 6.84$, $p < .05$).

6.3.2 *Delay Time, Session Time, and Assembly Time*

The majority of notifications were either handled immediately, or after a significant delay (Figure 6.5). This overall trend confirms results from prior work, but the distribution of *Delay Time* in our study differs from prior work in three ways [90]. In our data, no notifications were addressed within 30 seconds. Unlocking the study box, scanning the QR code, and potentially logging into the study server takes time. The shortest recorded delay time was 46 seconds. Participants generally took longer to address phone notifications compared to findings in prior work. Delay times for handling notifications were elevated for time consuming tasks, as participants needed to wait until they have an opportunity to address the notification properly. The peak of notifications handled immediately is flattened. This is likely due to variance in the time it takes to access the study box. The overall time to address notifications is increased. On average, participants took 44.6 minutes to start the study tasks (median 2.9 minutes, min 46 seconds, max 17.8 hours, sd 171.3 minutes). The median delay time per step in a session remains consistent as the session

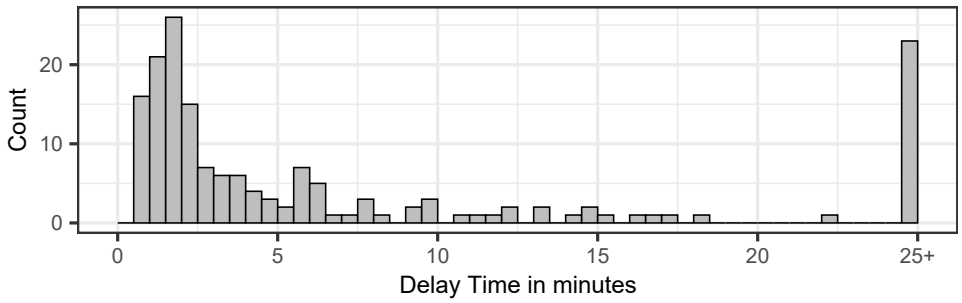


Figure 6.5: Histogram of the distribution of the *Delay Time* participants took to address each notification.

progresses (Figure 6.6). User responsiveness to making tasks does not decrease with repeated interruptions for at least up to eight interactions.

Each schedule spans a 4 hour period, so the theoretical minimum *Session Time* time to complete a session is 4 hours. The time to complete a session increases with interaction density (Figure 6.7). The median session time increases slightly, since every interaction takes time to complete. The effect on mean session time and standard deviation are much more pronounced. In particular, the overall session time is dominated by long pauses, for example, when the participant goes to sleep. The average time to complete a session was 7.7 hours (median 4.7 hours, min 4.1 hours, max 40.7 hours, sd 7.5 hours). The increase in mean session time instead is likely due to a higher likelihood to include long pauses as the number of pauses increases.

The average *Assembly Time* varied between participants (Figure 6.8). P10 reported spending a lot of time attempting to fix assembly errors during one session. P2, P5, P8 and P11 participated in the most labour intensive combination of conditions with 8 interactions and 12 pieces per box. P11 noted assembling LEGO for relaxation. The average assembly time across all participants was 3.0 minutes (median 1.5 minutes, min 4 seconds, max 46.4 minutes, sd 4.7).

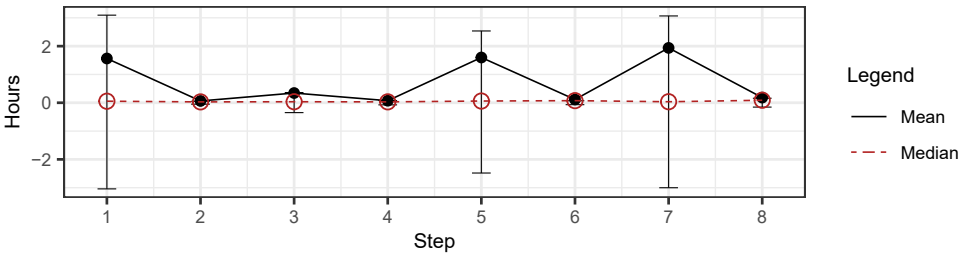


Figure 6.6: *Delay Time* by step number for DENSITY of 8 interruptions.

6.3.3 *Model Value*

Overall, the perceived value of assembled models was dramatically higher than the value of the model kit itself. The perceived value of a model was indicated to be as high as $40. In six cases, even the lowest valued of the three models was still valued higher than their retail price of $4.99. Models were valued on average $9.31 (median $8.25, min $1, max $40, sd $7.58). The median perceived value is 65 % above retail price, which closely matches the 63 % increase in valuation determined in the "IKEA effect" [74]. Eight participants chose to keep the green model. The blue and black & white model were each chosen twice. Six participants chose to keep the model they spent the most effort building. Nine participants chose the model to keep based on a preference for the model itself.

6.4 CODES FROM PARTICIPANT INTERVIEWS

We analyzed audio transcripts and notes from participant interviews, which lasted on average 45 minutes, using techniques from Thematic Analysis [16, 65]. We first identified meaningful statements made during the interviews and clustered related statements together. During the analysis, participant data was added using constant comparison while focusing on similar and conflicting experiences and a priori questions (e.g., experience with fabrication and location of study). We identified and developed five main codes by looking at the structure of mapped thematic clusters. We present and describe codes organized by topic below.

6.4.1 *Enjoyment Through Engagement*

All participants enjoyed the study and assembling LEGO generally. Various sources of enjoyment participants experienced are directly related to their level of engagement with the tasks.

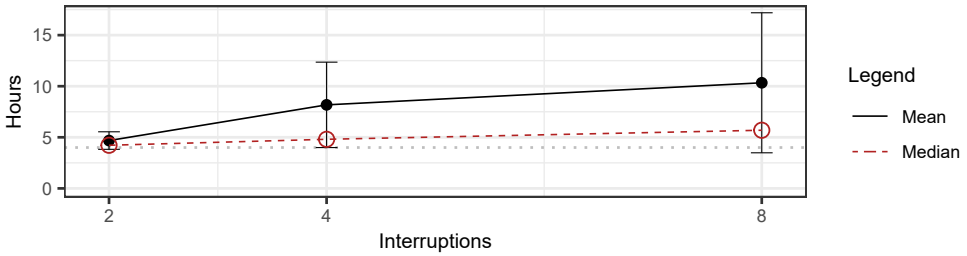


Figure 6.7: *Session Time* by interaction DENSITY. Error bars show 95% confidence intervals. The dotted grey line shows the minimum session time without any user interaction.

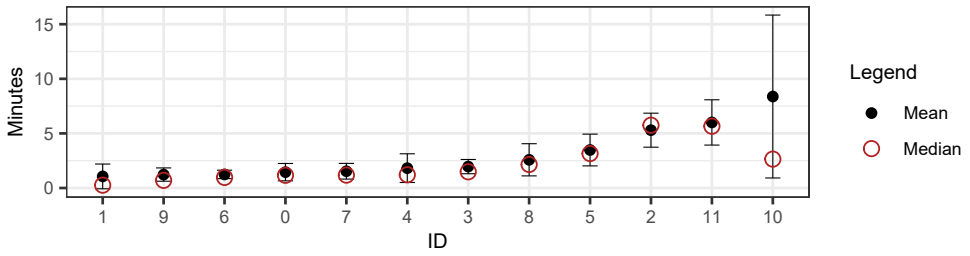


Figure 6.8: Mean *Assembly Time* of tasks per participant.

Challenge fosters engagement

Participants overwhelmingly enjoyed the complexity that came with assembling more pieces (P3, P4, P6-11), describing them as interesting challenges (P1), puzzles (P5, P12), and overall as the most enjoyable. P5 called the tasks a "*3D manipulation challenge.*" One participant was even adamant that they had performed only half as many of the most complex tasks than they actually had done, suggesting time passing quickly due to elevated engagement levels (P3), and they preferred identically coloured pieces and asymmetric models for increased assembly difficulty.

Overcoming challenges bolsters accomplishment

Being fully engaged in a challenge allows feelings of accomplishment after completing them (P10, P11). Two participants felt accomplished for correcting initial mistakes (P4, P9) and one participant described completing the complex instructions as "*defeating a Dark Souls boss*" (P7).

Manual tasks evoke engagement

Participants appreciated the tactile experience, stating it engaged a different part of the brain than passive activities like watching videos (P1, P4, P5). Participants also found tangible satisfaction in putting on the pliable tires or continuing to play with the completed models (P6, P8, P9, P11). Two participants however struggled with handling small building blocks (P2, P7). P7 described the tasks as "*tinkering with small pieces, with big hands.*"

Partitioned tasks enable offline engagement

Spreading the assembly process out over a longer period of time allowed participants to think about the tasks while not actively engaged in the assembly. Three participants noted enjoying the anticipation and trying to predict the next steps (P9-11).

6.4.2 *Enjoyment Through Progression*

Another source for enjoyment throughout the study was experiencing progress when completing an assembly step. While partitioned assembly heightened this sensation, it also delays the most gratifying part of completing the last step.

Creation tasks tangibly track progress

Since the assembly is a physical item, the state of the item provides a tangible progress record. It is enjoyable to see an assembly come together and to observe the impact each step makes (P1, P7). Participants especially appreciated the final steps (P2, P5, P6) and likened them to "*cutting the ribbon*" or "*putting the cherry on top*" (P4).

Partitioning delays gratification

Given that the last step is the most enjoyable, higher numbers of steps can feel like a delay in getting to the finish line (P4, P11). One participant even tried to 'cheat' the system by guessing the code for the very last box (P8).

6.4.3 *Task Partitioning is Loved and Hated*

Participants describe task partitioning as helpful and following the process can feel rewarding. However, excessive partitioning into too small sub tasks was the main source of annoyance during the study.

Partitioning processes promotes completion

Five participants found the segmented approach helped them finish the task, as attempting to tackle the entire task at once may lead to abandoning the task partway or being hesitant to start at all (P7-10, P12). Due to the smaller sub tasks, one participant was able to fit the assembly into a tight schedule (P9). A participant said that the study setup was "*more engaging, more active and more accountable than a to-do list.*" (P7). Partitioning was also noted to help handle task complexity, since it enforced a systematic approach and provided clarity where to start (P7, P8, P10, P12). A participant estimated that performing the same assembly in one go would take three times as long (P7).

Notification timing matters

Three participants noted that receiving multiple notifications within a short period of time can be irritating (P5, P11, P12). On the other hand, long periods between notifications can feel "*like a slog*" (P4). Waiting is harder when not otherwise occupied. Only participants who performed the study at home, rather than at work, lamented long waiting times (P3, P4, P6, P10). Participants who noted a preference described 2, 3 or 4 notifications during a

4 hour making process as ideal (P5, P6, P9, P10), as fewer interactions create pauses that are too long or offer "*too little involvement*" (P9).

Every interruption should be worth it

Two participants complained that for tasks with minimal complexity, waiting and being asked to interact was "*not worth it*" (P1, P10). Two participants suggested that knowing the purpose of the task can provide additional motivation (P1, P7). Notifications for more engaging tasks were met with excitement (P10, P11), and were described as receiving a "*dopamine rush*" (P9). The notifications in our study mostly avoided being seen as annoying as participants felt they served a purpose and were not repetitive (P1, P4-6, P10, P11).

Ongoing processes can be stressors

Some participants felt a strong sentiment to complete each step as soon as possible to avoid compounding delays (P1, P2, P12). A large number of interactions can increase the perceived urgency of the tasks, which can lead to stress. Due to travel plans, a participant felt the need to "*drop everything*" to respond immediately and avoid further extending the session (P5). Another participant reported taking assembly pieces to the bathroom to save time (P6). Participants were largely unconcerned with assembly errors, unless they blocked further progress. Our study setup did not offer the option to restart a session. One participant felt stressed when they were forced to continue the assembly in an error state that they failed to fix (P10).

Partitioned tasks mean having to follow a schedule

Some users rely on mental math to follow a schedule (P6, P9, P10), sometimes exclusively when notifications are muted (P12). Forgetting about tasks can be annoying and can cause delays. Very long breaks make forgetting more likely (P12), and two participants noted that 3 hour long pauses between interactions were too long to remember (P9, P10). Other participants praised that the phone notifications allowed offloading the mental effort of following a schedule (P7, P8).

6.4.4 *Participants had Opposing Strategies for Picking Schedules*

The majority of participants were satisfied with their chosen schedules and felt four options were adequate. One participant desired additional options (P9), but many options can lead to choice paralysis. Three participants avoided having to pick a schedule entirely by selecting a random schedule, or simply the first schedule listed (P1, P2, P8). They still discussed strategies they would consider using. Participants mainly relied on two opposing strategies while adjusting their approach to schedule around fixed events.

Clustering reduces perceived effort

A majority of participants preferred clustering tasks to minimize mental burden of context switching. Six participants specifically preferred clustering tasks at the start of the session, for example to not forget about the whole process and to quickly get a sense of what they are building.

Spreading interruptions can be useful

Five participants preferred the directly opposing approach of spreading interruptions evenly. They found short pauses distracting and preferred longer 1-hour gaps (P2) or noted 45 minutes being a more "*workable*" pause length (P12). One participant appreciated the uniformity of "*good, logical breaks*" that a uniform distribution provided, but also favoured clustering 2-3 tasks within 20 minutes (P5).

Event scheduling overrules other strategies

Accommodating specific events, primarily lunch, was noted repeatedly as an exception to other strategies. When applicable, event scheduling was always the top priority (P1, P2, P12).

6.4.5 *Participation can Increase Desirability and Perceived Value*

Three participants explicitly stated that a model's worth was proportional to the time spent building it (P2, P11, P12) and participants typically value the most desirable model the most. Monetary value is however not directly proportional to desirability. Two participants wanted to keep a different model than the one they valued most (P4, P10). In one case, the model was valued low due to assembly errors, but the participant was fully engaged in trying to overcome the assembly challenge and desired that model to try again after the study (P10). For the second case, the participant desired the model as a fan of the car that the chosen model was based on, but they had barely participated in the model's assembly (P4).

Participation threshold for ownership

Six participants noted a lack of ownership in conditions with low involvement (P4, P5, P7-9, P11). They described the process as "*repairing*" rather than "*building*" when placing only one piece at a time (P5), or as "*playing with someone else's toy*" (P11). A participant noted that they would pay more for a kit than for a pre-assembled model they "*didn't build*" (P8). Although partial assembly of the models generally increased their valuation by participants, the noted lack of ownership by some participants indicates the existence of a minimum threshold of participation to trigger the IKEA effect.

6.5 DISCUSSION

We discuss our results, thematic codes, and what they imply for intermittent user interactions in the context of making tasks below. We propose guidelines for systems with matching usage patterns. Additionally, we propose a method to predict or generate preferred schedules heuristically.

Participants generally felt they contributed to the outcome, except for some minimal complexity tasks for some participants. Perceived model value was generally strongly elevated over the market value. Our results demonstrate that the IKEA effect applies even for partial assembly of items over time above a minimal interaction threshold. Our qualitative results also show that the effect scales with engagement.

6.5.1 *Guidelines*

Based on our findings, we propose the following guidelines for designing systems with intermittent interaction.

Minimize annoyance, not effort

Fewer, more complex interruptions are both more rewarding and less annoying. Avoid repetition, like performing the same operation twice for a symmetric process. Minimal user interactions on the scale of placing 4 LEGO pieces may be insufficient to instill a sense of ownership. Avoid using more than 4 interruptions per project, or more generally, using more than one notification per hour of process duration.

Allocate complexity

Handling highly complex tasks feels impactful and engaging. Interactions that are too simple feel like wasted effort. Provide options to set an upper limit on interaction complexity, or to adjust complexity according to user preference, to avoid errors that lead to frustration.

Manage time expectations

Only inform users of upcoming tasks if they are imminent or urgent. Waiting periods under 15 minutes are typically considered acceptable. To mitigate the sense of prolonged waiting for tasks hours away, consider only informing the user when the task becomes available or 15 minutes prior.

Design purposeful notifications

Clearly articulate the purpose and the value of user interactions. A sense of contribution requires knowledge of what the user is working towards, or what their efforts accomplish. Consider including information such as material savings in communications with the user.

6.5.2 *Formalized Schedule Preferences*

Scheduling strategies are personal and varied. We formalize the primary objectives leading to the scheduling strategies participants used in Table 6.2 and provide a heuristic approach to support them.

Effort Minimization	
Task Clustering	Participants preferred grouping notifications closely together to minimize the cognitive load from disruptive context-switching.
Edge Clustering	Interruptions anchored around forced start and end points reduce mid-process disruptions.
Front-Loading	Clustering at the start preserves engagement momentum while in the task-oriented mindset of starting the process.
Extended Pauses	Clustering interruptions around the edges of the time frame incidentally schedules long, uninterrupted periods that allow accomplishing more demanding work.
Expectation Management	
Uniform Distribution	A consistent spread of interruptions allows for better anticipation and planning around them.
Pomodoro Alignment	Interruptions can be used to implement a 'Pomodoro'-style work method, breaking up work into focused intervals.
Event Scheduling	
Lunch-Break Scheduling	Users prioritize extended pauses during traditional break times, like lunch, to avoid disruptions during those periods.
Events are Contextual	Scheduling around events required contextual information, which can be entered manually, or for example by incorporating calendar information.
Mental Overhead Minimization	
Random Selection	Users may choose random or first-available schedules to eliminate the cognitive burden of planning.
Avoiding Overthinking	A quick selection allows users to commence tasks more swiftly and negates the need for detailed forethought.

Table 6.2: Scheduling objectives for planning brief intermittent interactions

Participants typically focus on either minimizing effort, or on making interruptions easier to manage and work around them by spreading them evenly throughout the process duration. Scheduling around events supersedes other strategies where necessary, but requires additional contextual information.

6.5.3 *Predicting Preferred Schedules*

We built heuristics that approximate two dominant scheduling strategies by encoding the rules we found in our interview analysis. These can be used to evaluate or generate schedules that can predict user selections well. A binary

preference value is required to choose which heuristic to use, which can be set simply by letting users choose between two indicative sample schedules.

One fourth of participants reported selecting random schedules to avoid having to make scheduling decisions. To support this user group, we recommend providing fully automated schedules based on preference settings. Other users can also benefit from strong default suggestions, for example by simply accepting a suggested schedule instead of searching through a list of options.

To evaluate the accuracy of scheduling predictions, we count the number of times the heuristic function selection matches participant selection. A match describes selecting the same one out of four options that were presented at session start. The 9 data points of participants that selected randomly were removed as outliers in this evaluation. After selecting a user preference to use or avoid clustering based on interview transcripts, our suggested approach matches user selection in 21/27 cases (77.7 % accuracy), compared to 40.0 % accuracy when selecting the better out of two functions that select randomly).

Our approach uses *cluster early* and *avoid short pauses* heuristics to evaluate schedules. The *cluster early* option counts and maximizes the number of interruptions that occur within 15 minutes after session start and after one another. The shortest total time length of this early cluster is used as a tie breaker. If there are no pauses that are as short as 15 minutes, the shortest pause length within the schedule is used instead of 15 minutes. The *avoid short pauses* option counts and minimizes the number of pauses that are shorter than the expected pause length if interruptions were distributed uniformly. A measure of how much shorter these pauses are than expected is used as a tie breaker.

Offering additional preference options, such as avoiding interactions during lunch time, may further increase prediction accuracy and allow systems that make use of intermittent interactions to better lift the burden of choosing a schedule off of the user.

6.6 CONCLUSION

We investigated how people perceive intermittent interactions with manual assembly tasks over long durations through a multi-day study. LEGO model assembly was used to represent more general kinds of interactions, and the context of 3D printing informed the design of the study and user interactions. Placing LEGO blocks requires some manual dexterity and high precision without being physically straining. These qualities match many types of interactions with digital fabrication systems and intermittent interactions within that space. However, our findings may not be applicable for certain interactions, for example tasks that are highly physically demanding. On the other hand, many tasks outside of the immediate context of digital fabrication, such as household tasks and handling appliances have a similar level of physical demand, and our findings may extend to other domains. Future work could explore interactions based on tasks of other turn-taking

scenarios like traditional crafts and arts or revising documents, or shorter times frames that better match other types of digital fabrication, like laser cutting.

Our results provide evidence that intermittent interaction is generally acceptable for long duration making tasks such as personal fabrication with a semi-automated machine. We found that scheduling strategies are diverse, but often predictable with simple heuristics. Using our guidelines, we hope future designers feel more justified and informed when integrating intermittent interaction into semi-automated fabrication systems and extend existing workflows, perhaps even fostering more meaningful engagements with technology.

7 CONCLUSION

Past research has largely avoided exploring the utility and characteristics of intermittent interactions in long duration processes. In the context of interacting with personal fabrication machines, almost no previous work has proposed a system requiring more than a single interaction. Our research has investigated this gap and found that intermittent interaction can make direct and substantial improvements to the sustainability of FDM 3D printing. In particular, returning to *RQ 1*, we show that including the user via intermittent interactions allows making changes to the fabrication process itself. This allows addressing shortcomings of the process directly, such as avoiding to fabricate wasteful but necessary supporting structures, to reduce material and energy usage of fabrication devices. We found that these process adjustments can be achieved by customizing existing workflows, such as integrating software support for intermittent interaction via plug-ins or software updates.

We can answer *RQ 2*, how intermittent interaction can enable novel output options for unmodified personal fabrication devices, after demonstrating how to perform sewing tasks using a commodity 3D printer. By including the user, additive fabrication software need not rely on the established bottom-up fabrication approach utilized by all common slicers. As a result, the fabrication process can handle alternative sequential approaches, such as tracing along a seams of cloth work pieces. Additionally, through user involvement, the system gains access to handling new types of material, such as fabric, that the automated machine itself is not equipped to manipulate.

Finally, we have identified subjective costs and benefits associated with intermittent interaction and have demonstrated how to parameterize and reduce the costs, answering *RQ 3*. To comprehensively consider user effort when planning interactions, it does not suffice to simply count the number of distinct actions to perform or to measure the time spent on these actions. Users prefer fewer, more complex and at times challenging interactions to addressing additional or simpler to handle interruptions. This preference for few but engaging interaction can even persist if users would spend overall more time interacting compared to less engaging options. Software that supports intermittent interaction should be able to distinguish between interactions of varying complexity, and should account for the number of interruptions beyond counting the overall number of actions to perform. We demonstrated that variations in these factors are relevant in how users perceive intermittent interactions, and that making these distinctions allows representing user effort during systematic optimization efforts.

All three system projects in this thesis have been designed to be compatible with existing commodity hardware and to integrate as seamlessly as possible into existing workflows to make them more accessible. By designing for affordable hardware, the potential user base is large and varied, and we found

that additional software considerations may be necessary to support this variety. We found that baking in domain knowledge can help less experienced users, such as including custom G-code for pausing a print job to perform intermittent interactions. For example, we re-calibrate the buildplate axis before resuming the print to ensure knocking against a movable buildplate does not disrupt the print. Similarly, we ensure that the default starting setup of our systems provide common options to minimize the initial cost to setup and use them. For example, we provide the option to use LEGO as support replacements without any additional setup in Substiports, and we provide a variety of default printed seam types with FabricAtion, which allow creating flexible, robust or temporary stitches right away.

In summary, we showed that intermittent interaction can improve fully automated processes, lower the effort of fully interactive processes, and open up novel interaction options in the space between. We found that people are often eager to get involved hands on, especially when their participation seems valuable, and challenging enough not to seem like busy work. We demonstrated that brief user interactions with personal fabrication can provide multiple avenues for people to take tangible action towards sustainability. Through participation, people can directly reduce waste material and can gain access to new options to make and repair locally. This could avoid the current status quo where people leave matters of sustainability to intangible commercial entities.

7.1 THE FUTURE OF INTERMITTENT INTERACTION

The intermittent interaction systems presented here follow the typical current fabrication workflow of planning the entire process ahead of time. Future work could explore systems that adapt the fabrication process mid-fabrication. Building on existing systems, this could mean offering interaction options in the moment, when a user is around, which can be performed just then. As a straightforward example, the screw jacks introduced in the Substiports system have a variable height that could, in theory, be inserted at any point during fabrication while the print height is within the extension range of the screw jack. Such a system would require the ability to inject machine code into ongoing fabrication processes, or might require machine firmware updates to allow handling optional pauses and code segments.

Long form partially automated tasks are not unique to personal fabrication, and the lessons learned how to design with human-in-the-loop in mind may well apply to other domains. As a primary, daily example, many household tasks such as cooking and baking, or household appliances with extended durations like dishwashers or vacuum robots, follow similar interaction patterns. Even outside of modern technology, intermittent interaction patterns find applications in modern life, such as how growing sourdough starter requires intermittent attention across multiple days. Future work could investigate applying intermittent interaction techniques to existing household applications and other domains.

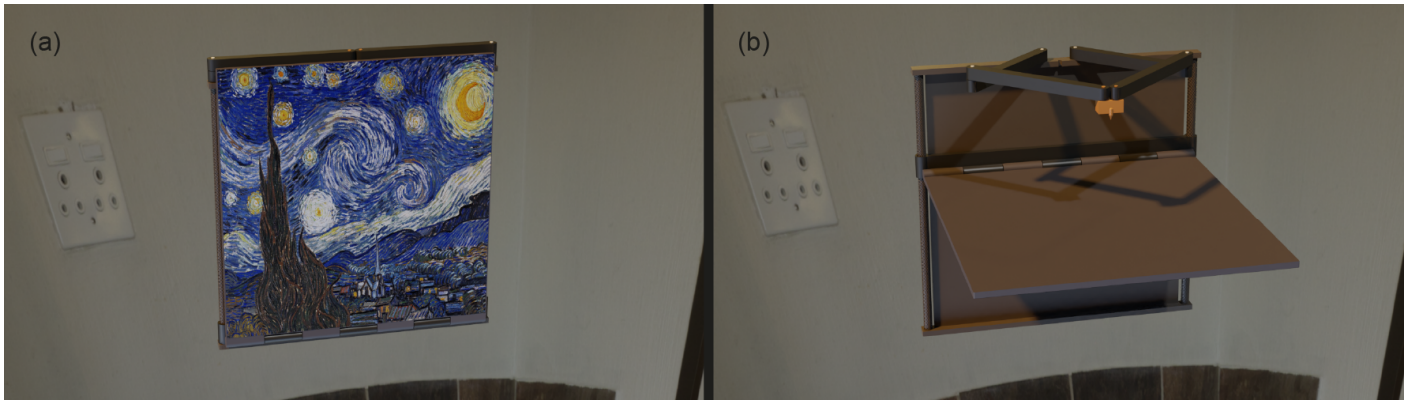


Figure 7.1: Fabrication in the living room: (a) the printer acts as decoration while out of use; (b) it unfolds to become a fully functional 3D printer. (Figure from [114])

Intermittent interaction requires users to be close by the fabrication machines. Currently, outside of workspace and makerspace environments, fabrication machines are typically placed out of sight in workshops and cellars. One of the reasons is that fabrication devices often appear as "large, clunky cubes", and the more capable a machine is, the larger it usually is as well. Although the fabrication process may be interesting to watch, the machine itself is typically not very decorative. Intermittent Interactions on the more automated side of the spectrum still work well, similar to handling appliances like laundry machines and dryers that are often stored away from living spaces due to their size and noisy operation. Notable walking distances could however spell too much effort for more frequent user involvement.

The spread of truly personal fabrication machines that share our living spaces or are small enough to be carried around like a notepad could promote all intermittent interactions. We performed a mechanical design study [114] of how future fabrication machines could achieve a smaller form factor by producing outputs beyond the confines of the machine itself. Such fabrication machines could blend seamlessly into living spaces, for example replacing the gantry of a 3D printer with a foldable pantograph that is taking the place of a painting while out of use, as shown in Figure 7.1. Pen and book shaped devices as portrayed in the study could even become portable enough to become a daily accessory, which would ensure that they are always close at hand.

7.2 TOWARDS THE PLENITUDE

In the modern life of Plenitude [32], people are surrounded by uncountable items, mass produced and standardized. With a click of a button and two-day shipping, almost anything can be added to the pile quickly and effortlessly. This availability however is limited to "products". Replacement parts are often hard to find, if it all. Access to digital information is even faster, only limited by server and network delays. Through AI systems, thoughts can be visualized

as fast as their descriptions can be written down. The availability of data and the access to digital products may however also be limited. Data is packaged as "content" in an effort to monetize it. News, music and entertainment are offered as a service that can only be accessed while online and authenticated. Even packaged products, such as boxed games, can become inaccessible overnight when a server that is pinged merely to confirm the latest software version is shut down.

Personal fabrication can act as another vector for the Plenitude. Rather than waiting days for shipping, items can be made available as soon as fabrication is complete. It is easy to envision that machine instructions could also be offered as a service, streamed directly to the machine. 3D printed articulated trinkets made from reflective filament in printerfarms can already be found in many gift shops due to eye catching colors and apparent novelty. But it can also act as an opposing vector. It can alleviate planned obsolescence by making it easier to repair items designed to fail. It allows designing lasting custom solutions for local problems, rather than working through off-the-shelf solutions to find the best fit. It allows labours of love of artists and meaningful personal projects to compete with fast fashion.

Much of the value that can be derived from personal fabrication today is based on access to open repositories of 3D models made by people who cared enough to create something unique and present it, or to solve a problem and make the solutions available publicly. Being part of the creation of something makes it easier to care for that thing, rather than just owning it, and to nurture the desire to contribute as well. Such contributions further increase the value of personal fabrication, which can attract new users, forming a positive feedback loop if they, too, care. As personal computation reaches the physical world through fabrication, keeping the human in the loop can help to ensure a positive impact of the technology.

7.3 FINAL WORD

Today, computation is ubiquitous. Yet, it is neither unlimited nor free. The billion dollar cryptocurrency industry was built on commodifying digital scarcity. Generative AI applications consume more energy than some countries [132] and access to them is limited by usage through "generation tokens". Like computation, fabrication consumes energy and requires specialized hardware, but additionally also consumes tangible resources. As such, it is safe to assume that fabrication will continue to incur costs after reaching ubiquity. Even in science fiction where any item can be "replicated" instantly from energy, access to replicators often still remains rationed. Regardless of the specific technology used, intermittent interaction will remain relevant as long as it can provide options to reduce fabrication costs or increase its sustainability.

BIBLIOGRAPHY

[1] Lotte Abildgren, Malte Lebahn-Hadidi, Christian Backer Mogensen, Palle Toft, Anders Bo Nielsen, Tove Faber Frandsen, Sune Vork Steffensen, and Lise Hounsgaard. "The effectiveness of improving healthcare teams' human factor skills using simulation-based training: a systematic review." In: *Advances in Simulation* 7.1 (May 2022), p. 12. ISSN: 2059-0628. DOI: 10.1186/s41077-022-00207-2. URL: https://doi.org/10.1186/s41077-022-00207-2.

[2] Harshit Agrawal, Udayan Umapathi, Robert Kovacs, Johannes Frohnhofen, 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*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, 427–436. ISBN: 9781450337793. DOI: 10.1145/2807442.2807505. URL: https://doi.org/10.1145/2807442.2807505.

[3] Yoh Akiyama and Homei Miyashita. "Fitter: A System for Easily Printing Objects That Fit Real Objects." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16 Adjunct. Tokyo, Japan: Association for Computing Machinery, 2016, 129–131. ISBN: 9781450345316. DOI: 10.1145/2984751.2985730. URL: https://doi.org/10.1145/2984751.2985730.

[4] Marc Alexa, Kristian Hildebrand, and Sylvain Lefebvre. "Optimal Discrete Slicing." In: *ACM Trans. Graph.* 36.1 (Jan. 2017). ISSN: 0730-0301. DOI: 10.1145/2999536. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2999536.

[5] Paul Alexander, Seth Allen, and Debasish Dutta. "Part orientation and build cost determination in layered manufacturing." In: *Computer-Aided Design* 30.5 (1998), pp. 343–356. ISSN: 0010-4485. DOI: https://doi.org/10.1016/S0010-4485(97)00083-3. URL: https://www.sciencedirect.com/science/article/pii/S0010448597000833.

[6] Stewart Allen. *Kiri:Moto: An open-source slicer for Laser cutting, 3D printing, CNC milling*. https://grid.space/kiri/. [Online; accessed 4-April-2023]. 2020.

[7] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. "Power to the People: The Role of Humans in Interactive Machine Learning." In: *AI Magazine* 35.4 (Dec. 2014), pp. 105–120. DOI: 10.1609/aimag.v35i4.2513. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2513.

[8] Inc. Autodesk. *Fusion360*. https://www.autodesk.ca/en/products/fusion-360/overview. [Online; accessed 17-September-2020]. 2020.

[9] 3devo B.V. *3devo: Turning 3D printing into a closed-loop cycle*. https://3devo.com/our-process/. [Online; accessed 17-September-2020]. 2020.

[10] Muinul Banna, Kaustav Bera, Ryan Sochol, Liwei Lin, Homayoun Najjaran, Rehan Sadiq, and Mina Hoorfar. "3D Printing-Based Integrated Water Quality Sensing System." In: *Sensors* 17.6 (2017). ISSN: 1424-8220. DOI: 10.3390/s17061336. URL: https://www.mdpi.com/1424-8220/17/6/1336.

[11] Gianluca Barile, Alfiero Leoni, Mirco Muttillo, Romina Paolucci, Gianfranco Fazzini, and Leonardo Pantoli. "Fused-Deposition-Material 3D-Printing Procedure and Algorithm Avoiding Use of Any Supports." en. In: *Sensors* 20.2 (Jan. 2020). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 470. DOI: 10.3390/s20020470. URL: https://www.mdpi.com/1424-8220/20/2/470 (visited on 01/20/2021).

[12] Patrick Baudisch. "Personal Fabrication in HCI: Trends and Challenges." In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '16. Bari, Italy: Association for Computing Machinery, 2016, 1–2. ISBN: 9781450341318. DOI: 10.1145/2909132.2934645. URL: https://doi.org/10.1145/2909132.2934645.

[13] Patrick Baudisch and Stefanie Mueller. "Personal Fabrication: State of the Art and Future Research." In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '16. San Jose, California, USA: Association for Computing Machinery, 2016, 936–939. ISBN: 9781450340823. DOI: 10.1145/2851581.2856664. URL: https://doi.org/10.1145/2851581.2856664.

[14] Gordon M. Becker, Morris H. Degroot, and Jacob Marschak. "Measuring utility by a single-response sequential method." In: *Behavioral Science* 9.3 (1964), pp. 226–232. DOI: https://doi.org/10.1002/bs.3830090304. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bs.3830090304. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/bs.3830090304.

[15] Niels van Berkel, Mikael B. Skov, and Jesper Kjeldskov. "Human-AI interaction: intermittent, continuous, and proactive." In: *Interactions* 28.6 (Nov. 2021), 67–71. ISSN: 1072-5520. DOI: 10.1145/3486941. URL: https://doi.org/10.1145/3486941.

[16] V. Braun and V. Clarke. *Thematic Analysis: A Practical Guide*. Thousand Oaks, California: SAGE Publications, 2021. ISBN: 9781526417299. URL: https://books.google.ca/books?id=eMArEAAAQBAJ.

[17] Xiang 'Anthony' Chen, Stelian Coros, and Scott E. Hudson. "Medley: A Library of Embeddables to Explore Rich Material Properties for 3D Printed Objects." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, 1–12. ISBN: 9781450356206. DOI: 10.1145/3173574.3173736. URL: https://doi.org/10.1145/3173574.3173736.

[18] 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*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, 73–82. ISBN: 9781450337793. DOI: 10.1145/2807442.2807498. URL: https://doi.org/10.1145/2807442.2807498.

[19] Paramita Das, Ramya Chandran, Rutuja Samant, and Sam Anand. "Optimum Part Build Orientation in Additive Manufacturing for Minimizing Part Errors and Support Structures." In: *Procedia Manufacturing* 1 (2015). 43rd North American Manufacturing Research Conference, NAMRC 43, 8-12 June 2015, UNC Charlotte, North Carolina, United States, pp. 343–354. ISSN: 2351-9789. DOI: https://doi.org/10.1016/j.promfg.2015.09.041. URL: https://www.sciencedirect.com/science/article/pii/S2351978915010410.

[20] Lucas W. Davis. *Evaluating the Slow Adoption of Energy Efficient Investments: Are Renters Less Likely to Have Energy Efficient Appliances?* NBER Working Papers 16114. National Bureau of Economic Research, Inc, June 2010. URL: https://ideas.repec.org/p/nbr/nberwo/16114.html.

[21] Laura Devendorf and Kimiko Ryokai. "Being the Machine: Reconfiguring Agency and Control in Hybrid Fabrication." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: Association for Computing Machinery, 2015, 2477–2486. ISBN: 9781450331456. DOI: 10.1145/2702123.2702547. URL: https://doi.org/10.1145/2702123.2702547.

[22] A. Dijkshoorn, P. Werkman, M. Welleweerd, G. Wolterink, B. Eijking, J. Delamare, R. Sanders, and G. J. M. Krijnen. "Embedded sensing: integrating sensors in 3-D printed structures." In: *Journal of Sensors and Sensor Systems* 7.1 (2018), pp. 169–181. DOI: 10.5194/jsss-7-169-2018. URL: https://jsss.copernicus.org/articles/7/169/2018/.

[23] Jérémie Dumas, Jean Hergel, and Sylvain Lefebvre. "Bridging the gap: automated steady scaffoldings for 3D printing." In: *ACM Transactions on Graphics* 33.4 (July 2014), 98:1–98:10. ISSN: 0730-0301. DOI: 10.1145/2601097.2601153. URL: https://doi.org/10.1145/2601097.2601153 (visited on 01/20/2021).

[24] Ben Ezair, Fady Massarwi, and Gershon Elber. "Orientation analysis of 3D objects toward minimal support volume in 3D-printing." In: *Computers & Graphics* 51 (2015). International Conference Shape Modeling International, pp. 117–124. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2015.05.009. URL: https://www.sciencedirect.com/science/article/pii/S0097849315000564.

[25] Gianfranco Fazzini, Paola Paolini, Romina Paolucci, Daniela Chiulli, Gianluca Barile, Alfiero Leoni, Mirco Muttillo, Leonardo Pantoli, and Giuseppe Ferri. "Print On Air: FDM 3D Printing Without Supports." In: *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 IoT)*. 2019, pp. 350–354. DOI: 10.1109/METROI4.2019.8792846.

[26] Ruiliang Feng, Xianda Li, Lin Zhu, Atul Thakur, and Xiangzhi Wei. "An Improved Two-Level Support Structure for Extrusion-Based Additive Manufacturing." en. In: *Robotics and Computer-Integrated Manufacturing* 67 (Feb. 2021), p. 101972. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2020.101972. URL: http://www.sciencedirect.com/science/article/pii/S0736584519308324 (visited on 01/20/2021).

[27] Alfonso Fernández. "Development of Actuators Using Material Extrusion Additive Manufacturing with Embedded Shape Memory Alloy Wire." MA thesis. University of Texas at El Paso, 2019.

[28] Jack Forman, Mustafa Doga Dogan, Hamilton Forsythe, and Hiroshi Ishii. "DefeXtiles: 3D Printing Quasi-Woven Fabric via Under-Extrusion." In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. Virtual Event, USA: Association for Computing Machinery, 2020, 1222–1233. ISBN: 9781450375146. DOI: 10.1145/3379337.3415876. URL: https://doi.org/10.1145/3379337.3415876.

[29] The Blender Foundation. *Blender V2.8*. https://www.blender.org/. [Online; accessed 17-September-2020]. 2020.

[30] Gabriel Gaal, Melissa Mendes, Tiago P. de Almeida, Maria H.O. Piazzetta, Ângelo L. Gobbi, Antonio Riul, and Varlei Rodrigues. "Simplified fabrication of integrated microfluidic devices using fused deposition modeling 3D printing." In: *Sensors and Actuators B: Chemical* 242 (2017), pp. 35–40. ISSN: 0925-4005. DOI: https://doi.org/10.1016/j.snb.2016.10.110. URL: https://www.sciencedirect.com/science/article/pii/S0925400516317373.

[31] 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*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, 437–446. ISBN: 9781450337793. DOI: 10.1145/2807442.2807476. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2807442.2807476.

[32] R. Gold. *The Plenitude: Creativity, Innovation, and Making Stuff*. Simplicity (Cambridge, Mass.) MIT Press, 2007. ISBN: 9780262072892. URL: https://books.google.ca/books?id=-kbuAAAAMAAJ.

[33] James Gopsill, Jonathan Shindler, and Ben Hicks. "Using finite element analysis to influence the infill design of fused deposition modelled parts." In: *Progress in Additive Manufacturing* 3 (Nov. 2017). DOI: 10.1007/s40964-017-0034-y.

[34] 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*. CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, 420–432. ISBN: 9781450333627. DOI: 10.1145/2858036.2858191. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2858036.2858191.

[35] Ammar Hattab and Gabriel Taubin. "Rough Carving of 3D Models with Spatial Augmented Reality." In: *Proceedings of the ACM Symposium on Computational Fabrication*. SCF '19. Pittsburgh, Pennsylvania: Association for Computing Machinery, 2019. ISBN: 9781450367950. DOI: 10.1145/3328939.3328998. URL: https://doi.org/10.1145/3328939.3328998.

[36] Jiman Hong, D.L. Wu, D.C. Li, and B.H. Lu. "Multi-objective optimization of the part building orientation in stereolithography." In: 35 (May 2001), pp. 506–509.

[37] Shen Hongyao, Ye Xiaoxiang, and Fu Jianzhong. "Research on the flexible support platform for fused deposition modeling." en. In: *The International Journal of Advanced Manufacturing Technology* 97.9 (Aug. 2018), pp. 3205–3221. ISSN: 1433-3015. DOI: 10.1007/s00170-018-2046-2. URL: https://doi.org/10.1007/s00170-018-2046-2 (visited on 01/20/2021).

[38] Xiaomao Huang, Chunsheng Ye, Siyu Wu, Kaibo Guo, and Jian-hua Mo. "Sloping wall structure support generation for fused deposition modeling." In: *International Journal of Advanced Manufacturing Technology* 42 (June 2009), pp. 1074–1081. DOI: 10.1007/s00170-008-1675-2.

[39] Gina Häußge. *The snappy web interface for your 3D printer.* https://octoprint.org/. [Online; accessed 21-July-2024]. 2024.

[40] American & Efird Canada Inc. *STITCHES PER INCH (SPI) - WHAT YOU SHOULD KNOW*. https://www.amefird.com/wp-content/uploads/2010/01/Selecting-the-right-SPI-2-5-10.pdf. [Online; accessed 22-July-2024]. 2010.

[41] Alec Jacobson. "Generalized Matryoshka: Computational Design of Nesting Objects." In: *Computer Graphics Forum* 36.5 (2017), pp. 27–35. DOI: 10.1111/cgf.13242. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13242. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13242.

[42] Alec Jacobson. "Generalized Matryoshka: Computational Design of Nesting Objects." In: *Computer Graphics Forum* 36 (Aug. 2017), pp. 27–35. DOI: 10.1111/cgf.13242.

[43] Seongje Jang, Byungjin Moon, and Kunwoo Lee. "Free-Floating Support Structure Generation." en. In: *Computer-Aided Design* 128 (Nov. 2020), p. 102908. ISSN: 0010-4485. DOI: 10.1016/j.cad.2020.102908. URL: http://www.sciencedirect.com/science/article/pii/S0010448520301019 (visited on 01/20/2021).

[44] Jingchao Jiang, Jonathan Stringer, and Xun Xu. "Support Optimization for Flat Features via Path Planning in Additive Manufacturing." In: *3D Printing and Additive Manufacturing* 6.3 (Nov. 2018). Publisher: Mary Ann Liebert, Inc., publishers, pp. 171–179. ISSN: 2329-7662. DOI: 10.1089/3dp.2017.0124. URL: https://www.liebertpub.com/doi/10.1089/3dp.2017.0124 (visited on 01/20/2021).

[45] Jingchao Jiang, Xun Xu, and Jonathan Stringer. "Optimisation of multi-part production in additive manufacturing for reducing support waste." In: *Virtual and Physical Prototyping* 14.3 (July 2019). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/17452759.2019.1585555, pp. 219–228. ISSN: 1745-2759. DOI: 10.1080/17452759.2019.1585555. URL: https://doi.org/10.1080/17452759.2019.1585555 (visited on 01/20/2021).

[46] Andrew Jones, Thomas Cameron, Benjamin Eichholz, Martin Eichers, Taylor Kray, and Jeremy Straub. "Reducing space sensing and other mission cost with 3D printing infill optimization." In: *Sensors and Systems for Space Applications XII*. Ed. by Genshe Chen and Khanh D. Pham. Vol. 11017. International Society for Optics and Photonics. Bellingham, Washington USA: SPIE, 2019, pp. 69 –74. DOI: 10.1117/12.2520261. URL: https://doi.org/10.1117/12.2520261.

[47] Shohei Katakura, Yuto Kuroki, and Keita Watanabe. "A 3D Printer Head as a Robotic Manipulator." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, 535–548. ISBN: 9781450368162. DOI: 10.1145/3332165.3347885. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3332165.3347885.

[48] Shohei Katakura, Yuto Kuroki, and Keita Watanabe. "A 3D Printer Head as a Robotic Manipulator." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, 535–548. ISBN: 9781450368162. DOI: 10.1145/3332165.3347885. URL: https://doi.org/10.1145/3332165.3347885.

[49] J. Kennedy and R. Eberhart. "Particle swarm optimization." In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. Perth, WA, Australia: IEEE, 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[50] J. Kennedy and R. Eberhart. "Particle swarm optimization." In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[51] Jeeeun Kim, Anhong Guo, Tom Yeh, Scott E. Hudson, and Jennifer Mankoff. "Understanding Uncertainty in Measurement and Accommodating Its Impact in 3D Modeling and Printing." In: *Proceedings of the 2017 Conference on Designing Interactive Systems*. DIS '17. Edinburgh, United Kingdom: Association for Computing Machinery, 2017,

1067–1078. ISBN: 9781450349222. DOI: 10.1145/3064663.3064690. URL: https://doi.org/10.1145/3064663.3064690.

[52] Marius Kintel and Clifford Wolf. *OpenSCAD: The Programmers Solid 3D CAD Modeller*. https://openscad.org/index.html. [Online; accessed 21-July-2024]. 2020.

[53] Philip J. Kitson, Ross J. Marshall, Deliang Long, Ross S. Forgan, and Leroy Cronin. "3D Printed High-Throughput Hydrothermal Reactionware for Discovery, Optimization, and Scale-Up." In: *Angewandte Chemie International Edition* 53.47 (2014), pp. 12723–12728. DOI: https://doi.org/10.1002/anie.201402654. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201402654. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201402654.

[54] Philip J. Kitson, Mark D. Symes, Vincenza Dragone, and Leroy Cronin. "Combining 3D printing and liquid handling to produce user-friendly reactionware for chemical synthesis and purification." In: *Chem. Sci.* 4 (8 2013), pp. 3099–3103. DOI: 10.1039/C3SC51253C. URL: http://dx.doi.org/10.1039/C3SC51253C.

[55] Robert Kovacs et al. "TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 2606–2616. ISBN: 9781450346559. DOI: 10.1145/3025453.3026016. URL: https://doi.org/10.1145/3025453.3026016.

[56] Jusung Lee and Kunwoo Lee. "Block-based inner support structure generation algorithm for 3D printing using fused deposition modeling." In: *The International Journal of Advanced Manufacturing Technology* 89 (Mar. 2017), pp. 2151–2163. DOI: 10.1007/s00170-016-9239-3.

[57] Kwang Y. Lee and Jong-bae Park. "Application of Particle Swarm Optimization to Economic Dispatch Problem: Advantages and Disadvantages." In: *2006 IEEE PES Power Systems Conference and Exposition*. 2006, pp. 188–192. DOI: 10.1109/PSCE.2006.296295.

[58] Danny Leen, Nadya Peek, and Raf Ramakers. "LamiFold: Fabricating Objects with Integrated Mechanisms Using a Laser cutter Lamination Workflow." In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. UIST '20. Virtual Event, USA: Association for Computing Machinery, 2020, 304–316. ISBN: 9781450375146. DOI: 10.1145/3379337.3415885. URL: https://doi.org/10.1145/3379337.3415885.

[59] S. Lloyd. "Least squares quantization in PCM." In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[60] Jonas Lundberg, Magnus Bång, Jimmy Johansson, Ali Cheaitou, Billy Josefsson, and Zain Tahboub. "Human-in-the-loop AI: Requirements on future (unified) air traffic management systems." In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. 2019, pp. 1–9. DOI: 10.1109/DASC43569.2019.9081674.

[61] LLC MakerBot Industries. *MakerBot Thingiverse*. https://www.thingiverse.com/. [Online; accessed 17-September-2020]. 2020.

[62] Laura Marcano, Finn Aakre Haugen, Ronny Sannerud, and Tiina Komulainen. "Review of simulator training practices for industrial operators: How can individual simulator training be enabled?" In: *Safety Science* 115 (2019), pp. 414–424. ISSN: 0925-7535. DOI: https://doi.org/10.1016/j.ssci.2019.02.019. URL: https://www.sciencedirect.com/science/article/pii/S0925753518305721.

[63] Microsoft. *Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux*. https://code.visualstudio.com/. [Online; accessed 21-July-2024]. 2024.

[64] Sam Morgan. *Air traffic control chief: Humans will always be in the loop*. https://www.euractiv.com/section/aerospace/interview/air-traffic-control-chief-humans-will-always-be-in-the-loop/. [Online; accessed 21-July-2024]. 2019.

[65] Clark E. Moustakas. *Phenomenological research methods*. Thousand Oaks, California: SAGE Publications, Inc., 1994. DOI: 10.4135/9781412995658. URL: https://doi.org/10.4135/9781412995658.

[66] Stefanie Mueller. "Rethinking Interaction Techniques for Personal Fabrication." In: *IEEE Computer Graphics and Applications* 38.5 (2018), pp. 18–25. DOI: 10.1109/MCG.2018.053491727.

[67] 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*. UIST '14. Honolulu, Hawaii, USA: Association for Computing Machinery, 2014, 273–280. ISBN: 9781450330695. DOI: 10.1145/2642918.2647359. URL: https://doi.org/10.1145/2642918.2647359.

[68] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. "Interactive Construction: Interactive Fabrication of Functional Mechanical Devices." In: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. UIST '12. Cambridge, Massachusetts, USA: Association for Computing Machinery, 2012, 599–606. ISBN: 9781450315807. DOI: 10.1145/2380116.2380191. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2380116.2380191.

[69] Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. "FaBrickation: Fast 3D Printing of Functional Objects by Integrating Construction Kit Building Blocks." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, 3827–3834. ISBN: 9781450324731. DOI: 10.1145/2556288.2557005. URL: https://doi.org/10.1145/2556288.2557005.

[70] Stefanie Mueller, Anna Seufert, Huaishu Peng, Robert Kovacs, Kevin Reuss, François Guimbretière, and Patrick Baudisch. "FormFab: Continuous Interactive Fabrication." In: *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '19. Tempe, Arizona, USA: Association for Computing Machinery, 2019, 315–323. ISBN: 9781450361965. DOI: 10.1145/3294109.3295620. URL: https://doi.org/10.1145/3294109.3295620.

[71] Sachith Muthukumarana, Moritz Alexander Messerschmidt, Denys J.C. Matthies, Jürgen Steimle, Philipp M. Scholl, and Suranga Nanayakkara. "ClothTiles: A Prototyping Platform to Fabricate Customized Actuators on Clothing using 3D Printing and Shape-Memory Alloys." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: 10.1145/3411764.3445613. URL: https://doi.org/10.1145/3411764.3445613.

[72] René K. Müller. *3D Printing: 90° Overhangs without Support Structure with Non-Planar Slicing on 3-axis Printer*. https://xyzdims.com/2021/03/03/3d-printing-90-overhangs-without-support-structure-with-non-planar-slicing-on-3-axis-printer/. [Online; accessed 4-April-2023]. 2021.

[73] Martin Nisser, Christina Chen Liao, Yuchen Chai, Aradhana Adhikari, Steve Hodges, and Stefanie Mueller. "LaserFactory: A Laser Cutter-based Electromechanical Assembly and Fabrication Platform to Make Functional Devices & Robots." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: 10.1145/3411764.3445692. URL: https://doi.org/10.1145/3411764.3445692.

[74] Michael I. Norton, Daniel Mochon, and Dan Ariely. "The IKEA effect: When labor leads to love." In: *Journal of Consumer Psychology* 22.3 (2012), pp. 453–460. ISSN: 1057-7408. DOI: https://doi.org/10.1016/j.jcps.2011.08.002. URL: https://www.sciencedirect.com/science/article/pii/S1057740811000829.

[75] William T. Odom, Abigail J. Sellen, Richard Banks, David S. Kirk, Tim Regan, Mark Selby, Jodi L. Forlizzi, and John Zimmerman. "Designing for Slowness, Anticipation and Re-Visitation: A Long Term Field Study of the Photobox." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, On-

tario, Canada: Association for Computing Machinery, 2014, 1961–1970. ISBN: 9781450324731. DOI: 10.1145/2556288.2557178. URL: https://doi.org/10.1145/2556288.2557178.

[76] William Odom, Ron Wakkary, Ishac Bertran, Matthew Harkness, Garnet Hertz, Jeroen Hol, Henry Lin, Bram Naus, Perry Tan, and Pepijn Verburg. "Attending to Slowness and Temporality with Olly and Slow Game: A Design Inquiry Into Supporting Longer-Term Relations with Everyday Computational Objects." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3173651. URL: https://doi.org/10.1145/3173574.3173651.

[77] Jukka Pakkanen, Diego Manfredi, Paolo Minetola, and Luca Iuliano. "About the Use of Recycled or Biodegradable Filaments for Sustainability of 3D Printing." In: *Sustainable Design and Manufacturing 2017*. Ed. by Giampaolo Campana, Robert J. Howlett, Rossi Setchi, and Barbara Cimatti. Cham: Springer International Publishing, 2017, pp. 776–785. ISBN: 978-3-319-57078-5.

[78] Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. "RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, 1–12. ISBN: 9781450356206. DOI: 10.1145/3173574.3174153. URL: https://doi.org/10.1145/3173574.3174153.

[79] 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*. CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, 887–896. ISBN: 9781450333627. DOI: 10.1145/2858036.2858106. URL: https://doi.org/10.1145/2858036.2858106.

[80] Cody W. Pinger, Andrew A. Heller, and Dana M. Spence. "A Printed Equilibrium Dialysis Device with Integrated Membranes for Improved Binding Affinity Measurements." In: *Analytical Chemistry* 89.14 (2017). PMID: 28648046, pp. 7302–7306. DOI: 10.1021/acs.analchem.7b01848. URL: https://doi.org/10.1021/acs.analchem.7b01848.

[81] Jiankang Qiu, Lifang Wu, and Yuxin Mao. "A Novel Supporting Structure Generation Scheme to 3D Printing." In: *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*. ICIMCS '15. Zhangjiajie, Hunan, China: Association for Computing Machinery, 2015. ISBN: 9781450335287. DOI: 10.1145/2808492.2808561. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2808492.2808561.

[82] Tim Reiner and Sylvain Lefebvre. "Interactive Modeling of Support-free Shapes for Fabrication." In: *EG 2016 - Short Papers*. Ed. by T. Bashford-Rogers and L. P. Santos. The Eurographics Association, 2016. DOI: 10.2312/egsh.20161006.

[83] Prusa Research. *PrusaSlicer: An open-source, feature-rich, frequently updated tool that contains everything you need to export the perfect print files for your 3D printer.* https://github.com/prusa3d/PrusaSlicer/. [Online; accessed 27-July-2023]. 2016.

[84] Prusa Research. *PrusaSlicer: G-code generator for 3D printers*. https://www.thingiverse.com/. [Online; accessed 17-September-2020]. 2020.

[85] 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*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 497–508. ISBN: 9781450346559. DOI: 10.1145/3025453.3025460. URL: https://doi.org/10.1145/3025453.3025460.

[86] Alec Rivers, Andrew Adams, and Frédo Durand. "Sculpting by Numbers." In: *ACM Trans. Graph.* 31.6 (Nov. 2012). ISSN: 0730-0301. DOI: 10.1145/2366145.2366176. URL: https://doi.org/10.1145/2366145.2366176.

[87] Bruna Goveia da Rocha, Johannes M. L. van der Kolk, and Kristina Andersen. "Exquisite Fabrication: Exploring Turn-Taking between Designers and Digital Fabrication Machines." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: 10.1145/3411764.3445236. URL: \url{https://doi.org/10.1145/3411764.3445236}.

[88] Branko Rogic, Demin Nalic, Arno Eichberger, and Stefan Bernsteiner. "A Novel Approach to Integrate Human-in-the-Loop Testing in the Development Chain of Automated Driving: The Example of Automated Lane Change**This work is founded by the Austrian Federal Ministry of Transport, Innovation and Technology as part of the FFG Program "EFREtop"." In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 10188–10195. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2020.12.2747. URL: https://www.sciencedirect.com/science/article/pii/S2405896320335102.

[89] Thijs Roumen, Bastian Kruck, Tobias Dürschmid, Tobias Nack, and Patrick Baudisch. "Mobile Fabrication." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: Association for Computing Machinery, 2016, 3–14. ISBN: 9781450341899. DOI: 10.1145/2984511.2984586. URL: https://doi.org/10.1145/2984511.2984586.

[90] Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. "Large-Scale Assessment of Mobile Notifications." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, 3055–3064. ISBN: 9781450324731. DOI: 10.1145/2556288.2557189. URL: https://doi.org/10.1145/2556288.2557189.

[91] Oliver Sand, Sebastian Büttner, Volker Paelke, and Carsten Röcker. "smARt.Assembly – Projection-Based Augmented Reality for Supporting Assembly Workers." In: *Virtual, Augmented and Mixed Reality*. Ed. by Stephanie Lackey and Randall Shumaker. Cham: Springer International Publishing, 2016, pp. 643–652. ISBN: 978-3-319-39907-2.

[92] Joy Sarkar, Md Abdullah Al Faruque, and Moni Mondal. "Modeling the seam strength of denim garments by using fuzzy expert system." In: *Journal of Engineered Fibers and Fabrics* 16 (Feb. 2021), pp. 1–10. DOI: 10.1177/1558925021988976.

[93] Ryan Schmidt and Nobuyuki Umetani. "Branching Support Structures for 3D Printing." In: *ACM SIGGRAPH 2014 Studio*. SIGGRAPH '14. Vancouver, Canada: Association for Computing Machinery, 2014. ISBN: 9781450329774. DOI: 10.1145/2619195.2656293. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2619195.2656293.

[94] Eldon Schoop, Michelle Nguyen, Daniel Lim, Valkyrie Savage, Sean Follmer, and Björn Hartmann. "Drill Sergeant: Supporting Physical Construction Projects through an Ecosystem of Augmented Tools." In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '16. San Jose, California, USA: Association for Computing Machinery, 2016, 1607–1614. ISBN: 9781450340823. DOI: 10.1145/2851581.2892429. URL: https://doi.org/10.1145/2851581.2892429.

[95] Gianmario Scotti, Sofia M. E. Nilsson, Markus Haapala, Päivi Pöhö, Gustav Boije af Gennäs, Jari Yli-Kauhaluoma, and Tapio Kotiaho. "A miniaturised 3D printed polypropylene reactor for online reaction analysis by mass spectrometry." In: *React. Chem. Eng.* 2 (3 2017), pp. 299–303. DOI: 10.1039/C7RE00015D. URL: http://dx.doi.org/10.1039/C7RE00015D.

[96] P. Frazer Seymour, Justin Matejka, Geoff Foulds, Ihor Petelycky, and Fraser Anderson. "AMI: An Adaptable Music Interface to Support the Varying Needs of People with Dementia." In: *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. ASSETS '17. Baltimore, Maryland, USA: Association for Computing Machinery, 2017, 150–154. ISBN: 9781450349260. DOI: 10.1145/3132525.3132557. URL: https://doi.org/10.1145/3132525.3132557.

[97] C. Shemelya, F. Cedillos, E. Aguilera, D. Espalin, D. Muse, R. Wicker, and E. MacDonald. "Encapsulated Copper Wire and Copper Mesh Capacitive Sensing for 3-D Printing Applications." In: *IEEE Sensors Journal* 15.2 (2015), pp. 1280–1286.

[98] Gabriel Skantze. "Turn-taking in Conversational Systems and Human-Robot Interaction: A Review." In: *Computer Speech & Language* 67 (2021), p. 101178. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2020.101178. URL: https://www.sciencedirect.com/science/article/pii/S088523082030111X.

[99] Snapmaker. *Snapmaker 2.0 Modular 3-in-1 3D Printer*. https://www.snapmaker.com/snapmaker-2. [Online; accessed 21-July-2024]. 2017.

[100] Albert Solernou, Richard Romano, Ioannis Souflas, Foroogh Hajiseyedjavadi, Evangelos Paschalidis, and Natasha Merat. "Integration and training of a ROS autonomous driver for human-like driving style in a complex multi-component driving simulator." In: *Proceedings of the Driving Simulation Conference 2020 Europe VR*. Ed. by Andras Kemeny, Jean-Rémy Chardonnet, and Florent Colombet. Driving Simulation Association. Antibes, France, Sept. 9, 2020, pp. 11–18.

[101] G. Strano, L. Hao, R. M. Everson, and K. E. Evans. "A new approach to the design and optimisation of support structures in additive manufacturing." en. In: *The International Journal of Advanced Manufacturing Technology* 66.9 (June 2013), pp. 1247–1254. ISSN: 1433-3015. DOI: 10.1007/s00170-012-4403-x. URL: https://doi.org/10.1007/s00170-012-4403-x (visited on 01/20/2021).

[102] Haruki Takahashi and Jeeeun Kim. "3D Pen + 3D Printer: Exploring the Role of Humans and Fabrication Machines in Creative Making." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, 1–12. ISBN: 9781450359702. DOI: 10.1145/3290605.3300525. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3290605.3300525.

[103] Haruki Takahashi and Jeeeun Kim. "3D Printed Fabric: Techniques for Design and 3D Weaving Programmable Textiles." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, 43–51. ISBN: 9781450368162. DOI: 10.1145/3332165.3347896. URL: https://doi.org/10.1145/3332165.3347896.

[104] 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*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, 83–91. ISBN: 9781450337793. DOI: 10.1145/2807442.2807467. URL: https://doi.org/10.1145/2807442.2807467.

[105] Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. "Turn-by-Wire: Computationally Mediated Physical Fabrication." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New Orleans, LA, USA: Association for Computing Machinery, 2019, 713–725. ISBN: 9781450368162. DOI: 10.1145/3332165.3347918. URL: https://doi.org/10.1145/3332165.3347918.

[106] Ravi Toor. *The 3D Printing Waste Problem: Causes and Solutions*. https://www.linkedin.com/pulse/3d-printing-waste-problem-causes-solutions-ravi-toor-cbgee/. [Online; accessed 21-July-2024]. 2023.

[107] Wenn-Chieh Tsai, Po-Hao Wang, Hung-Chi Lee, Rung-Huei Liang, and Jane Hsu. "The Reflexive Printer: Toward Making Sense of Perceived Drawbacks in Technology-Mediated Reminiscence." In: *Proceedings of the 2014 Conference on Designing Interactive Systems*. DIS '14. Vancouver, BC, Canada: Association for Computing Machinery, 2014, 995–1004. ISBN: 9781450329026. DOI: 10.1145/2598510.2598589. URL: https://doi.org/10.1145/2598510.2598589.

[108] UltiMaker. *UltiMaker Cura: free, easy-to-use 3D printing software trusted by millions of users. Fine-tune your 3D model with 400+ settings for the best slicing and printing results*. https://ultimaker.com/software/ultimaker-cura/. [Online; accessed 27-July-2023]. 2014.

[109] Udayan Umapathi, Hsiang-Ting Chen, Stefanie Mueller, Ludwig Wall, Anna Seufert, and Patrick Baudisch. "LaserStacker: Fabricating 3D Objects by Laser Cutting and Welding." In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, 575–582. ISBN: 9781450337793. DOI: 10.1145/2807442.2807512. URL: https://doi.org/10.1145/2807442.2807512.

[110] J. Vanek, J. A. G. Galicia, and B. Benes. "Clever Support: Efficient Support Structure Generation for Digital Fabrication." In: *Computer Graphics Forum* 33.5 (2014), pp. 117–125. DOI: https://doi.org/10.1111/cgf.12437. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12437. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12437.

[111] J. Vanek, J. A. Garcia Galicia, B. Benes, R. Měch, N. Carr, O. Stava, and G. S. Miller. "PackMerger: A 3D Print Volume Optimizer." In: *Computer Graphics Forum* 33.6 (2014). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12353, pp. 322–332. ISSN: 1467-8659. DOI: https://doi.org/10.1111/cgf.12353. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12353 (visited on 01/20/2021).

[112] Joshua Vasquez, Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. "Jubilee: An Extensible Machine for Multi-tool Fabrication." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for

Computing Machinery, 2020, 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376425. URL: https://doi.org/10.1145/3313831.3376425.

[113] Adrian Wagner, Paul Miles Preuschoff, Philipp Wacker, Simon Voelker, and Jan Borchers. "FabricFaces: Combining Textiles and 3D Printing for Maker-Friendly Folding-Based Assembly." In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI EA '23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 9781450394222. DOI: 10.1145/3544549.3585854. URL: https://doi.org/10.1145/3544549.3585854.

[114] Ludwig Wilhelm Wall. "-1D Fabrication: Investigating Decoupling Dimensions of Fabrication Output and Fabrication Machine Size." In: *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '22. Daejeon, Republic of Korea: Association for Computing Machinery, 2022. ISBN: 9781450391474. DOI: 10.1145/3490149.3502423. URL: https://doi.org/10.1145/3490149.3502423.

[115] Ludwig Wilhelm Wall, Alec Jacobson, Daniel Vogel, and Oliver Schneider. "Scrappy: Using Scrap Material as Infill to Make Fabrication More Sustainable." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: 10.1145/3411764.3445187. URL: https://doi.org/10.1145/3411764.3445187.

[116] Ludwig Wilhelm Wall, Oliver Schneider, and Daniel Vogel. "Substiports: User-Inserted Ad Hoc Objects as Reusable Structural Support for Unmodified FDM 3D Printers." In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. UIST '23. San Francisco, CA, USA: Association for Computing Machinery, 2023. ISBN: 9798400701320. DOI: 10.1145/3586183.3606718. URL: https://doi.org/10.1145/3586183.3606718.

[117] 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*. UIST '15. Charlotte, NC, USA: Association for Computing Machinery, 2015, 93–102. ISBN: 9781450337793. DOI: 10.1145/2807442.2807451. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2807442.2807451.

[118] Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen. "MixFab: A Mixed-Reality Environment for Personal Fabrication." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, 3855–3864. ISBN: 9781450324731. DOI: 10.1145/2556288.2557090. URL: https://doi.org/10.1145/2556288.2557090.

[119] Chenming Wu, Chengkai Dai, Guoxin Fang, Yong-Jin Liu, and Charlie C.L. Wang. "RoboFDM: A robotic system for support-free fabrication using FDM." In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 1175–1180. DOI: 10.1109/ICRA.2017.7989140.

[120] J. Wu, N. Aage, R. Westermann, and O. Sigmund. "Infill Optimization for Additive Manufacturing—Approaching Bone-Like Porous Structures." In: *IEEE Transactions on Visualization and Computer Graphics* 24.2 (2018), pp. 1127–1140.

[121] Yang Xu, Ziqi Wang, Siyu Gong, and Yong Chen. "Reusable support for additive manufacturing." en. In: *Additive Manufacturing* 39 (Mar. 2021), p. 101840. ISSN: 2214-8604. DOI: 10.1016/j.addma.2021.101840. URL: http://www.sciencedirect.com/science/article/pii/S2214860421000051 (visited on 01/20/2021).

[122] Suguru Yamada, Hironao Morishige, Hiroki Nozaki, Masaki Ogawa, Takuro Yonezawa, and Hideyuki Tokuda. "ReFabricator: Integrating Everyday Objects for Digital Fabrication." In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '16. San Jose, California, USA: Association for Computing Machinery, 2016, 3804–3807. ISBN: 9781450340823. DOI: 10.1145/2851581.2890237. URL: https://doi.org/10.1145/2851581.2890237.

[123] I. E. Yiğit, M. Isa, and I. Lazoglu. "ADDITIVE MANUFACTURING WITH MODULAR SUPPORT STRUCTURES." In: 2018.

[124] Ya-Ting Yue, Xiaolong Zhang, Yongliang Yang, Gang Ren, Yi-King Choi, and Wenping Wang. "WireDraw: 3D Wire Sculpturing Guided with Mixed Reality." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 3693–3704. ISBN: 9781450346559. DOI: 10.1145/3025453.3025792. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/3025453.3025792.

[125] Amanda K. Yung, Zhiyuan Li, and Daniel Ashbrook. "Printy3D: In-Situ Tangible Three-Dimensional Design for Augmented Fabrication." In: *Proceedings of the 17th ACM Conference on Interaction Design and Children*. IDC '18. Trondheim, Norway: Association for Computing Machinery, 2018, 181–194. ISBN: 9781450351522. DOI: 10.1145/3202185.3202751. URL: https://doi.org/10.1145/3202185.3202751.

[126] Clement Zheng, Ellen Yi-Luen Do, and Jim Budd. "Joinery: Parametric Joint Generation for Laser Cut Assemblies." In: *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*. C&C '17. Singapore, Singapore: Association for Computing Machinery, 2017, 63–74. ISBN: 9781450344036. DOI: 10.1145/3059454.3059459. URL: https://doi.org/10.1145/3059454.3059459.

[127] Kening Zhu, Alexandru Dancu, and Shengdong Zhao. "FusePrint: A DIY 2.5D Printing Technique for Good-Fit Fabrication with Daily Objects." In: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, 413–416. ISBN: 9781450346566. DOI: 10.1145/3027063.3050430. URL: https://doi.org/10.1145/3027063.3050430.

[128] Amit Zoran and Joseph A. Paradiso. "FreeD: A Freehand Digital Sculpting Tool." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: Association for Computing Machinery, 2013, 2613–2616. ISBN: 9781450318990. DOI: 10.1145/2470654.2481361. URL: https://doi.org/10.1145/2470654.2481361.

[129] Amit Zoran and Joseph Paradiso. "The FreeD: A Handheld Digital Milling Device for Craft and Fabrication." In: *Adjunct Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. UIST Adjunct Proceedings '12. Cambridge, Massachusetts, USA: Association for Computing Machinery, 2012, 3–4. ISBN: 9781450315821. DOI: 10.1145/2380296.2380299. URL: https://doi.org/10.1145/2380296.2380299.

[130] Amit Zoran, Roy Shilkrot, and Joseph Paradiso. "Human-Computer Interaction for Hybrid Carving." In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. UIST '13. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, 2013, 433–440. ISBN: 9781450322683. DOI: 10.1145/2501988.2502023. URL: https://doi-org.proxy.lib.uwaterloo.ca/10.1145/2501988.2502023.

[131] Fabiano M. de Oliveira, Edmar I. de Melo, and Rodrigo A.B. da Silva. "3D Pen: A low-cost and portable tool for manufacture of 3D-printed sensors." In: *Sensors and Actuators B: Chemical* 321 (2020), p. 128528. ISSN: 0925-4005. DOI: https://doi.org/10.1016/j.snb.2020.128528. URL: https://www.sciencedirect.com/science/article/pii/S0925400520308741.

[132] Alex de Vries. "The growing energy footprint of artificial intelligence." In: *Joule* 7.10 (2023), pp. 2191–2194. ISSN: 2542-4351. DOI: https://doi.org/10.1016/j.joule.2023.09.004. URL: https://www.sciencedirect.com/science/article/pii/S2542435123003653.

APPENDICES

APPENDIX A

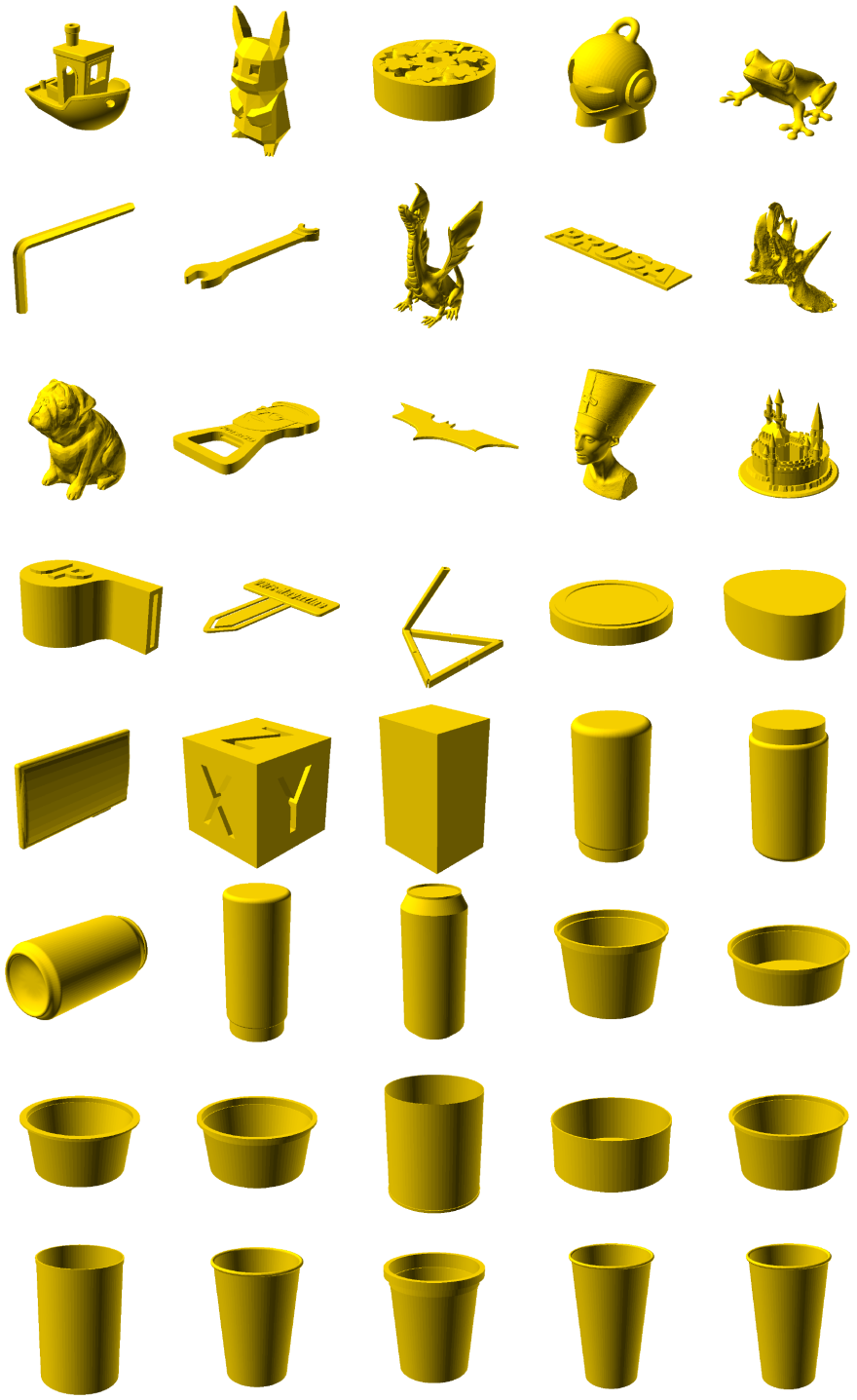


Figure A.1: The 40 objects that form the initial scrap library used during evaluation of the Scrappy system.

APPENDIX B

This appendix includes details for all Substiports printing and slicing tests and the model test set used in Substiports.

B.1 *Substiports Model Test Set*

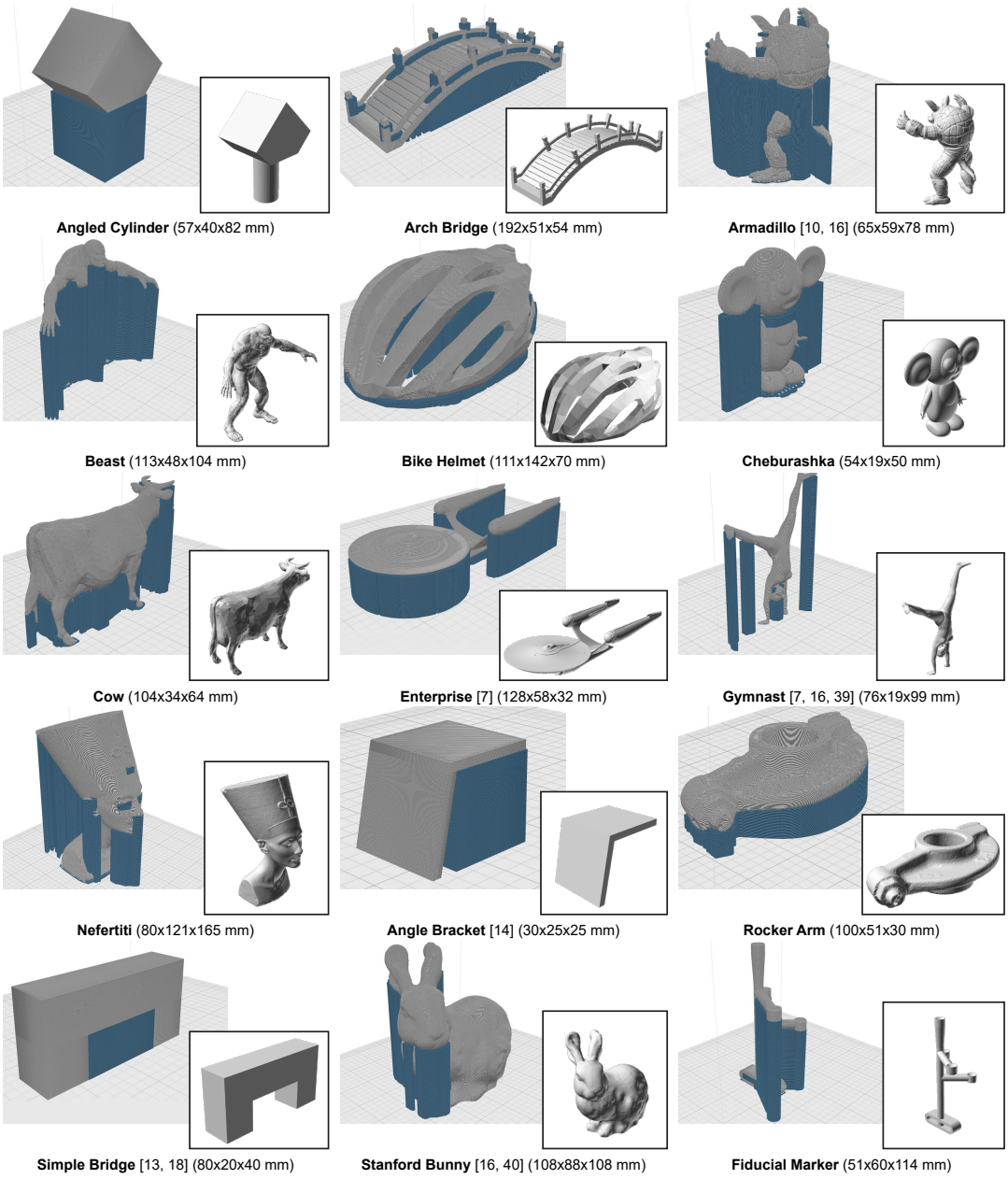
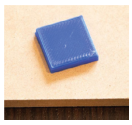


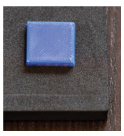
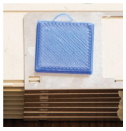
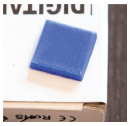
Figure B.1: The 15 models that were used for testing the Substiports system.

B.2 *Testing Printing on Objects*

This section reports on six tests related to printing on top of replacement objects: adhesion to different materials and their re-usability; filament type; tolerances; effects of extended idle periods; support structure type; and tipping resistance of replacement objects. All tests use a standard 0.4 mm nozzle and recommended print settings for the filament type used.

Table B.1: Object surface adhesion tests.

Material	Adhesion	Degradation	Notes	
PLA	Strong	Minor surface marring (melting)	Using support objects that match the printing material results in strong adhesion, but can usually still be separated easily by hand, much like regular printed supports.	
ABS	Glue/tape required	Minor surface marring (melting)	Large assemblies of toy bricks should be tightened before use. In case of the stacks used, simply pressing down on the stack with some force is sufficient.	
PVC (Floppy Disc)	Strong	Little to none	Surface ridges and paper labels have no noticeable effect on adhesion, but the metal shutter may require glue/tape.	
Medium Density Fiberboard (MDF)	Strong	Little to none		
Hardwood (Walnut)	Recommend Glue/tape	Little to none	The level of adhesion depends on the type of wood and its surface finish, ranging from strong to poor.	
Softwood (Pine)	Recommend Glue/tape	Little to none	The level of adhesion depends on the type of wood and its surface finish, ranging from strong to poor.	
High Density Foam (isolation board)	Strong	Surface marring (melting)	Some compression during printing, some minor melting due to print.	

Material	Adhesion	Degradation	Notes	
Medium Density Foam	Strong	Surface marring (melting)	Some compression during printing, some minor melting due to print.	
Low Density Foam	Warping	Major melting	Even with tape applied, the foam below may melt, which can cause warping to the material above. For foam with large pores, the tape itself may not adhere well to the foam.	
Metal	Glue/tape required	Little to none		
Cardboard (box)	Strong	Surface marring	Some compression during printing. Some delamination after removing printed material.	
Hardcover Book	Recommend Glue/tape	Surface marring (delamination)	To protect the cover from surface degradation, blue tape should be applied. The level of adhesion otherwise may also depend on the cover material.	
Softcover Book	Recommend Glue/tape	Surface marring (delamination)	To protect the cover from surface degradation, blue tape should be applied. The level of adhesion otherwise may also depend on the cover material.	

B.2.1 *Object Surface Adhesion*

The approach relies on the ability to print on top of foreign support replacement objects. We conducted tests to examine filament adhesion with common replacement object materials. A 1 mm thick (5 layer) 2*cm* square of standard PLA was printed on top of 14 different replacement objects with different material properties. Additionally, 1 to 3 mm of support was printed in between the square and the replacement objects. The level of adhesion was judged by visual inspection and light forces applied by hand. In case of poor adhesion, the print was repeated when object surface was covered in blue tape and glue stick applied. The printed square was removed and the material surface inspected for degradation to test for re-usability.

Results

Table B.1 provides details for the 14 tests. The results show good adhesion when printing on objects made of PLA, PVC, Medium Density Fibreboard (MDF), high density foam, medium density foam, and cardboard. Tape or glue is required when printing on ABS, wood, metal, and paper book covers. Low density foam is not recommended for replacement objects due to heat warping.

Most object surfaces exhibited little or no degradation caused by printing suggesting they can be re-used multiple times. Cardboard, high density foam, and medium density foam had noticeable marring, but maintained structural and dimensional integrity suggesting they could be re-used a few times. A single layer of blue masking tape would reduce or eliminate marring for these objects, and we found the same tape surface could be reused multiple times.

B.2.2 *Adhesion with Different Filaments*

We test printing different filament materials on top of PLA (a common material used for replacement objects) and blue masking tape (which can be applied to any object). A 1 mm thick (5 layer) 1.8*cm* square and 3 mm of support (15 layers) were printed on a PLA calibration cube using PLA, PETG, ABS, and TPU.

Results

Figure B.2 shows results. TPU, PETG, and PLA have strong adhesion. For ABS, adding tape or glue is recommended. In principle, covering the top of an object with tape or glue should enable printing with most types of filament (e.g. soluble PVA, Nylon).

B.2.3 *Object Measurement Precision*

Printed holders ensure that replacement objects are securely held in position. Fit depends on the size of the objects and the holders, so objects need to be measured with a reasonable level of precision. Similarly, objects should have

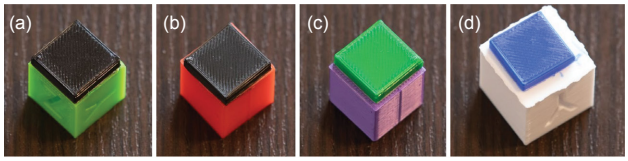


Figure B.2: Adhesion test: (a) TPU; (b) ABS; (c) PETG; (d) PLA.

the correct height to be able to print on top of them reliably. We adjusted the "measured size" of a 4x4 LEGO block to cause deliberate "measurement errors" and attempted inserting the block into the resulting printed holder and then attempted to print on top of the block. The error was increased in steps of 0.05 mm at a time.

Results

We found that a ±0.25 mm error range around the true width and length allowed for a secure fit. If smaller, the object can no longer fit inside the printed holder. If larger, insertion will no longer have a noticeable snap. Eventually, lightweight/tall objects may move or start to tip. Only one dimension has to be in the 0.5 wide mm "snapping range" for the tactile insertion feedback. If the holder is overall too large, or was accidentally knocked off the build plate, it may still be used as a positioning guideline, while double sided tape ensures that the replacement object stays in the correct position. For the height of the object, printing was reliable between -0.2 mm to +0.3 mm error compared to the actual height of the object. Below this range, a collision may be detected by the printer, or the nozzle may dig into material that can melt. At +0.35 mm, printing succeeded but adhesion started to become spotty, and for larger errors adhesion will eventually fail fully. This range may depend slightly on printing layer height, which was set to 0.2 mm for the tests. Based on this range of acceptable heights, we were able to print on solid objects with noticeably angled top surfaces, such as a rough-cut piece of wood.

Flexible materials (such as cardboard boxes or foam plates) may allow successful printing even for very imprecise measurements (e.g. measurement errors greater than 1mm), such as when squeezing the material with the calipers during measurement. As the print head is lowered onto the support replacement object from above, the print head can compress such objects without triggering collision detection. Many slicers enable this movement pattern, as they lift the print head during retractions (i.e., when switching between printing locations). We found that top surfaces with extreme deviations that exceed the lifting height (up to several millimetres) still do not cause issues if the deformations peak near the centre and the surface is sufficiently pliable. This is because the print head gradually pushes the object surface down as it prints, essentially undoing deformations temporarily.

B.2.4 *Stability with Build Plate Movement*

As replacement objects are located on the build plate, moving the build plate could potentially tip replacement objects over. We placed an otherwise unsupported 4x4 stack of LEGO into a printed holder. The stack was more than twice as high as it was wide. The build plate was then moved back and forth at the maximum speed allowed by the printer firmware. The length of each movement was reduced consecutively each time after switching the movement direction.

Results

The stack remained in its holder without tipping. Figure B.3 shows the test setup.

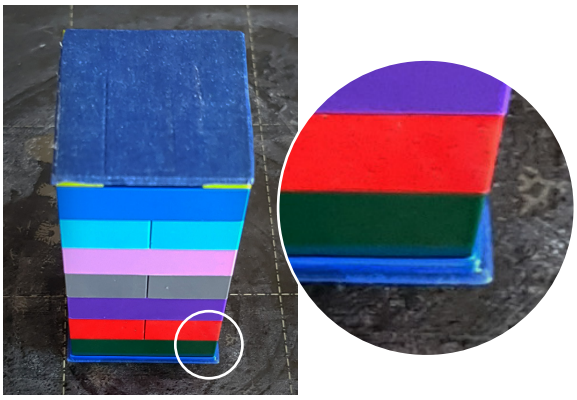


Figure B.3: Stability test setup: a tall 4x4 stack of bricks held in place by a printed holder.

B.2.5 *Finish Quality after Long Pauses*

If no user is available to perform an interaction, the printer may idle for extended periods of time. We compare the same print job when a pause is handled immediately and when a pause is handled after an 8-hour idle period.

Results

Figure B.4 shows results. For standard PLA printing processes, rapid and extended pauses (8 hours) seem to have no noticeable effect on print quality. For processes that require for example a heated printing chamber, pausing parameters may need to be customized, for example to ensure the temperature equalizes within the chamber after reheating when continuing after an extended pause. Heating could be maintained for a while during the pause to accommodate shorter pauses more readily. Any change in geometry may alter the printing direction of outer perimeters, which may be visible at certain reflection angles.

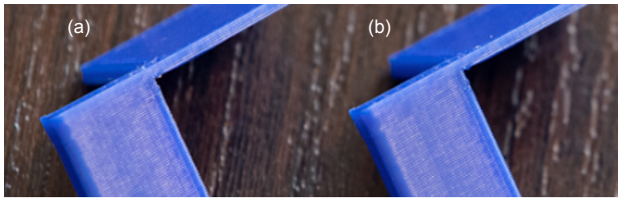


Figure B.4: Finish quality after long pauses: (a) resuming printing immediately; (b) resuming printing after 8 hours.

B.2.6 *Compatibility with Support Structure Types*

The approach should work with any type of support structure, as long as printing on top of the chosen support replacement objects is possible. The modified slicer currently only supports generating one type of support structure. We tested extreme slicing settings of this type of support generation to gauge whether the method is affected by different types of support. We printed on top of support replacement objects with varied support slicing settings: no support infill (i.e., printing support outlines only); 100% support infill (i.e., printing solid supports); and thin support towers of 1 to 3 mm width to emulate tree-style supports.

Results

Different support fill methods showed no impact on adhesion. When testing the 1 to 3 mm support towers, all 37 adhered strongly to the support replacement object.

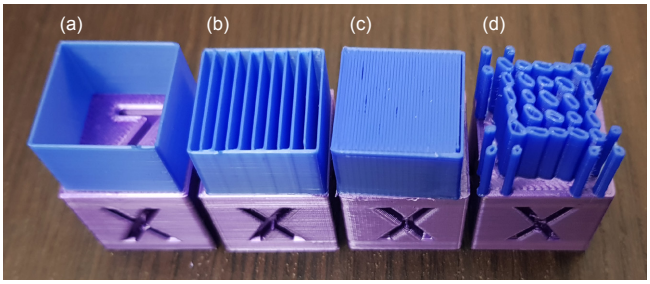


Figure B.5: Adhesion using different support settings: (a) no support infill; (b) default support infill; (c) 100% support infill; (d) thin tree-style supports.

B.3 *Comparison with Other Support Generation Methods*

Table B.2: *Support Material* in grams used by different slicer support generation methods and by our system.

Model	Prusa Grid	Prusa Snug	Prusa Organic	Cura Normal	Cura Tree	Kiri Snug	Min. Replaced	Max. Replaced
Angle Bracket	3.4	3.1	1.9	3.7	1.8	4.3	1.0	0.1
Gymnast	9.6	6.3	7.0	8.4	5.2	5.9	3.0	0.4
Cheburashka	3.5	1.9	3.7	2.4	2.3	2.0	1.1	0.3
Fiducial Marker	15.2	9.8	7.0	13.7	4.9	18.1	12.4	3.7
Stanford Bunny	21.1	14.4	13.3	14.5	8.3	13.8	10.5	3.8
Armadillo	20.5	15.6	15.1	20.3	11.9	24.2	12.4	7.6
Simple Bridge	3.9	3.7	2.4	4.1	2.6	5.1	2.7	1.6
Enterprise	21.7	17.7	17.5	18.1	14.4	23.7	15.8	7.9
Nefertiti	96.8	70.8	40.7	33.8	23.3	20.9	13.9	7.0
Beast	30.2	22.2	16.4	27.4	11.6	38.0	27.1	13.3
Cow	13.4	10.1	9.2	11.2	6.3	12.7	9.2	4.7
Angled Cylinder	19.0	17.8	11.1	9.5	4.4	13.3	10.6	5.1
Arch Bridge	55.8	44.6	47.3	51.7	-	87.0	57.9	41.2
Rocker Arm	7.7	6.1	7.2	6.0	4.3	8.3	5.5	4.0
Bike Helmet	66.2	50.9	47.7	54.3	32.5	60.0	42.3	35.5

Table B.3: *Support Printing Time* in minutes used by different slicer support generation methods and by our system.

Model	Prusa Grid	Prusa Snug	Prusa Organic	Cura Normal	Cura Tree	Kiri Snug	Min. Replaced	Max. Replaced
Angle Bracket	9.0	8.0	21.0	16.0	15.0	57.3	21.4	1.7
Gymnast	73.0	45.0	60.0	66.0	37.0	162.5	82.6	7.4
Cheburashka	30.0	17.0	39.0	21.0	17.0	53.4	28.2	7.5
Fiducial Marker	104.0	58.0	94.0	65.0	27.0	324.4	224.7	79.0
Stanford Bunny	214.0	152.0	166.0	104.0	65.0	264.3	196.6	75.6
Armadillo	199.0	161.0	170.0	124.0	86.0	543.4	254.8	174.6
Simple Bridge	17.0	16.0	37.0	18.0	20.0	65.0	50.1	33.9
Enterprise	183.0	143.0	267.0	106.0	119.0	352.5	255.2	130.6
Nefertiti	840.0	626.0	437.0	293.0	175.0	467.2	323.4	160.4
Beast	259.0	188.0	186.0	163.0	82.0	845.4	517.5	261.9
Cow	123.0	93.0	129.0	71.0	57.0	255.1	189.2	105.4
Angled Cylinder	86.0	89.0	134.0	51.0	32.0	205.2	172.0	108.4
Arch Bridge	424.0	343.0	704.0	260.0	-	1610.1	938.7	751.9
Rocker Arm	71.0	55.0	113.0	144.0	153.0	124.9	92.4	71.9
Bike Helmet	628.0	492.0	621.0	346.0	258.0	1199.3	831.3	696.5

Some slicers offer a variety of support generation methods. We found the kiri:moto method we used in our submission is most similar to Prusa Snug. Prusa Organic saved an average of 30.9% material compared to Prusa Grid,

but only saved 7.5% compared to Snug. In 4 models, Organic used 6% to 96% more material than Snug. Prusa Organic required an average of 18% more time to print compared to Snug. Cura Tree saved an average of 39.5% compared to Cura Normal, but the Arch Bridge model failed to slice with Tree supports. Some slicers offer a variety of support generation methods. We found the kiri:moto method we used in our submission is most similar to Prusa Snug. Prusa Organic saved an average of 30.9% material compared to Prusa Grid, but only saved 7.5% compared to Snug. In 4 models, Organic used 6% to 96% more material than Snug. Prusa Organic required an average of 18% more time to print compared to Snug. Cura Tree saved an average of 39.5% compared to Cura Normal, but the Arch Bridge model failed to slice with Tree supports.

APPENDIX C

We provide the custom G-code to print standing needles. The code uses relative coordinates and expects to start from the X and Y location of the needle, 0.8 mm above the previously printed layer.

```
G91
G1 Z-.85 E1.1 ; lowering nozzle slightly into layer below
G1 X1 Z0.05 F500 ; preparing the needle location surface
G1 Y-1 E0.05
G1 X-2.0 Y2.0 E0.05
G1 Y-2.0 E0.05
G1 X2.0 Y2.0 E0.05
G1 X-1 E0.05
G1 Y-2.0 E0.05
G1 X-1 Y1 E0.05
G1 X1 E0.05
G1 Z0.05 E0.1
G04 P100 ; heating the center of the needle location
G1 Z0.8 E0.8 F80 ; printing the needle
G1 Z1.0 E0.7 F80
G1 Z1.2 E0.6 F80
G1 Z1.4 E0.6 F80
G1 Z1 E-0.8 F120 ; retract upwards to create a sharp point
G04 P800 ; dwell to stop further stringing
G1 Z4 E-0.2 F1200 ; repeat
G04 P300
G1 Z2
G04 P100
G1 Y5 Z2 ; move sideways to remove leftover stringing
G90
```