Soft Technologies

Lea Albaugh

CMU-HCII-24-106
August 12, 2024

Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Committee
Mark D. Gross
ATLAS Institute, University of Colorado Boulder
Jessica Hammer
Human-Computer Interaction Institute, Carnegie Mellon University
Lining Yao
Mechanical Engineering Department, University of California, Berkeley
Scott E. Hudson
Human-Computer Interaction Institute, Carnegie Mellon University

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

Funding sponsors: The research reported here was supported by National Science Foundation Grants Career IIS-2047912, IIS-2017008, IIS-2118924, IIS-1718651, and Engineering Research Center Planning Grant (ERC PG) 1937031, as well as Adobe Research and the Siebel Scholars program.

Keywords: human-computer interaction, computational fabrication, softness, technical systems, contextual design, textiles, knitting, weaving, underdetermination, emergence, material grain, procedural design, improvisation, modification, composition, deployability

Abstract

This work explores ***soft technologies*** in computational fabrication: ways of creating with materials that are flexible, dynamic, and/or uncertain. Soft fabrication systems can be built to work with unusual materials, and to adapt to current and futures needs; they can be appropriate to a wide variety of contexts, including those outside of industrial and production work such as materials research labs or personal creative practice.

I develop the lens of "softness" through a combination of technical systems development and design inquiry, resulting in computational fabrication systems which explore softness at the levels of physical materials, contexts of use, and the workflows that bridge between them. In documenting the individual systems, I provide a number of supporting contributions, including techniques for producing complex mechanisms with machine knitting, demonstrations of inexpensive and easily deployable camera-based sensing for fabrication tasks, and insights from creative practitioners. Uniting the findings from these, I construct a conceptual frame and a set of system-building tactics that can be used to create flexible and adaptable computational fabrication systems, with implications for how complex materials can be used, by whom, and in what contexts.

Contents

0. Introduction

⁂ Overview **1**

1. Softness **2**
  1.1. What: [Under]defining Softness 2
  1.2. Why: Motivations and Contributions 3
  1.3. How: Methods 4

2. Fabrication, Materials, and Creativity in HCI **5**
  2.1. Computational Fabrication 5
  2.2. Materials in HCI 6
  2.3. Creativity Support 6

I. Pliable Materials

3. Background and Related Work: Soft Material Properties and Operations **9**
  3.1. Textiles 9
  3.2. Fabricating Functional Structures in Textiles 10

4. Fabrication Technique: Knitting **11**
  4.1. Knitting Machines 12

⁂ Knit Mechanisms **14**

5. Tendon Actuation **15**
  5.1. Inlay: Horizontal Tendons 15
  5.2. Yarn Carrier Tangling: Vertical Tendons 16
  5.3. Diagonal Tendons 17
  5.4. Basic Topologies: Sheets and Tubes 17
  5.5. Shaping: Short Rows and Increase/Decrease Shaping 18
  5.6. Anisotropy: Knit/Purl Texture 20
  5.7. Composition 22
  5.8. Materials Selection Guidelines 22
  5.9. Embedding Interactivity 25
  5.10. Complete Objects 26
  5.11. Summary 28

6. Spacer Fabrics **29**
  6.1. The Spacer Structure 29
  6.2. Producing Spacer Fabrics on a V-Bed Weft Knitting Machine 30
  6.3. Fabrication Parameters 30
  6.4. Resulting Spacer Fabric Characteristics 33
  6.5. Robotic Skins 37
  6.6. Sensing and Soft Switches 38
  6.7. Integration into Knitted Objects 40
  6.8. Summary 41

II. Malleable Contexts

7. Background and Related Work: Soft Situation and Forms **43**
  7.1. Personal and Site-Specific Fabrication 43
  7.2. Production *vs* Practice 44
  7.3. Computational Craft 45

7.4. Textiles and the Fiber Arts 46
7.5. Aesthetics of Computation 48

8. Fabrication Technique: Weaving 51
8.1. Looms and Jacquard Weaving 51
8.2. Production and Handweaving 53
8.3. Software Tools for Weaving 53
8.4. Weaving in Human-Computer Interaction Research 54

⁂ Personal Practice in Computational Handweaving 55

9. A Personal Jacquard Loom 56
9.1. Contemporary Handlooms 57
9.2. Technical Implementation 58
9.3. Personal Weaving Systems 63
9.4. Limitations and Future Work 67
9.5. Summary 67

10. Underdetermined Computational Handweaving 68
10.1. Design Process and Principles 69
10.2. Systems, Artifacts, and Findings 72
10.3. Technical Implementation 72
10.4. “Blobs”: Designing With Paper Cutouts 73
10.5. Slit-Scan Self-Portrait 75
10.6. “Twitch Plays Loom”: Anonymous Networked Editing 76
10.7. Discussion: Procedural Generation Under Risk 78
10.8. Limitations and Future Work 82
10.9. Summary 83

III. Liminal Workflows

11. Background and Related Work: Soft Abstractions and Tools 85
11.1. Artifact and Experience 85
11.2. Interactive Fabrication 86
11.3. Experimental, Experiential, and Hybrid Fabrication 88
11.4. Representations and Modular Workflows 90

12. Fabrication Technique: Knitting Again 94
12.1. Representations and Tools for Hand Knitting 94
12.2. Representations and Tools for Industrial Machine Knitting 96
12.3. Representations and Tools for Hybrid & Manual Machine Knitting 97
12.4. New Workflows 98

⁂ Experiential Workflows for Knitting 99

13. Augmenting a Manual Knitting Machine 100
13.1. Operating a Manual Knitting Machine 101
13.2. Implementing an Augmented Knitting Machine 105
13.3. Interface Modules 110
13.4. Improvisation by Novice Users 114
13.5. Scaffolded Learning 116
13.6. Interaction with Hybrid Processes 117
13.7. Embodied Knowledge in Manual Machine Processes 117
13.8. Discussion 118
13.9. Future Work 119
13.10. Summary 120

14. Finding a Grain in Brioche Knitting **121**
14.1. Grain 121
14.2. Grain Spaces 122
14.3. Brioche Knitting 124
14.4. System and Physical Inputs 126
14.5. Manipulation in Brioche Space 131
14.6. Grain Spaces for Novice Creators: User Study 134
14.7. Future Work 139
14.8. Summary 139

IV. Structuring Softness

15. Structuring Softness **141**
15.1. Aspects of Softness 142
15.2. Approaches to Softness 143
15.3. Soft Values 144
15.4. Soft Techniques 146
15.5. What softness is not: a note about “AI” 151

16. Softer Technologies: Future Work **152**
16.1. Sustainable and Local Material Futures 152
16.2. Material Learning 152
16.3. Fuzzy Computation 153

17. Conclusion **154**

***

References **156**
Acknowledgements **189**
Colophon **191**

0.

Introduction

Overview

This dissertation document is arranged in five parts. First, **in this Introduction** (Part 0), I introduce and motivate the theme of ***soft technologies*** in computational fabrication—ways of creating with materials that are flexible, dynamic, and/or uncertain—and I establish background on material technologies in human-computer interaction, including computational fabrication, materials research, and creativity support.

In the next three Parts, I document and discuss a series of computational fabrication systems that exemplify multivalent aspects of softness. These Parts are structured around the concept of a "material technology" as a three-level stack which encompasses, from low-level to high:

1. The physical **properties** of a material and the **operations** that can shape them
2. Emergent **abstractions** for reasoning about those properties and operations and **workflows** for manipulating those abstractions
3. The material's **contexts** of use and **aesthetics** guiding and resulting from those contexts, such as recurring forms and cultural associations

Each of these levels affects the others; for example, recurring forms and aesthetics arise out of both the underlying material properties and personal, historical, and cultural workflows. In organizing the document this way, I discuss aspects of "softness"—some more literal, like the elasticity that enables a plush robotic gripper, and some more metaphorical, like an improvisational workflow—as distinct but interconnected. I introduce these topics non-linearly: first, Part I characterizes low-level material properties; next, Part II is about high-level contexts; finally, after establishing these anchors, Part III describes the workflows that bridge between them.

Within each of these three Parts, I begin with a chapter that establishes the scope of that Part and discusses specific related work. The next chapter provides technical background on the fabrication techniques included in that Part (knitting in Part I and Part III, and weaving in Part II). Following that, a brief summary page introduces the two technical systems to be presented in that Part, and then I present a detailed account of each system in its own chapter.

The work presented in this dissertation focuses on textiles: knitted, woven, or otherwise structured fibers. The complex soft structures of textiles, and the ubiquity of their production and use, are particularly well-suited for exploring softness in technical systems. However, the principles hold for other computational material technologies as well. **In the last part**, Part IV, I discuss generalized principles of softness, their applicability in computational fabrication more broadly, and technical guidelines for achieving them in fabrication support tools.

1. Softness

“Soft” is hard. We tend to focus our established “high” technologies on solid, dependable stuff like rigid body mechanics and quantitative data. Materials that are malleable, not easily simulatable, or that have fabrication methodologies that diverged millenia ago from the rest of industrial manufacturing are often difficult to integrate into engineering pipelines [244]; it’s easier to dismiss them as a nuisance. Workflows that are portable or personal, or that operate on indeterminate or evolving specifications, are difficult to generalize; it’s easier to dismiss these as too niche to bother with. Contextual aspects of technology, like cultural aesthetics and social meaning, are subjective and difficult to analyze; it is *much* easier to dismiss these as out of scope for technical research.

However, in balking at malleable, indeterminate, and/or emergent technologies, we neglect a tremendous amount of creative potential. Soft materials, especially textiles, are frankly unavoidable—we spend our lives surrounded by them—and they can be fabricated in incredibly precise ways with a wide variety of purposes. Soft systems can be flexibly adapted to specific users and uses, such as for individual accessibility or to work with low-cost or sustainable materials. Emergent hybridity between materials and computational processes can provide new abstractions and new experiences of fabrication, leading to exploration and invention.

In this dissertation, I aim to highlight the unique strengths of softness, such as adaptability and emergent complexity, and to show how they can be usefully deployed to support creativity and innovation in computational fabrication. In providing exemplars from three separate but intertwined levels of “material technology,” I demonstrate how to recognize and reason about softness throughout how we create with materials and computation.

1.1. What: [Under]defining Softness

This work agglomerates multiple meanings of “softness” centering around adaptability, emergence, indeterminacy, and hybridity. I chose the term “soft” to encompass these shifting meanings because it is already intriguingly overloaded, with differing implications at each of the three levels of material technology I describe. On a literal, physical level, the word “softness” can mean pliable, like an elastomer, or “soft to the touch” like a kitten. Metaphorically, it can mean indistinct, like a soft focus; or blendable, like a soft melody; or gentle or calm, like a soft breeze. It even sometimes means “subjective,” as in the term “soft skills.”

By assembling these meanings into one lens, albeit a multivalent one, I encourage study of how these meanings relate to each other and can be usefully intertwined. For example, a challenge of softness at the material property level, such as the difficulty of precisely simulating an output, might be met with softness at a workflow level by generating composably parametric test swatches. Parts I-III of this document demonstrate different aspects of softness. In Part IV, I reflect on these various aspects as loci of softness that recur across the work, and apply them to generate useful technical approaches to future soft systems.

1.2. Why: Motivations and Contributions

Softness is ubiquitous, but underexplored in technical human-computer interaction research. Soft materials surround us, from our clothes and furnishings to medical and safety devices, but the disconnect between rigid-body engineering practices and our millenia of human expertise in textile fabrication means that textiles are less frequently examined as "serious," engineerable materials. Sources of uncertainty and contexts that reward adaptability are similarly abundant: artisanal processes have always incorporated some level of indeterminacy (as discussed in Section 7.2), and flexibility is a key component of enabling access across financial contexts, physical abilities, and differing environments. Working improvisationally, often with irregular materials and unusual locations, is critical for mending and maintenance, which are increasingly understood to be cornerstones of sustainable material practice. However, these are similarly disconnected from the main narrative of computational fabrication as a kind of production-oriented micro-manufacturing which prioritizes predictability and efficiency.

This does a disservice to both the material possibilities of softness and to the study of computation's role in creative practice. While the predominant drive is to use it as a regularizing force, computation itself is tremendously malleable. Similarly, while predictable and efficient production can a worthwhile endeavor, it is just one of many modes available to computational creativity overall and computational fabrication specifically. Computation can blend, distort, filter, and compile data and operations in ways that enable not just specific material outcomes but entirely different, uniquely computational ways of reasoning about material processes that would be intractable otherwise. Over-focusing on computational control as a source of precision and repeatability under-serves other roles of computation. For example, a computational system can track and analyze in-progress outcomes, offering interpretation and suggesting varying possibilities and helping interpret outcomes; this role of scaffolding creative reflection might be especially desirable for artists and inventors whose goals are not quite pre-determined.

Understanding softness in its own right, not as something untrustworthy, fickle, or simply unknowable, but as something with its own logics (albeit malleable and emergent ones) can re-connect the genuine breadth of computational approaches to the powerful complexity of soft materials and contexts. In this work, I show how softness can underpin material innovation and creative practice, from soft textile materials to parametric materiality as the basis of design tools. Along the way, I demonstrate the power of machine knitting for complex materials, investigate relationships between fiber arts and new media practices, enable access to textile fabrication through inexpensive new and augmented machines, and introduce the idea of a parametric material "grain space" as an approach to building creative material design tools. By characterizing where softness occurs in computational fabrication systems, and by exploring ways to channel and scope softness, the work in this dissertation establishes a unified lens for understanding structured underdetermination in computational fabrication. This has implications for how creators, from artists to inventors, can design and fabricate complex materials, ultimately broadening what can be made, and by whom.

1.3. How: Methods

The work in this dissertation combines aspects of ***design*** research, including inquiry into the contexts, abstractions, and aesthetics of computational fabrication systems, with ***technical systems*** research including building real, full-stack hardware/software systems incorporating complex and cutting-edge algorithms and machines.

As such, the methods vary. In Part I, I focus on low-level material properties, which I characterize using approaches from morphing matter and metamaterials framings. In Part II, I discuss situated and contextual aspects of softness using autoethnographic and research through design methods. In Part III, I specify system abstractions and incorporate participant studies.

2. Fabrication, Materials, and Creativity in HCI

Because of the broad-spanning nature of this dissertation, I have grouped deeper discussions of related work into each of the sub-parts of this document. However, the entire work is in conversation with fabrication, materials, and creativity research in human-computer interaction, so I give a background summary of these areas in this chapter.

2.1. Computational Fabrication

Historically, a range of terms has been used to describe the group of related processes that includes loom-controlled weaving, Computer-Numerically Controlled (CNC) machining such as with a mill or lathe, and computer-controlled additive methods like selective laser sintering or stereolithography. “Computer Aided Manufacturing” (CAM) is used in conjunction with “Computer Aided Design” (CAD) to describe a pipeline from engineering concept to fabricated output; “CAM” is particularly used to describe subtractive (machining) tasks where the toolpath generation requires expert human guidance for best results. Conversely, computer-controlled additive manufacturing is often called “3D printing,” a term that often connotes a straightforward translation between an on-screen representation and a printed object. “Advanced manufacturing” emphasizes the use of computer control for precise technical outcomes (such as nanofabrication); “rapid prototyping” recognizes that many of these processes have low production overhead, making them particularly suitable for prototyping and one-off production. The term “digital fabrication” is a popular umbrella term for these processes, particularly in makerspace and academic contexts.

I use the term “computational fabrication” to emphasize the unique aspects computation can bring, such as algorithmic and generative design, and to include contexts in addition to “manufacturing” and “prototyping,” such as the production of artifacts for research [112] or personal use [28].

When used in personal projects, computational fabrication enables one-off and context-specific outcomes. For example, when computational fabrication is used to repair or modify an existing object [419], a creator might go beyond restoring an object’s original functionality to create something uniquely suited to their needs.

2.1.1. Experiential Fabrication

Research in *interactive fabrication* blends fabrication engineering, sensing, and interaction design, to expand the role of a fabrication machine from being the endpoint of a creative process to being itself a site for creativity, whether for rapid prototyping [254, 297] or a more open-ended material exploration [379, 409].

In my emphasis on flexibility, multiplicity, and experience, I draw inspiration from “digital craftsmachineship” [19, 287], which emphasizes situational expertise, “computational craft” which emphasize hand and body engagement with materials [35, 158], “hybrid fabrication” systems which offer intertwining roles for human and machine contributions to an outcome [195, 377, 436], and experiential and experimental fabrication which centers fabrication processes as a site for reflection on the relationship between "makers, materials, and environments" [76]. I describe these approaches and their relation to my work in more detail in Chapter 11.

2.2. Materials in HCI

While it is often directly intertwined with fabrication research, there is an increasingly distinct separate thread of research inquiry in HCI around materials themselves as a locus of computation. Morphing Matter and Metamaterials work studies complex material properties which can be manipulated for programmable, interactive effects in non-electronic objects. Morphing Matter research combines computational modelling, material characterization, and interaction design to produce active materials that can controllably change shape, properties, or functionalities [292]; for example, such materials can be engineered to respond to stimuli including temperature [18], pH [167], moisture [219], pressure [403], and magnetic fields [431]; responses can include changes in shape, texture [203], permeability [285], or stiffness [92]. Metamaterials research specifically considers the structure of a material, in combination with its substance, as a way to engineer properties like texture [154] and strain response [371]. Under this framing, a material can be a complex mechanism [153]. I explore this idea in Chapter 6. Both areas typically make use of computational fabrication methods to produce the designed materials; especially in Morphing Matter work, the fabrication equipment may be purpose-built to handle particular material processes, such as computer-controlled heat-sealing [284], dough-stamping [373], and fiber-drawing [96]. The ability to adapt to experimental material processes is a motivation of my work.

2.2.1. Textiles

Textiles in particular are prized in human-computer interaction research for their suitability for on-body interaction. Materials research in this area has explored engineering and design of actuable fibers [96, 97, 174] and textile structures as a basis for complexly engineered robotic mechanisms [334]. Additionally, fabric-like structures have been explored as ways to obtain flexible and breathable materials from 3D printing [29, 95, 322, 372].

Many textile-related works in HCI specifically center "e-textiles," particularly either for on-body interaction [366] or as a topic for teaching interdisciplinary K-12 workshops [163, 165, 344]. Although I include e-textile examples in every Part of this dissertation, e-textiles, narrowly defined — physically integrating electronic circuitry into fabric — are not central to my work. However, many of the accompanying values and methodologies are. I discuss textile structures for engineered functionality [274] in Part I, incorporating cultures of textile artisanship [71] in Part II, and site-specificity, reuse, and augmentation [301] in Part II and Part III.

2.3. Creativity Support

"Creativity support tools" are "any tool that can be used by people in the open-ended creation of new artifacts," and typically more specifically "software applications that are used to create digital artifacts or are used as part of the process of working toward the completion of an artifact" [55]. While this definition is deliberately imprecise, creativity support research often specifically targets novice (or simply "nonexpert") users. Such computational tools can embed expert knowledge about a particular type of creative output — for example, art made with polarization filters [348], or crocheted plush animals [151]—allowing a user to produce complex outputs that they might not otherwise achieve. Creativity support may seek to "democratize" [18] a creative process, enabling relative novices feel agency over their creation. For example, highly-scaffolded creativity support tools can center "the fast, confident, and pleasurable exploration of a possibility space" (e.g. character customization interfaces in videogames) engendering "feelings of pride, ownership, and creativity" [60]. While I do not share the specific narrow focus of novice users, I highlight adaptability over time —the possibility of changing to suit a specific user's skill set and desires—as an advantage of soft systems. Additionally, a grain space (as defined in Chapter 14) is a form of embedded material expertise.

Some creativity support tools are positioned as collaborators; others may be providers of a creative experience, or spaces to explore [55]. I discuss several roles for systems—as mediators (Chapter 10), as guides (Chapter 13), and as curated spaces (Chapter 14).

2.3.1. Physically-Situated Creativity

The use of curational and tactile techniques for creativity—investigating one's own context, capturing site-specific details, and applying hands-on manipulations—is well-explored in the visual arts, for example in the practice of bricolage [80]. Within human-computer interaction, researchers have explored systems for bringing physical inputs into digital contexts in seamless and playful ways [333], as an engine of inspiration [3, 349] or to provoke the designer's engagement with their own surroundings [75]. I see this situated and contextual creative practice as underexplored in fabrication, despite its tactility being particularly suited to the often messy and analog low-level complications of physical fabrication. I incorporate physical inputs throughout Part II and Part III, particularly in Chapter 9, Chapter 13, and Chapter 14.

I.

Pliable Materials

3. Background and Related Work: Soft Material Properties and Operations

In english, we use the word "hard" to mean both "rigid" and "difficult," but many engineers will tell you that soft objects can be appreciably more difficult to work with than rigid ones [244]. **Pliable materials** can be challenging to model and predict, and fabrication techniques for them have evolved separately from the shaping, cutting, extruding, and printing processes that are most well-known in mechanical engineering. Nonetheless, soft materials surround us.

Physical softness is prized for comfort, adaptability, and suitability to be in, on, and near our bodies. Soft materials can absorb or redirect collisions or acoustic energy, and can be fashioned into *compliant mechanisms* that use engineered flexibility to reduce part complexity, store forces, and "accomplish complex tasks with very few parts" [143]. Within HCI, soft interfaces have been developed for locomotion [260], manipulation [259], display [426] in contexts including prototyping [398], biomonitoring [432] and for generating different physical affordances [109]; the deformability of soft objects particularly supports haptic modalities in contexts such as wearables [56] and mobile devices [109].

3.1. Textiles

Throughout this dissertation but particularly in this Part ("Pliable Materials") on soft material properties, I focus on **textiles** as a material known for complexity and difficult softness. *Textile* describes materials made of *fibers*; fibers are simply materials which are much longer than they are wide, and are implicitly flexible [334]. (I'll use "fabric" to refer to a textile sheet material, such as "a yard of velvet," as distinct from a complete textile object, though the word "fabric" can also refer to soft, sheet-shaped materials that aren't fiber-based, such as a rubber sheet.)

The material structures of fabric are literally complex (Latin "*com-* together + *plexus* plaited" [286]), and the range of possible material characteristics is broad. There exist numerous ways of obtaining fibers (most commonly: harvesting them from plants and animals, or extruding them from dissolved or molten polymer compounds), optionally spinning and plying several fibers into *yarns*, and forming those fibers or yarns into flat or shaped textile surfaces with processes such as weaving, felting, heat-bonding, knitting, and sewing. The specific arrangement of fibers into a yarn, then of yarns into a fabric, can determine many material properties, including: optical properties like light transmission and reflectivity; tactile properties like surface texture and drape [312]; functional properties like stress/strain responses and active fiber routing (as in e-textiles [100]); and even topological properties, with, e.g., "double-cloth" structures allowing for integrated tubes and pockets [73].

A textile is inherently hierarchical: the micro-scale molecular properties and shape of a fiber affect its restitution and friction; these affect the density, twist, and other meso-scale characteristics of a yarn; these in turn affect the overall anisotropic stretch, tensile strength, and surface qualities of a fabric.

It can thus be helpful to conceptualize textiles as *metamaterial*: an aggregate material where intentionally engineered sub-structures influence higher-level properties. Metamaterials have been an active topic of research in optics [421], acoustics [236], antenna design [116], and mechanics [371]. One popular style of metamaterial fabrication uses repeated, tunable cells. By computationally selecting and positioning cell varieties, properties such as weight/strength [31] or flexibility [290]) can be optimized, either throughout the whole material or at specific locations within it. Cell-based transmission of forces can be used to create mechanisms, especially when supported with computational design tools [153–156]. One particularly relevant metamaterial structure is "bristles": hair-like structures that have a high aspect ratio suited to transmit and amplify vibrations [281] or as tactile or high-friction surfaces [192, 282]. Other structures that can achieve a similarly biased surface friction and have been used in soft actuations include stretched kirigami patterns [315] and angled origami folds [398]. (In Chapter 6, I explore bristles for both velcro-like applications and directionally biased actuations.) Approaching textiles as a metamaterial underscores their incredible functional complexity; more critically, it points toward ways of reasoning about their emergent phenomena without having to resort to fiber-level simulation.

3.2. Fabricating Functional Structures in Textiles

Within textiles, functional material characteristics can be produced in a number of ways. For example, quilting is a process which consists of layering padding between cloth and sewing through the layers to attach them; the density of stitching has been shown to control the thickness of the resulting structure [115]. In industry, 3D weaving—a generalization of double-cloth weaving—is used to produce composite reinforcement [82, 202]; recently, researchers and designers demonstrated that these tools can be extended to make 3D-woven shoes [126] with varying functional zones. Further, yarn can be felted [146], fabric sheets laminated [298], or thermoplastic electro-spun and folded [318] into objects with embedded functionality.

Electronic "e-textile" fabrics can incorporate conductive and resistive elements through hand- and machine-embroidery [84, 110, 222, 308], dyeing [138] and printing [393], weaving [100, 370], braiding [274], knitting [182, 283, 303], and crochet [74].

Knitting, the fabrication technology used in Chapter 5 and Chapter 6, is particularly suited to producing complex textile structures [376]. Knitting machines—a mature fabrication technology that has existed for centuries and has been computerized for decades—can create a wide range of 3D topologies, either through careful hand-design [391]; with the aid of primitive-based design systems [171, 233]; or by directly converting 3D models [262, 263, 304]. (I will more thoroughly discuss tools and workflows for machine knitting in Chapter 12.)

4. Fabrication Technique: Knitting

Knitting is a way of forming a surface out of rows and columns of loops of yarn. Unlike a woven fabric with its separate warp and weft axes (as I will discuss in Chapter 8), a simple knit structure can be formed from a single continuous length of yarn, and it is inherently stretchy because of the meandering yarn path. The oldest extant knit swatches are estimated to have been produced in the fourth or fifth century CE in Egypt, and the first machine for knitting (called a "knitting frame" or "stocking frame") was developed between 1589 and 1600 in England [332]. Today, knitting is associated with the thicker, cozy fabrics of sweaters and socks, but it is also the main fabric of t-shirts and sporting and athleisure wear, and is increasingly found in highly technical applications such as medical stents.

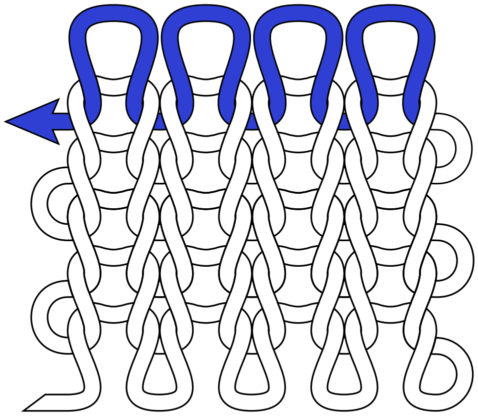


Figure 4.1 *A basic knit swatch: rows of loops connect to other loops in columns. The structure is formed under gravity, progressing from the bottom to the top.*

Whether by hand or by machine, most knit fabrics are either formed by working back and forth in a flat plane as in Figure 4.1, or in a spiral to form a tube. Each loop in a column is formed by pulling yarn through the previous loop in the column, either from back to front (a proper "knit" in hand-knitting parlance) or from front to back (a "purl"). Because each loop has a tendency to bend out of the plane of the fabric in the direction it was pulled, mixing knits and purls within a sheet of knitting can give rise to stretchy, curled, and bumpy surfaces. Along with combination of loop directions, knit loops can be overlapped, split, and transposed, perturbing the underlying grid of loops; these grid variations provide a rich variety of functionality and embellishment—affecting aggregate properties such as elasticity, opacity, thickness/stiffness, and visual aesthetics—and are often a primary locus of creativity for hand knitters [43, 365]. At the stitch level, knit fabrics can integrate a wide range of surface textures with different mechanical properties such as auxetic behavior [105], which can in turn augment the functionality of designed objects [132]. Additionally, changes to the grid can shape an overall fabric from a plain rectangle or tube to a fully-knit complex topology like a sweater or a plush animal [15, 233].

In this work, I follow existing knitting scholarship in distinguishing overall *shaping* from *textures* within a fabric [263], but these two kinds of manipulations are not necessarily fully separable: for example, mergers between adjacent columns of loops, which may be required for net shape changes, also create distinct visual artifacts [173].

4.1. Knitting Machines

In this work, I am specifically discussing *weft knitting*, which is one of the two main kinds of machine knitting. It produces textile structures which are conceptually of the same kind as hand knitting, but typically at a different scale, often at a higher complexity, and therefore often with a different aesthetic result. As a fabrication tool, knitting machines have been used to generate databases of knit material properties [132] to underpin material-forward ways of designing and using knit fabrics, and to create complex technical materials for wearable [178, 213] and/or robotic contexts [9, 179, 376].

I present here a simplified description of the operation of flat-bed weft knitting machines; in-depth coverage of this topic can be found in Spencer [362], with summaries for HCI and Graphics audiences in Kaspar [170], Narayanan [261], and McCann et al [233].

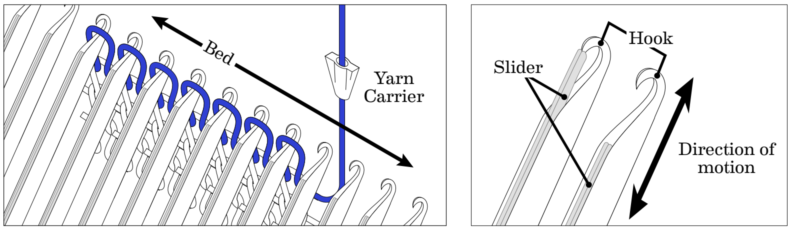


Figure 4.2 *Knitting machines have "beds" of needles. Unlike hand-knitting needles, machine needles are hook-shaped. The hook can be closed off by a secondary component; in this case, it is a independently-actuable slider.*

A flat-bed knitting machine forms a knitted structure using rows (*beds*) of parallel needles (Figure 4.2). These *slide* needles have two major parts each: a hook, which holds the topmost stitch or stitches in a column, and a slider, which can be independently actuated to close the hook. The machine additionally has a number of *yarn carriers*, through which yarn flows after passing through a tensioning apparatus. The yarn carriers are synchronized to the needle action to provide yarn to new stitches being formed. Each needle can be actuated to perform a *knit* operation, Figure 4.3: reach forward, grab yarn from a yarn carrier and form it into a new loop, and pull the new loop from back to front through the previously-held loop. (This same sequence of motions would *drop* the previous loop if no yarn carrier were involved.)

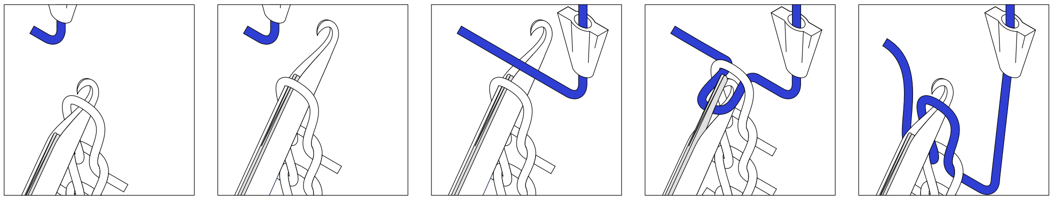


Figure 4.3 *The knit operation: 1) before the operation; 2) the needle and slider move up, and any loops in the hook slide down past the slider; 3) the yarn carrier moves past the needle, laying yarn into the hook; 4) the needle moves back down while the slider stays up, closing the hook and allowing any previous loops to fall off the tip of the hook; 5) after the operation, the new loop is held in the hook.*

Needles can also perform a *tuck* operation (Figure 4.4), which holds the new yarn in the needle without pulling the existing loop through it.

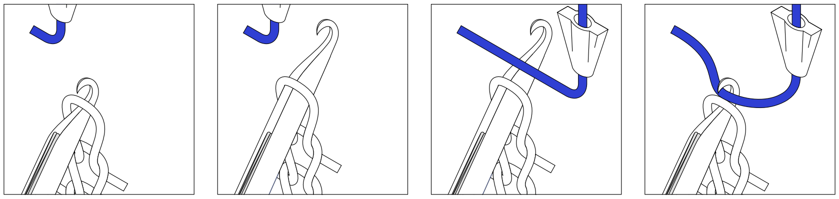


Figure 4.4 *The tuck operation: 1) before the operation; 2) the needle slides forward, but the slider stays down; 3) yarn is laid into the hook; 4) the needle slides back down, having captured the new yarn without dropping any existing stitches.*

A *v-bed* knitting machine has two beds of needles that meet in an inverted 'v' shape, shown in Figure 4.5. A single bed of needles can produce a flat sheet of knitting, with the loops all pulled in the same direction (from back to front). With two beds, more complex structures are possible. Because the needles are arranged facing each other, both loop pull directions (knit and purl) can be constructed. The two beds also enable making two separate faces of fabric; these faces can be connected into tubular topologies or into more elaborate multi-face structures such as spacer fabrics (Chapter 6) and brioche knitting (Chapter 14). Additionally, v-bed machines support a third needle operation: the *transfer* (Figure 4.5), in which a stitch is passed from a needle on one bed to a corresponding needle on the other bed.

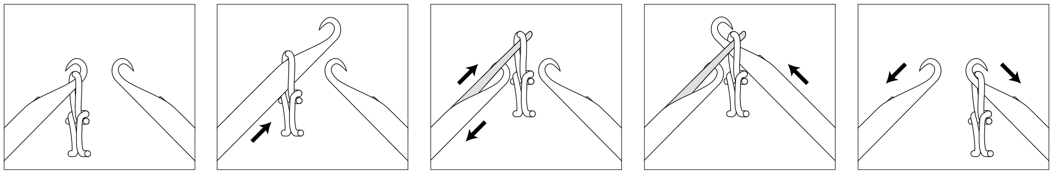


Figure 4.5 *The transfer operation: 1) before the operation; 2) the first needle slides forward to nudge its loops onto the slider; 3) the slider moves past its hook; 4) the other needle's hook grabs the loop[s]; 5) both needles return down.*

When combined with *racking*—moving the back bed laterally, changing the alignment between the two beds—transfers can move loops to nearby needles on either bed (Figure 4.6). This enables mixing knits and purls within a column of knitting as well as merging, splitting, and transposing stitches. Choreographing an optimal transfer sequence can be very complex [207, 233, 261], though it is somewhat simplified through the use of *half gauge*, in which alternating needles within a bed are allocated for knitting and transferring, respectively [233].

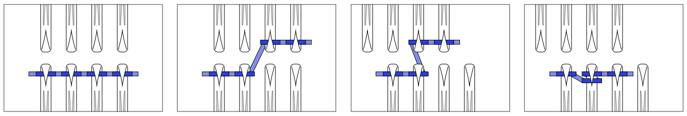


Figure 4.6 *The transfer operation being used to move a stitch: 1) stitches on the front bed; 2) the rightmost two stitches are transferred to the back bed; 3) the back bed moves leftward relative to the front bed; 4) the stitches are transferred back to the front bed, now left-ward by one of where they started.*

⁂ Knit Mechanisms

In Chapter 5 and Chapter 6, I describe two mechanisms that can be produced with v-bed weft knitting. Each relies on a combination of low-level material properties—friction and restitution of fiber materials like yarn, polyfill stuffing, nylon monofilament—and low-level machine-knitting operations like the relative positions of yarn carriers and the sequence of stitch formation. In Chapter 5, I identify a method for inlaying actuable tendons into a knit surface directly within the knitting process and discuss how sub-structural variations available to machine-knitting, such as overall surface topology and knit/purl texture, can produce a variety of actuation effects. In Chapter 6, I describe how complex "spacer fabrics" can be made on a weft knitting machine, and characterize the complex material properties of such fabrics.

All knitting in Part I was done on a Shima Seiki "Wholegarment" SWG091N2 v-bed 15 gauge knitting machine using half gauge (as described on the previous page). Low-level instructions for the knitting machines were expressed in the Knitout format [232] and generated using a set of modular Javascript functions based on parameters discussed in each section.

Together, the work in this Part contributes to our understanding of weft knitting as a general-purpose fabrication technology for programmable **compliant structures** with applications ranging from robotics to wearable and architectural functional fabric.

5. Tendon Actuation

A "tendon" is flexible structure (such as a cable or yarn) that transmits a force along its length. In robotics, mechanisms are often explored which mimic the tendons in human hands, allowing nimble motion at a distance from bulky actuators [289]. In textiles, such a structure might also be called a pull-string or a drawstring; a simple tendon might be hand-sewn into a soft surface for various motion effects [183]. Tendons are well-suited to the relatively high tensile strength of textiles.

This work explores fabricating tendon-actuable structures precisely and repeatedly using machine knitting. Notably, the tendons are inserted by the knitting machine, during the knitting process, but they are *not* knit in—that is, the tendon yarn does not participate in any knit or tuck operations, and it does not form stitches itself; rather, it is *interlaced* with the stitches by crossing in front of some stitches and behind others. This is an outcome that is not directly supported by the machine's typical production software, but it can be produced by orchestrating low-level machine operations according to two techniques: *inlay*, which can be used to produce horizontal tendons, and *yarn carrier tangling*, for vertical ones.

5.1. Inlay: Horizontal Tendons

The simplest tendon arrangement leverages an existing knitting technique, "inlay," to interlace a tendon horizontally, across a row of knitting. Inlay technique is commonly used to introduce yarns that could not be directly knit due to their stiffness or fragility; however, it is typically accomplished using special yarn feeders.

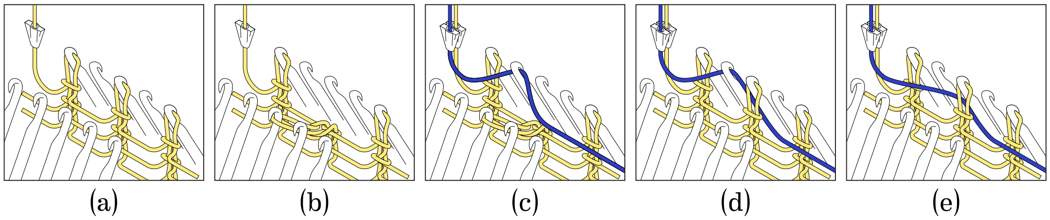


Figure 5.1 *Inlay technique for embedding horizontal tendons. a) Stitches formed on the back bed; b) Every other stitch temporarily moved to the front bed; c) Inlay yarn pulled across; d) Stitches returned to the back bed; e) Temporary tucks dropped.*

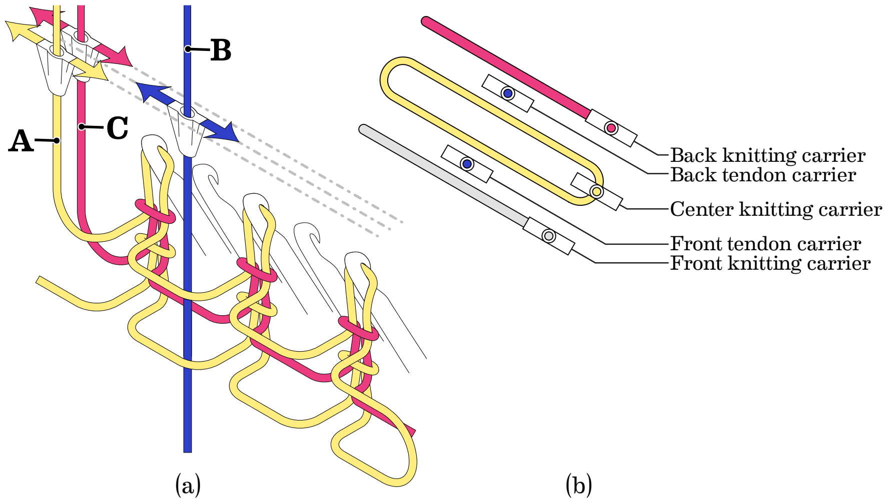
A yarn can be inlaid horizontally into a row in a way that is analogous to weaving: a subset of the stitches are moved temporarily to the other bed, the inlay yarn is pulled across, and then the displaced stitches are moved back to their main needles (Figure 5.1). In order to keep the inlay yarn tidily in place while the knit stitches are transferred back over it, it is temporarily tucked onto reserved "holding" needles at the beginning and end of its trajectory as well as at intervals between the two. On the 15 gauge machine, tucking every third holding needle is sufficient. These temporary tucks are dropped after the main knitting is transferred back into place.

5.2. Yarn Carrier Tangling: Vertical Tendons

Producing tendons in the column direction requires interlacing with the row-wise connections of the main knit structure. In conventional use of the machine, this outcome is typically considered an undesirable "tangling," which can frustratingly be caused by innocuous-seeming operations like swapping which carrier is used for a particular part of a structure. By understanding the knitting machine at a low level, it is possible to invoke this behavior deliberately for vertical tendons.

The situation arises from the arrangement of multiple yarn carriers on a knitting machine. A knitting machine can have multiple yarn carriers, typically to supply different colors or types of yarn to different parts of a knit job. These carriers move along rails as they synchronize with the needles; on nearly all machines, each carrier has its own rail. These are arranged in parallel from the closest to the front of the machine to the closest to the back of the machine. Often the carriers simply pass by each other when they need to; however, if a carrier has just supplied yarn to one of the beds, the end of the yarn is necessarily attached to that bed. That carrier can then trap a yarn from a carrier that is closer to that bed than it is.

One can thus interlace vertical tendons using three carriers arranged as in Figure 5.2: a tendon carrier ("B") which stays in place, passively supplying yarn into the resulting fabric in a vertical column, and two main knitting carriers, one of which moves in front of the tendon carrier ("A"), and one of which moves in back ("C"). When "A" knits a row, it passes in front of the tendon carrier; when "C" knits a row, it passes behind; by alternating carriers for the main knitting, rows can weave around the tendon.



(a) (b)

Figure 5.2 *a) "Tangling" technique for embedding vertical tendons. When carriers must pass each other going right or left, each can only move along on its own rail. b) Birds-eye schematic of the relative positions of the five carriers required to produce a tendon on each of two faces of fabric.*

Figure 5.2(a) shows a single face of fabric being produced on the back bed. To extend this technique to make it suitable for producing tubes, on both beds, the other face of the tube must also be knit entirely from the carrier that does not entangle with the tendon. For example, in the case when the tendon runs up the front of a tube, the back face of the tube must be knit with the back carrier. Therefore, if a tendon is called for on each of the front and back faces of a tube, another knitting carrier must be added that is positioned between the front and back tendon carriers (Figure 5.2(b)), making for three knitting carriers and two tendon carriers. A machine with ten carriers can thus produce eight vertical tendons in the same face, or seven vertical tendons across two faces.

5.3. Diagonal Tendons

Diagonal interlacements can be accomplished with either the yarn carrier tangling or inlay technique by applying the methods described above in a stair-step fashion—working for a small distance, then moving in the orthogonal direction for a small amount, then returning to application of the technique. For example, to use the inlay technique, the tendon can be interlaced with just one or two stitches per row, before moving up a row to interlace one or two more stitches, etc.

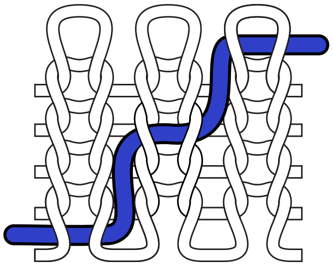


Figure 5.3 *Diagonal tendon.*

5.4. Basic Topologies: Sheets and Tubes

As described in Section 4.1, knitting can be arranged into flat sheets, or tubes composed of a front and back face, connected either at the sides by a continuous spiral of yarn, or at top and bottom by a yarn path that zigzags between the faces (Figure 5.4). For the purposes of this chapter—creating tendon-driven soft actuation within knitting—tubes are notable because they can contain elements that provide restoring force after activation, such as stuffing or strips of springy material.

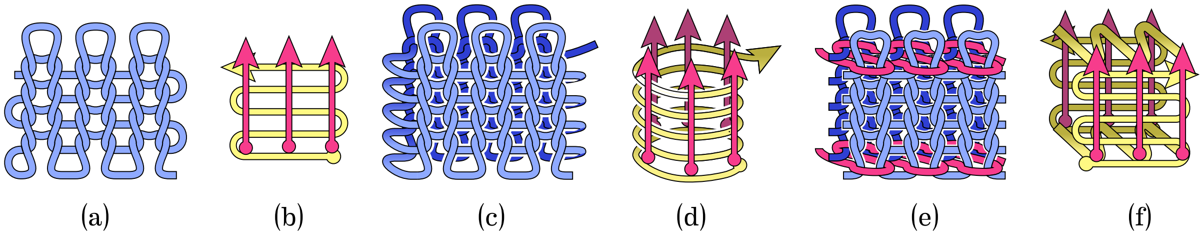


Figure 5.4 *a) A sheet. b) A simplified representation of the row and column connections in the sheet. c) A vertical tube of knitting. The lighter-colored front face could be formed on the front bed of the knitting machine, and the darker back face on the back bed. d) Rows and columns in the vertical tube. e) A horizontal tube, joined at top and bottom by a zig-zagged yarn path. f) Rows and columns in the horizontal tube.*

Per Section 5.2, tendons can be integrated into both the front and back face of the tube for bending in each direction, as seen in Figure 5.5(a). A single tendon can also interlace with both faces sequentially, producing an s-shaped bend (Figure 5.5(b)). A diagonally-set tendon can produce an asymmetric bend (Figure 5.5(c)); two diagonal tendons placed opposite each other can produce twist (Figure 5.5(d)).

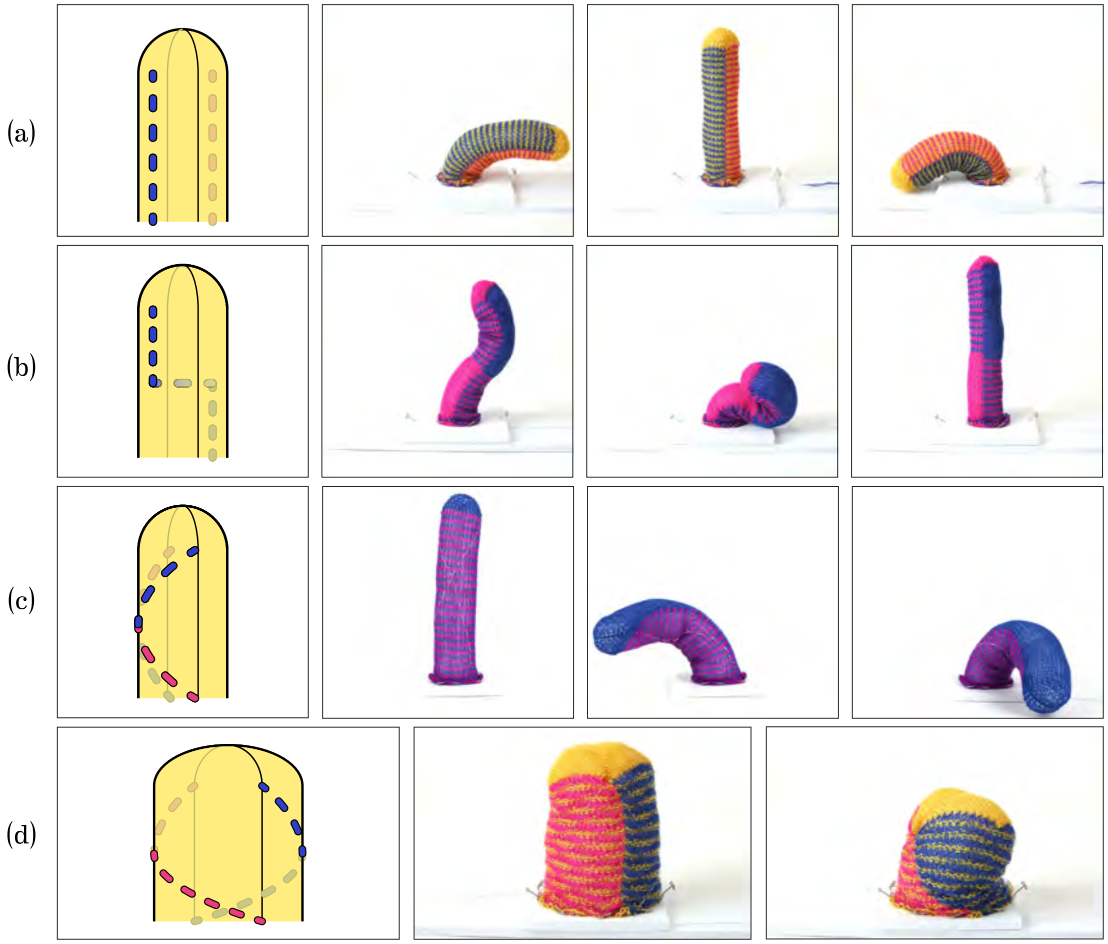
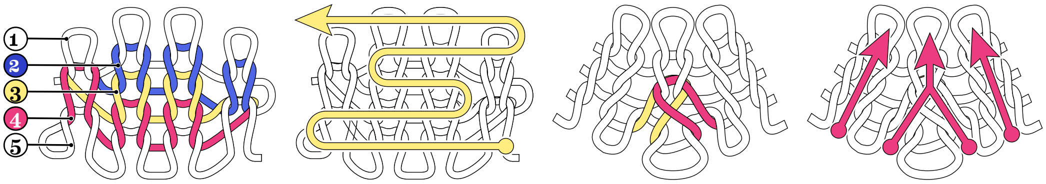


Figure 5.5 *Variations on tendon placement.*

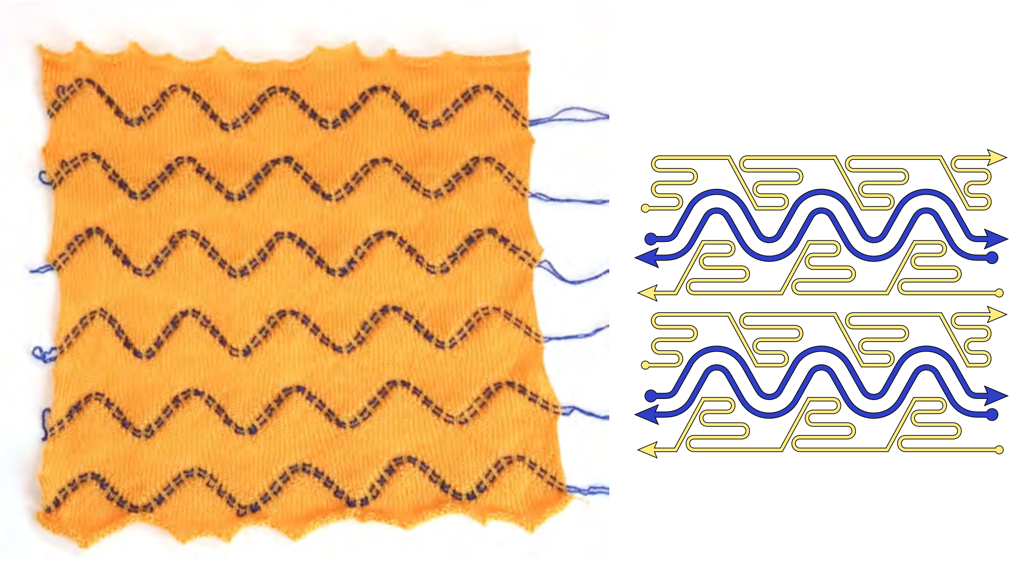
5.5. Shaping: Short Rows and Increase/Decrease Shaping

In addition to straight tubes with consistent cross-sectional geometry, I incorporate two shaping techniques that are common in hand knitting, where they are used for, e.g., the heel of a sock (short rows) or the taper of a hat (decreases). I show them here to increase the diversity of shapes available for soft knit actuators.



(a) (b) (c) (d)

Figure 5.6 *a) Short rows. The second, third, and fourth rows (counting from the bottom) are highlighted and contain fewer than a full row's count of stitches. b) The abstract yarn path of the short rows. c) A decrease. Two stitches in the second row are overlapped and thus share a descendant stitch in the third row. d) The abstract column path of the decrease.*



(a) (b)

Figure 5.7 *a) Short rows distorting the stitch grid into waves. b) Abstract yarn path shows the sequence of short rows.*

"Short row shaping" (Figure 5.6) refers to a technique in which some of the rows of the knit structure do not extend the full width of the structure (Figure 5.6, left). In other words, they are a way to distort the row/column grid. This distortion can be used to create local curvature out of plane, or, if used with "matching" nearby short rows to fill the gaps, to create rows that meander across the fabric (Figure 5.7).

The row/column grid can also be distorted via the introduction of *increases* or *decreases*: adding a column of stitches, or merging two columns into one (Figure 5.6, right). For example, this can be used to make vertically-oriented tubes branch or merge (such as the thumb on a glove), or horizontally-oriented tubes bend (Figure 5.13).

5.6. Anisotropy: Knit/Purl Texture

In addition to tendons and shaping, stitch-level anisotropy can play a critical role in actuation behavior. Knitting terminology draws a distinction between a loop that has been pulled through its parent from the nominal back of the fabric to the nominal front, typically just called a "knit," and a loop that has been pulled through from the nominal front of the fabric to the nominal back, often called a "purl" (Figure 5.8).

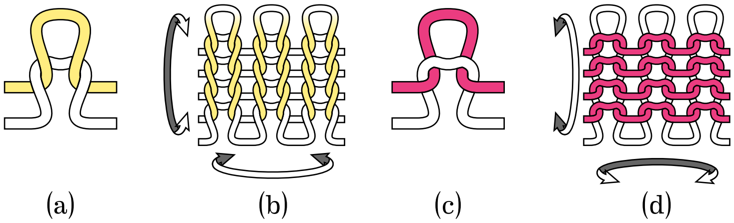


Figure 5.8 *Knit vs purl. a) A single "knit" stitch. b) An area of all knit stitches, which has a tendency to curl forward along the vertical axis and backward along the horizontal axis. c) A single "purl" stitch. d) An area of all purl stitches, with the opposite tendencies.*

Patterns of knit and purl stitches have specific anisotropic profiles because every stitch has a tiny bit of intrinsic springy curvature: each "knit" stitch has a tendency to curl forward in the vertical direction and backward in the horizontal direction, and each "purl" stitch does the opposite. This small amount of curvature accumulates with more stitches: a knit fabric made entirely of just one of the two variants will curl visibly.

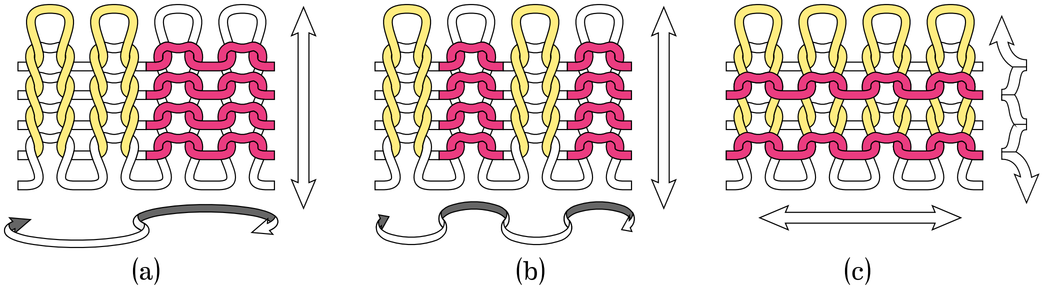


Figure 5.9 *a) A "rib" pattern that alternates two knits with two purls. Like all rib patterns, this pattern will tend to curl forward and backward (pulling inward) horizontally and be stable vertically. b) A rib pattern with one-stitch-wide ribs. c) A "garter" stitch pattern that alternates between a row of knits and a row of purls. This pattern will pull inward vertically but be stable horizontally.*

However, knit structures can be designed to use knits and purls in equal or near-equal quantities; these are called "balanced knits" in both hand- and machine-knitting. A common knitting structure, "ribbing," alternates between knits and purls in a row; since the direction of curl is switched for each vertical "rib," the fabric tends to draw in sideways but not curl from top to bottom; this makes it popular for use in the cuffs and hems of sweaters. Another common knitting structure, "garter stitch," alternates between full rows of knits and full rows of purls: the fabric is extra stretchy top to bottom and resistant to curling laterally.

The design of knit/purl patterns can be quite complex—Glazzard [105] discusses their use in making auxetic textiles—but I used the effect in this work primarily to create areas of directed bending. In contrast to the usual knitting emphasis of balancing knits to prevent curling, I deliberately introduced sections of all-knits or all-purls in order to form a localized hinge or pleat, Figure 5.10 and Figure 5.11. The curl direction of each stitch as shown in Figure 5.8 means that a vertical hinge of purls or a horizontal hinge of knits will result in a "valley fold," whereas the opposite arrangements will result in a "mountain fold."

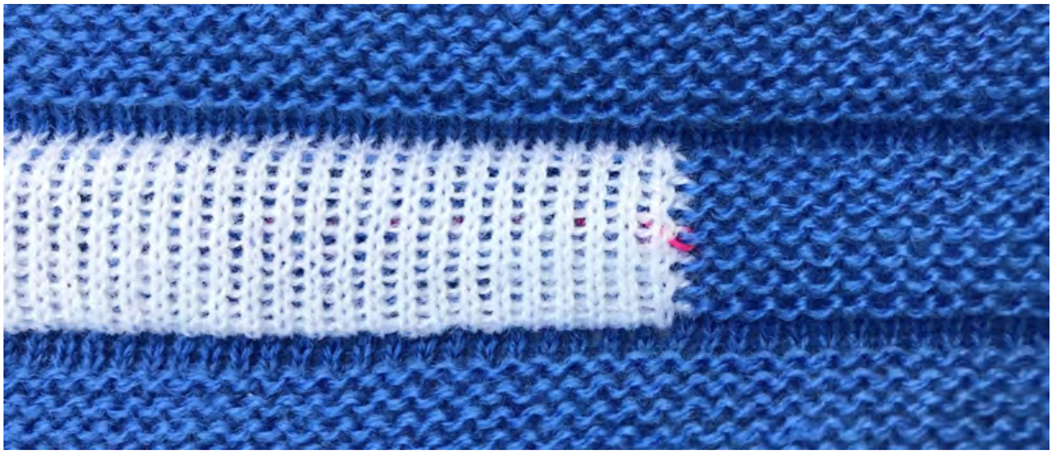


Figure 5.10 *A sheet with a pocket, which is part of the lamp example. The blue area is a single layer of knitting, primarily garter stitch, and the white is a second layer that connects to the blue layer at the top, bottom, and right edge, forming a pocket. Just above and below the pocket, full rows of all-knits define pleats that will bend forward, as a valley fold.*

Figure 5.11 *Hinge areas allow the active position of a sheet-shaped sample to fold along pleat lines.*

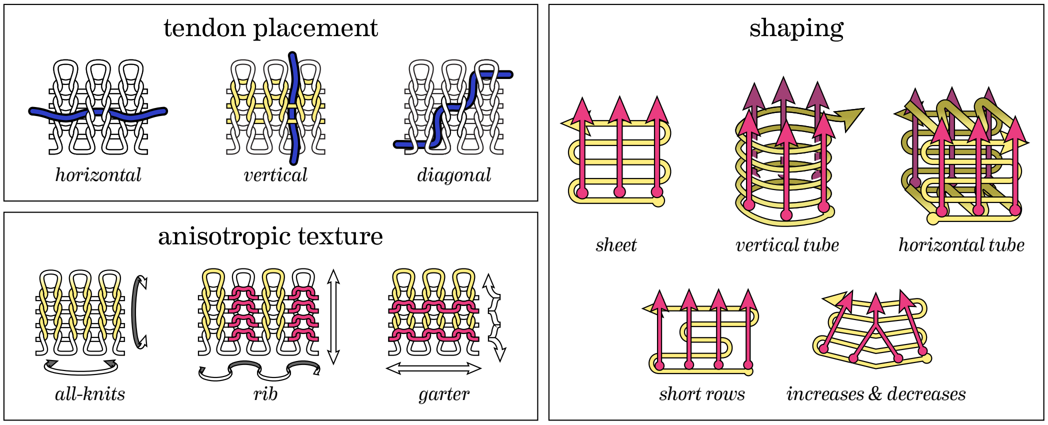


Figure 5.12 *Machine-knit actuation design elements: 1) the placement of actuatable tendons: horizontal, vertical, or diagonal with respect to knitting time; 2) a set of basic shapes (sheets, vertical tubes, and horizontal tubes) and techniques for modifying them (short rows and increases/decreases); 3) an approach to using the inherent anisotropy of knit stitches to produce areas of the knit surface with contrasting tendencies to curl, to produce local bending and pleating effects.*

5.7. Composition

To summarize, this work identifies three categories of machine-knit actuation design elements, shown in Figure 5.12. These techniques can be composed to produce more-complicated shapes. For example, Figure 5.13 shows a composition with a horizontal tube, a horizontal tendon, and decreases at the center of the tube to pull it into a v-shape. Short rows are used to taper the edges of the tube.

Figure 5.13 *A horizontal tube with inlaid tendon and decrease shaping to bend the tube into a v-shape.*

5.8. Materials Selection Guidelines

For repeatable motions, it is important that a mechanism can return to its original position. In a soft material tendon system, the recovery force must normally be supplied by a stuffing material; this force must be great enough to overcome both the friction along the tendon and the stiffness of the main knitting: for good recovery, one might ideally have a fairly stiff stuffing material, a fairly limp covering material, and a fairly slick tendon material. However, there are tradeoffs: a stuffing material that is too stiff might lose some of the benefits of soft actuation, depending on context. A covering yarn may have additional constraints on its appearance or other properties, such as conductivity. A tendon that is extremely slippery may slip out during the fabrication process, or be too weak to actuate without breaking.

I tested four yarns for the main knitting:

Brand	Name	Fiber	Weight
Yeoman	Volga 50/50	wool/polyester	7,143 m/kg
Yeoman	Polo	merino wool	15,000 m/kg
Yeoman	Supersheen	acrylic	15,000 m/kg
Tamm	Petit	acrylic	16,390 m/kg

There are many systems for characterizing the thickness of yarn. Yarn that is sold on cones for machine knitting is often labeled by "the yarn count system" which describes the number of strands spun together ("plies") in the yarn as well as the thickness of each ply relative to a standard thickness. While the "Supersheen" and "Petit" yarns have different mass per meter, they have very similar thicknesses; both are characterized as "2/30" (two plies, with each ply 1/30 of the standard) in the yarn count system. The "Volga" yarn is about twice as thick. While the thickness of the "Volga" yarn made for a perceptually stiffer and more opaque fabric than the others, this stiffness was not enough to affect the motion of cluster-stuffed mechanisms. I ultimately chose yarns based on visual design characteristics; for example, the "Supersheen" yarn lends a lacy appearance in the lamp example, and the "Volga" yarn provides a denser look to the sweater.

Because the exact characteristics of the main knitting yarn does not greatly affect the mechanical properties, it is possible to use specialty yarns in this role. For example, I used conductive yarn to create a capacitive touch sensor in the toy bunny (Figure 5.20).

5.8.1. Friction

I tested several tendon materials: the same Tamm Petit acrylic yarn I used for the main knitting, a 2/60 weight pure silk yarn, Superior Threads "Omni" polyester-wrapped quilting thread, and a 0.045″ nylon monofilament.

Of these materials, the fine silk yarn was the slickest, and thus offered the best recovery for cluster-stuffed objects. However, the quilting thread was stronger so I chose to use it for actuating objects stiffened with PETG. The nylon monofilament was neither stronger nor slicker than the silk or quilting thread, and additionally was stiff enough to periodically fail to knit cleanly, so I dropped it from consideration.

5.8.2. Stiffness

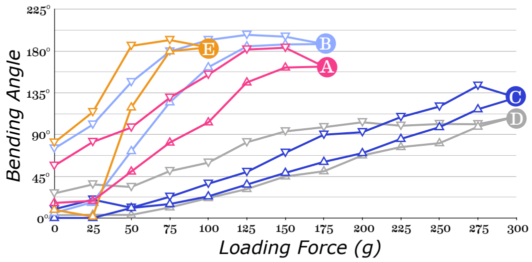


Figure 5.14 *A plot of the bending-recovery data shown in Figure 5.15.*

The stiffness of a knit object is determined primarily by the material that is used to stuff it, along with its height and thickness. I tested three stiffeners (samples A, B, and C): "Morning Glory" brand "Cluster Stuff" polyester fiberfill, 0.30″ PETG sheet, and 3mm EVA craft foam. I also tested the effect of larger or smaller stuffing areas when using cluster stuffing (samples D and E). To show the hysteresis in recovery position, I measured the bending angle of each sample under increasing loads from 0g until maximum curvature was achieved, then decreasing loads back to 0g. The samples are shown in Figure 5.15 and the data are plotted in Figure 5.14.

The PETG sheet offers the most complete and quick recovery, but required the most force to fully bend at the thickness tested; it may thus be too stiff for smaller-scale actuations. The craft foam was too weak to recover fully at the size tested, but it required the least force to actuate so it is suitable for smaller-scale motions that are desired to be relatively flat—for example, I used it in the ears of the toy bunny (Figure 5.20). The cluster stuffing offered less recovery force than the PETG sheet, but it has the advantages of being fully soft and capable of filling three-dimensional volumes, such as the arms of the bunny. The recovery ability of cluster stuffing is greatly influenced by its available volume. A very thick tube, such as Sample D, can return to very nearly its home position whereas a very thin one (Sample E) cannot.

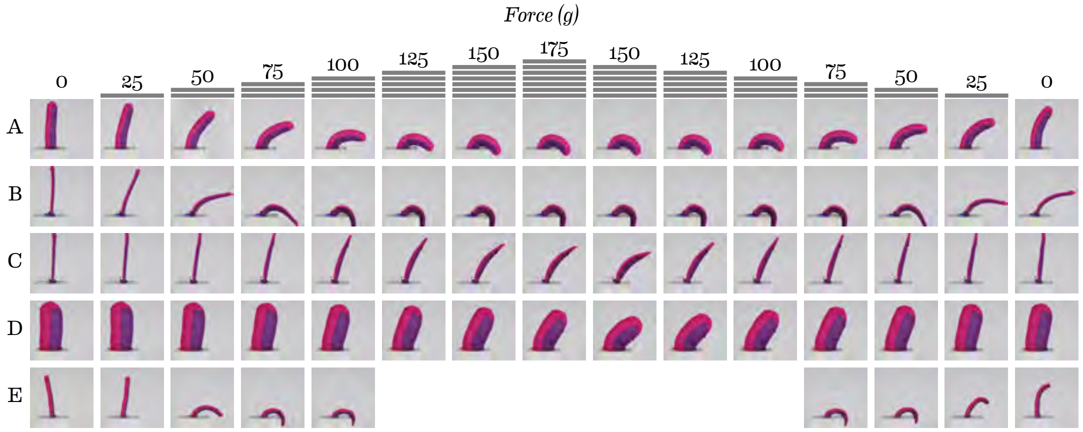


Figure 5.15 *Comparison of stuffing materials, A-C, and cross-sectional areas of stuffing, D-E. Sample A is 1.5″ and stuffed with "Cluster Stuff." Sample B is 1.5″ and stuffed with 3mm craft foam. Sample C is 1.5″ and stuffed with 0.30″ PETG sheet. Samples D and E are both stuffed with "Cluster Stuff"; D is 3″ and E is 0.75″. I show the full actuating range of Sample A; some other samples achieved their fully-bent state at higher or lower end forces.*

5.9. Embedding Interactivity

To demonstrate the flexibility of weft knitting, I showcase several approaches to adding electronic capabilities.

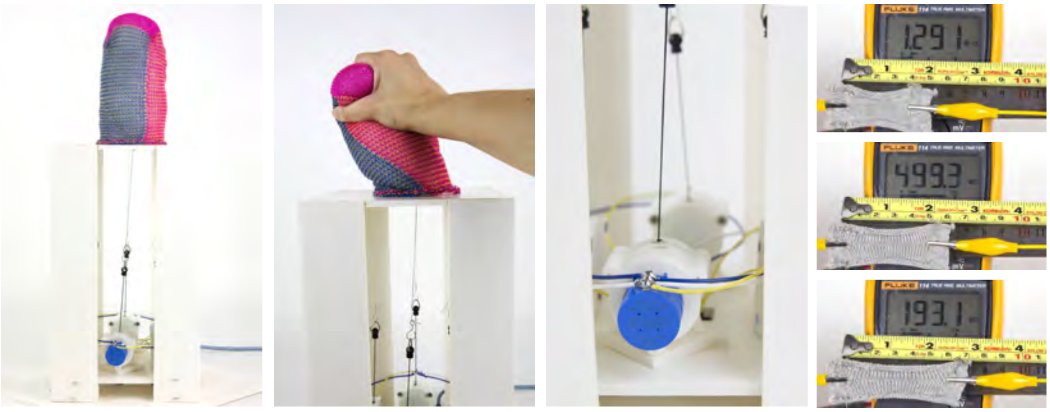
5.9.1. Motor Control

Motor control can be accomplished with a servo or DC motor setup like the one documented in the Soft Robotics Toolkit [137, 429]. As shown in my stiffness experiments (Figure 5.14), the force required to actuate my samples ranged from 100 to 300g. Assuming a 1.5″ diameter reel, this requires 190-570 gram centimeters or 2.6-7.9 ounce inches torque, well within range of a standard servo or DC hobby motor.

5.9.2. Sensing

I integrated three sensing mechanisms. First, I used the capability of the tendons themselves to transmit forces by coupling the tendons to a linear encoder. I used a simple string potentiometer made from a 10-turn potentiometer and the return spring from a badge lanyard [25]. By attaching a sensor to each of four tendons—front face vertical, back face vertical, clockwise diagonal, and counter-clockwise diagonal—wforward and backward bend and twist can be sensed (Figure 5.16(a)).

Two other sensing approaches involve knitting with a conductive yarn. First, an area of conductive knitting can be used as a contact pad for capacitive touch sensing, as I did in the toy bunny example. Second, because the loop structure of knitting makes variable contact as a knit swatch is stretched, an area of conductive knitting can be used as a resistive strain sensor [24, 399]. I saw resistance values of 1.29 mΩ at 0% stretch, 499 kΩ at 25% stretch, and 193 kΩ at 50% stretch for a swatch that was 2.5 cm by 4 cm between the test leads, knit from Bekaert 50/2 Cotton (Figure 5.16(b)).



(a) (b)

Figure 5.16 *a) Bend and twist sensing using string potentiometers coupled to front, back, and diagonal tendons. Each tendon is coupled to a string potentiometer which acts as a linear encoder. b) A swatch knit with Bekaert 50/2 Cotton plated with a hair-thin elastic.*

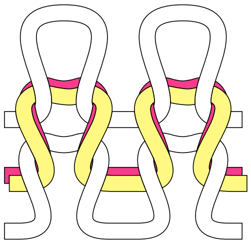


Figure 5.17 Two yarns knit into the same stitches using "plating."

Conductive yarns are often brittle and therefore difficult to knit reliably. To make a more physically robust sensor, the conductive yarn can be "plated" with another yarn. Plating is a technique in which two separate yarn carriers both contribute yarn to the same stitch, Figure 5.17. Because two separate carriers are used, the yarns don't twist around each other; instead, one yarn is always closer to the nominal front of the fabric and the other backs it. For the capacitive touch sensor in the bunny's belly (Figure 5.20), I plated the conductive yarn with Tamm Petit yarn for strength. For the strain sensor in Figure 5.16(b), I plated with an elastic yarn to ensure that the swatch returns to its original shape after stretching.

5.10. Complete Objects

In this section, I showcase machine-knit objects which combine tendons with shaping, knit/purl texture, and material variations for complex soft actuation.

Figure 5.18 *The* ***three-way tentacle*** *combines vertical tendon and shaping techniques: it has three vertical tendons and is shaped with decreases at the top.*

Figure 5.19 *The* ***gripper*** *combines both tube types and both tendon types: a horizontal (inlaid) tendon is set into a horizontally-formed tube, and a vertical (yarn tangling) tendon is set into a vertically-formed tube. An eyelet at the intersection of the tendons makes it easy to pull the strings through to a Bowden tube. The gripper is stuffed with cluster stuffing.*

Figure 5.20 *The* ***bunny*** *combines both tube types, both tendon types, both shaping techniques, and sensing. It is formed similarly to the gripper, but shows off the shaping complexity that is achievable with short row and increase/decrease shaping. Vertical tendons can actuate the ears, which are stuffed with craft foam. The rest of the bunny is stuffed with cluster stuffing, and horizontal tendons can actuate the arms in a hugging motion. The bunny is primarily knit in Yeoman Yarns "Volga" wool/polyester blend yarn, with an inlay of conductive yarn (Bekaert VN35X4) knit using the "plating" technique with Tamm Petit yarn for strength. a) Schematic of the bunny (not to scale). A: direction of knitting. B: Short rows bend the legs and torso. C: Decreases to shape the torso. D: Short rows at the edges of a horizontal tube provide roundness. E: Increases to shape the neck and head. F: Horizontal tendon for arms. G: Vertical tendons for ears. b) and c) A bunny with conductive belly and separately actuated ears and arms.*

Figure 5.21 *The sleeve of the* ***sweater*** *combines horizontal tubes, sheets, horizontal tendons, short row shaping, and anisotropic bending techniques. The body of the sweater shows typical sweater shaping with ribbing at the hem and collar. The sweater was knit primarily out of Volga yarn to give it an appropriate heft as a garment, with the pink sleeve inlays knit out of the thinner Supersheen to encourage them to buckle back into place more easily. The sweater sleeve was knit with a doubled-over tendon—a tendon was laid horizontally leftward, then, one row later, the tendon was laid back rightward again. Because of this, no knotting is needed to anchor the tendon in the knit fabric.*

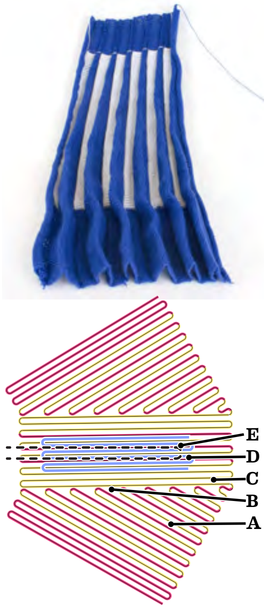
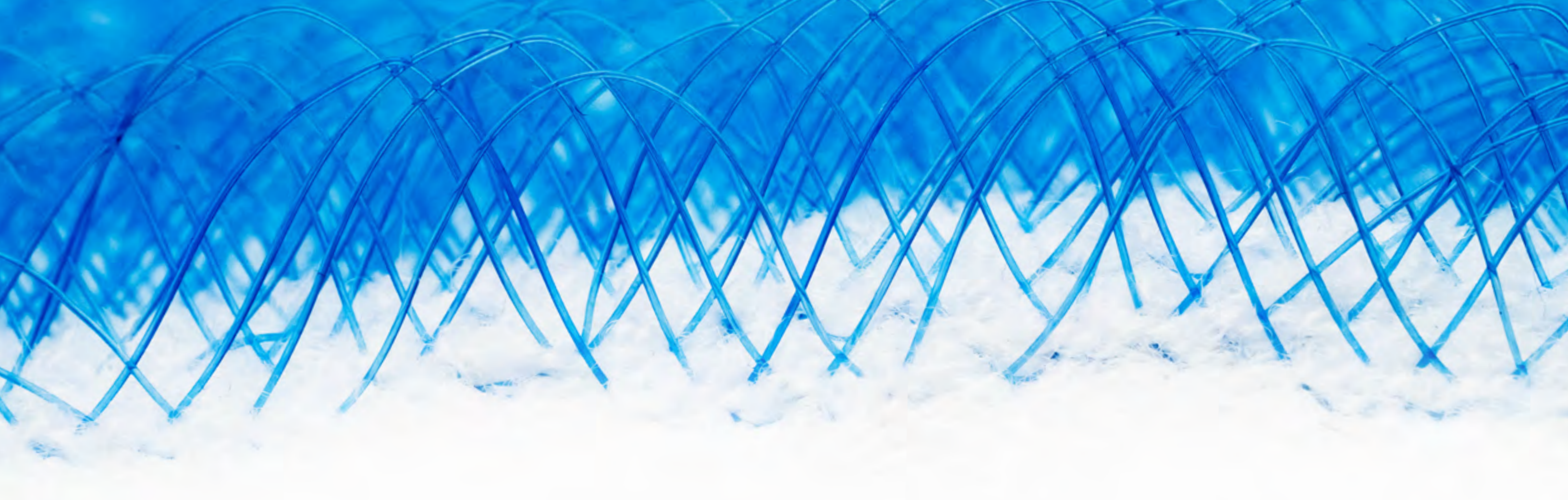


Figure 5.22 *Like the sweater, the* ***lampshade*** *combines horizontal tubes, sheets, horizontal tendons, short row shaping, and anisotropic bending techniques. The lampshade was knit out of Supersheen, which was the visually thinnest of my yarns, to give it a lacy appearance. Each horizontal tube is extended above, below, and to the side by a sheet to form a "sheet with a pocket." The tendon is inlaid into the pocket, which can contain a PETG sheet. Each sheet section has short row shaping to form it into a wedge—one such wedge is diagrammed to the right (bottom), not to scale. This section was repeated six times for the complete lampshade, shown from the back as-knitted to the right (top). Within the wedge, the main knitting (A) is done in a stable garter stitch with short rows (B) for overall shaping; areas of all-knit and all-purl (C) form pleats when the lampshade is relaxed. The pocket (D) is knit with double layers, as shown in Figure 5.10. At the middle of the pocket, the tendon is inserted horizontally (E); it is doubled as described in Figure 5.21.*

5.11. Summary

In this chapter, I have introduced industrial machine-knitting as a general-purpose fabrication technology of relevance to the field of human-computer interaction. Using the example of embedding tendons directly during the knitting process, I have demonstrated the **flexibility of low-level knitting machine operations**; by discussing the selection of **soft material characteristics** and how to compose tendons with knitting-specific shaping maneuvers like short row and increase/decrease shaping, I show how **material-level *soft technology*** can be approached.



6. Spacer Fabrics

Spacer fabrics are a category of knit structures that have a unique springy feel, breathability, and low density, making them useful for a range of applications such as uppers of running shoes and padding for orthotics. They are also used as replacement for foam rubbers for their relative sustainability and resistance to degrading [240] and as a structured fiber reinforcement for concrete [328]. However, these fabrics are typically produced on warp knitting machines, which are characterized by dedicated yarn feeders for each needle in a bed; these machines are very large, have relatively high setup overhead, and are typically used to produce undifferentiated yardage with very little within-fabric programmability. By using weft knitting instead of warp, the production parameters of the fabric can be tuned directly on a stitch-by-stitch basis, incorporating multiple functional characteristics in a given object.

6.1. The Spacer Structure

A knit spacer fabric consists of knit *faces* and a lofty *filler*, Figure 6.1. In a classic spacer fabric such as the warp-knit ones in the related industrial work, this filler yarn is semi-stiff and holds the two faces apart at a distance. In this work, I additionally investigate two variants: 1) a single-face variant, the “bristles” structure which I show in Section 6.5, and 2) a variant with soft filler yarn which can be used as a soft padding or in velcro-like applications as in Section 6.6.

As stretchable, compressible, and resilient materials [422] with excellent air and moisture permeability [212, 400], spacer fabrics have been used to replace conventional foam rubber in the context of cushioning and body support [364] and in architectural acoustics [211].

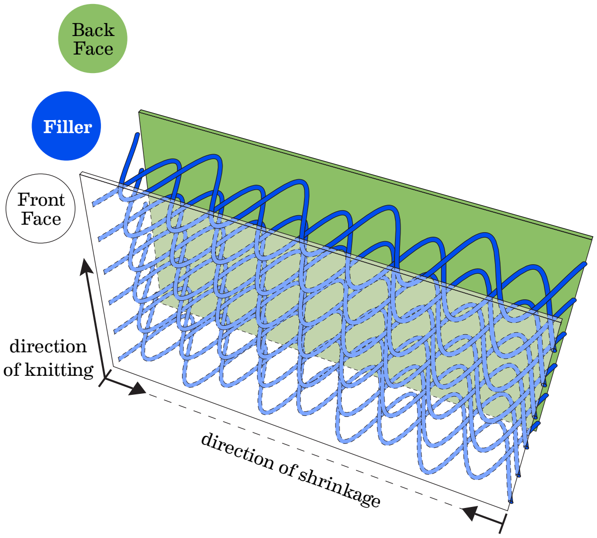


Figure 6.1 *The basic structure of a spacer fabric. Faces are shown in green and white, and filler in dark blue. A face may be removed to produce the “bristles” structure.*

6.2. Producing Spacer Fabrics on a V-Bed Weft Knitting Machine

The main industrial approach to produce spacer fabrics is in bulk using warp knitting [364]; as such, most material characterizations of spacer fabric focus on warp-knit materials (with some notable exceptions—e.g., my findings in section Subsection 6.4.1 improve upon brief claims in [22]). By investigating how to make spacer fabrics on the v-bed weft knitting machine instead, it becomes possible to make not just yardage, but entire shaped objects, embedding functionality from the tunable mechanical properties of the spacer structure as well as electrical routing (Section 6.6) and complex shaping (Section 6.7).

On a v-bed weft knitting machine, I knit spacer fabrics by alternating between two basic steps:

A number of **face rows** are formed using sequences of the *knit* operation, Figure 4.3. Rows are added to both the front and back faces, which are knit on the front and back beds respectively. At this time, therefore, the two faces of fabric are separated by only the small gap between the two needle beds. This is notably different than the warp-knitting process, in which the faces are created with the appropriate distance at knit time.

Filler rows are added at the same needles using the *tuck* operation, which incorporates the yarn into the stitch without forming another row of fabric, Figure 4.4. By tucking at a regular interval onto alternating beds, the filler yarn forms a shallow lengthwise zig-zag, Figure 6.2. Because tucking does not add height to the fabric, subsequent passes of filler yarn add density to the same face row of the fabric, as shown in Figure 6.4.

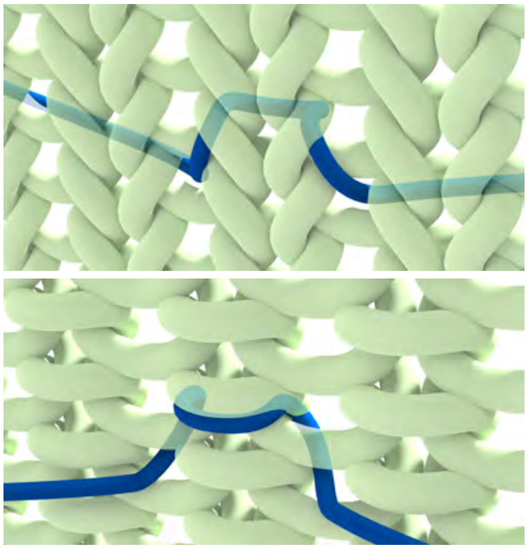


Figure 6.2 *A blue yarn tucked into a pale green knit stitch, shown from the front (outside) and back (inside).*

The key component of making spacers with weft knitting is that the faces are knit with an **elastic yarn** which is in tension at knitting time. After knitting, the tensioned elastic yarn in the faces causes them to shrink laterally (Figure 6.1), drawing in the filler yarn zig-zag and pushing the faces apart into the characteristic "fluffy" thickness of the spacer fabric.

6.3. Fabrication Parameters

As composite materials, spacer fabrics have properties that arise out of the interplay between individual input materials and how they are arranged. As shown in Figure 6.3, I organize these into *material* parameters, determined by the yarns used, and *geometric* parameters, which are controlled by knitting process and therefore ultimately by the machine code.

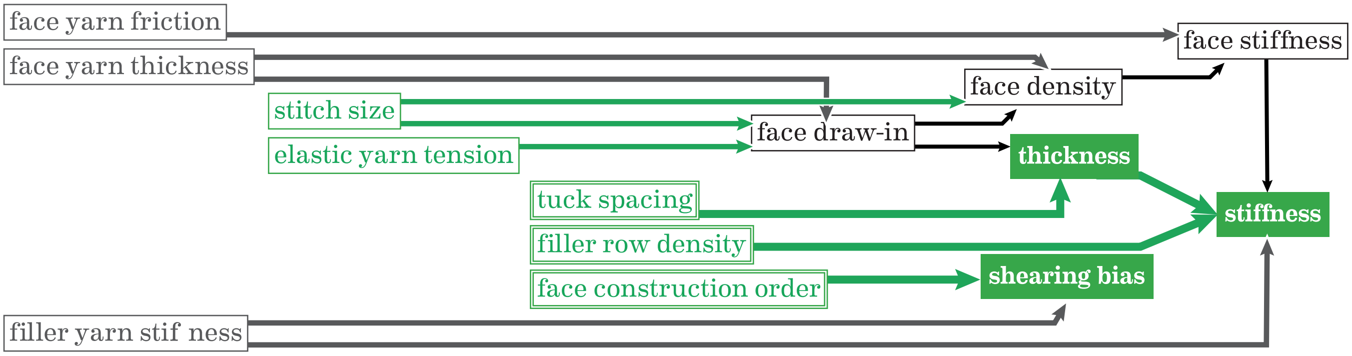


Figure 6.3 While the structural properties of a spacer fabric are influenced by many factors, I identify several material parameters (shown in dark grey italic text) and knit programming parameters (shown in green text) that can ultimately affect useful fabric characteristics (shown in white bold text).

6.3.1. Material Parameters

The material parameters include the stiffness, friction, and stress/strain properties of the yarns in the fabric. A spacer fabric has at least three component yarns:

Main Face Yarn[s]: The yarn which forms the faces of the spacer structure can be any machine-knittable yarn. It is trivial to knit the two faces in two different yarns, and possible to mix yarns within a face. The thickness of these yarns will directly contribute to the stiffness of the faces at a given stitch size.

Elastic: An elastic yarn is required to provide the lateral draw-in which produces the lofty spacer structure. The face yarn can itself be elastic, or an elastic yarn can be used alongside a non-elastic face yarn using plating (Figure 5.17). The tension on the elastic yarn as it is knit can be tuned—elastic knit under tighter tension will draw in more—but the elastic must be able to withstand the strain.

Filler Yarn: The filler yarn forms the zig-zag path between the faces. In order to achieve the lofty spacer form, the filler yarn must be stiff enough to push the faces apart; because it is only tucked into the face loops (not pulled through into new knit loops), it can be stiffer than the face yarns.

The main and filler yarns may also have task-specific material properties such as conductivity. In addition to individual yarn properties, the friction between the face yarns and the filler yarn affects the tactile qualities of the spacer fabric. "Stickier" face yarns do not spring back quite as easily, and can be stretched out of shearing bias. For example, I found that the combination of Bekaert VN35X4 conductive polyester yarn and nylon monofilament that I used in the capacitive sensor shown in Figure 6.14(a) was somewhat "sticky" in the spacer configuration. This gave the button a more malleable feel, with delayed return, than swatches knit with the default slightly-fuzzy acrylic yarn.

As a practical matter, the yarns used for the spacer fabric must be compatible with machine knitting—that is, they must not be too thick for the gauge of the machine, nor too weak or brittle to survive the knitting process. Unless otherwise noted, the spacer fabrics in this chapter were knit with Tamm "Petit" acrylic yarn with hair-thin (0.06 mm⌀) latex elastic for faces and KastKing 6lb test monofilament (0.22 mm⌀) for color-tinted filler and Hi-Seas "Grand Slam Mono" 6lb test monofilament (0.25 mm⌀) for clear filler.

6.3.2. Geometric (Programmable) Parameters

The *geometric* parameters of spacer fabric are based on the sequence of knitting operations, so they can be altered programmatically.

Tuck spacing: The distance in needles between a filler yarn tuck and the next tuck (on the opposite bed). The distance between tucks must be less than six and greater than one to ensure clean tuck formation on our 15g machine.

Tuck pattern offset: The distance in needles between the tuck positions in one pass and the tuck positions in the previous pass.

Filler row density: The ratio of filler rows to rows of face fabric height. As shown in Figure 6.4, filler rows do not add height. In knitting, every row has a direction of formation (leftward or rightward) which must be alternated. To greatly simplify the programming of these structures, I always pair a leftward pass with a rightward pass; therefore, in all my examples, the number of rows of each are always even; I factor out this duplication in expressing the filler row density as a ratio. The fabric shown in Figure 6.4 has a filler row density of 2/1: four filler row passes for every two face passes (leftward and rightward).

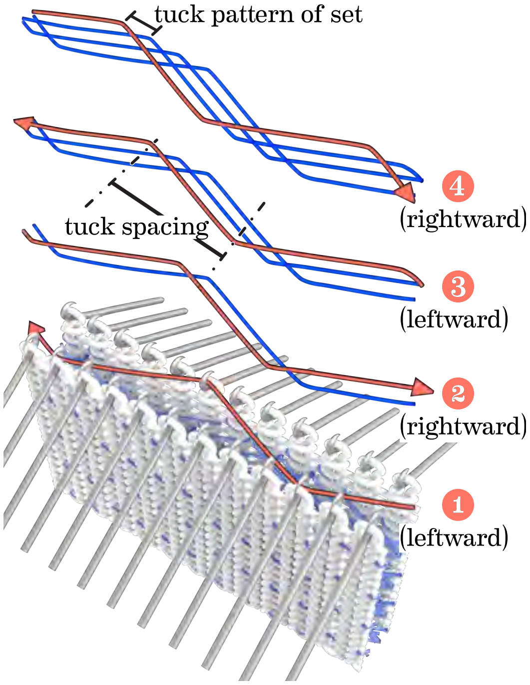


Figure 6.4 *Tuck spacing and offset.*

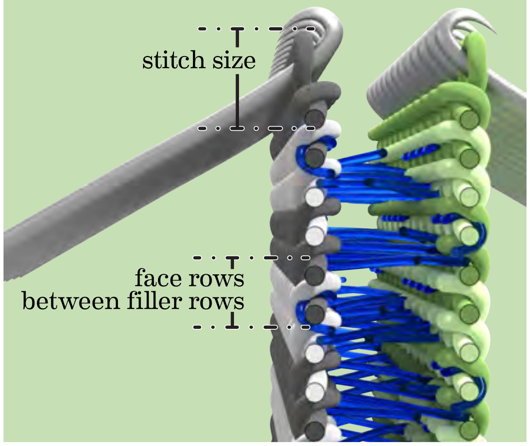


Figure 6.5 *Row density and stitch size.*

Stitch size: The size of knit stitches produced by a knitting machine arises from a combination of factors, including yarn tension and post-processing. The most dynamically adjustable factor is the *nominal stitch size*, which is the amount of yarn pulled into a loop as it is formed, as determined by programmable stitch cam settings. For a given yarn thickness and friction, stitch size affects the density and stiffness of a knit fabric. Within a spacer fabric, face fabric density can affect draw-in, as a fabric that is already quite dense may not be able to draw in further. While a given knitting machine has inalterable physical spacing between the needles, the stitches can be made farther apart by integral multiples of this spacing; in this work, I knit at “half gauge,” with intervals of one needle between stitches. I did this both to allow for ease of shaping operations (as described in [233]) and to promote good draw-in.

Face fabric knitting order: Which face (front or back) is knit first after a set of filler rows.

Yarn tension: Depending on the knitting machine, it may be possible to control the tension on the yarns during knitting. Our Shima Seiki machine allows row-by-row tension settings for the elastic yarn feeder only. Elastic yarns are typically knit under tension; in the context of spacer fabrics, the force induced by this tension causes the face fabric draw-in. I chose a hair-thin elastic for this work to best support mixing with a wide range of other face yarns. I found that this elastic has a narrow range of working tensions without breaking, so in practice, I chose to keep the elastic yarn tension constant. For the non-elastic yarns, tension is typically set to minimize breakage and dropped stitches and it is not altered during knitting.

To keep my explorations tractable, I decided to maintain constant yarn tensions, tuck pattern offset (1), nominal stitch size (as expressed in the machine units of the Shima Seiki, this was “stitch” 40, and “leading set” 25), and half gauge throughout the presented work.

6.4. Resulting Spacer Fabric Characteristics

As described in the previous section, some of these parameters must be kept in a narrow range to ensure knittability (e.g., elastic yarn tension), and I kept some constant to avoid exponential explosion. However, others can be altered to affect the overall characteristics of the composite spacer fabric structure. In the following sections, I outline such characteristics and describe how underlying production parameters can affect them. These effects are particularly useful in the case of the geometric/programmable parameters: because these can be altered with machine instructions, hence they can be varied within a given knit object. I provide an overview of the relations between knitting parameters and resulting fabric characteristics in Figure 6.3.

6.4.1. Thickness

A defining characteristic of spacer fabrics is their thickness. While the thickness of the individual faces can be altered with thicker or thinner face yarns, the dominant factor of the overall spacer fabric thickness is the distance between the faces. Assuming a filler yarn that is semi-stiff, such as a nylon monofilament, this distance is determined by two factors: the spacing of the tucks, and the lateral draw-in of the face fabrics induced by their elastic.

As illustrated in Figure 6.6, when viewed from the top, the filler yarn layout can be diagrammed as triangular struts. Consider each section of filler yarn as the hypotenuse of a right triangle: when the fabric is knit, the tuck spacing sets the base length x of this triangle, and the gap between knitting machine beds ("bed gap") sets its height b.

As the hypotenuse, the filler yarn has length $h = \sqrt{h^2 + b^2}$. When the face fabric is drawn in by factor s to final length $x' = sx$, the filler yarn remains the same length, resulting in an overall fabric thickness of $t = \sqrt{h^2 - x^2} = \sqrt{x^2 + b^2 - (sx)^2}$

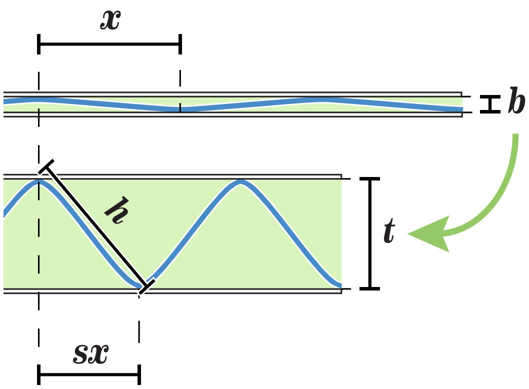


Figure 6.6 *The spacer fabric "draw-in transition" can be modeled by thinking of the filler yarn as the fixed-length hypotenuse of a right triangle whose base shrinks as the face fabric shrinks.*

The lateral draw-in of the fabric is determined primarily by the tension of the elastic yarn at knit time and to a lesser extent by the face yarn thickness and stitch size. As described in section Subsection 6.3.2, the range of possible elastic tensions for a given elastic may be narrow. The spacing between tucks also has practical limits (too small and it's not a spacer; too large and the tucks may not form cleanly) but there is variability possible within that range.

In order to test this model, I created sample swatches with tucks spaced 1, 2, 3, 4, and 5 stitches apart, with other design factors held constant, and measured the thickness of each sample in three locations. The proposed model is a good fit to these thickness measurements (Figure 6.7); and, further, the coefficients of the model (bed gap $b = 3.913\text{mm}$, shrink factor $s = 0.653$) are close to the measured values for the Shima Seiki machine (bed gap of ≈4mm) and test fabric (shrink of 68%); suggesting that the model might work equally well as a predictive model given just this information.

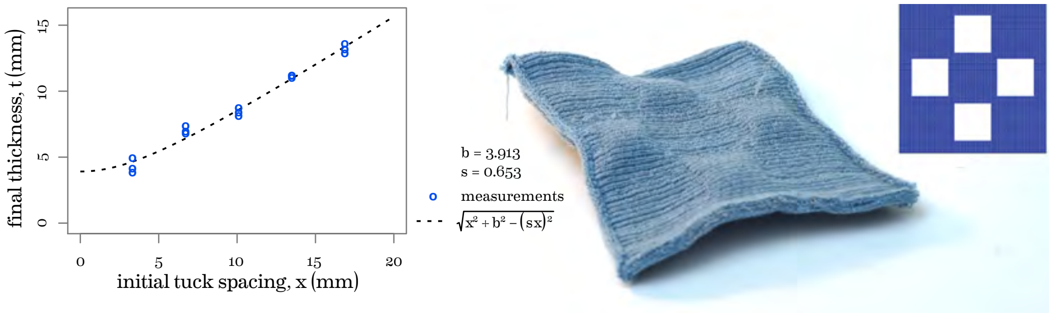


Figure 6.7 *This simple model fits measured test data. Differences in tuck spacing can be used to generate fabrics with areas of different thickness.*

Because thickness can be varied with tuck distance changes, different thicknesses can be implemented within a fabric without special attention to yarn carriers or extra yarn-inserting maneuvers. This allows thickness-based surface patterning such as the example in Figure 6.7.

6.4.2. Stiffness

Spacer fabrics are also notably stiffer than non-spacer fabrics knit with comparable yarns. Stiffness is the result of several properties: the stiffness and stretchiness of the face yarns (determined by face yarn, stitch size, and tension), the filler row density, and the thickness of the resulting structure (determined by tuck spacing, as described above).

To test these proposed stiffness parameters, I knitted swatches varying tuck spacing and the ratio between face rows and filler rows. Because I found that fabrics with multiple monofilament tucks on a needle at the same time consistently had problems knitting cleanly, I excluded cases in which close tuck spacing combined with higher numbers of filler rows resulted in re-visiting a tuck location before knitting face fabric at that location. I additionally tested swatches with 1a) filler yarn tucks all on the same bed and 1b) no filler yarn at all, to isolate the effect of the monofilament's stiffness unaffected by the spacer thickness; and 2) no elastic yarn at all, to isolate the effects of face fabric density and elastic recovery. These were all knit with the same number of stitches per face and the same stitch size.

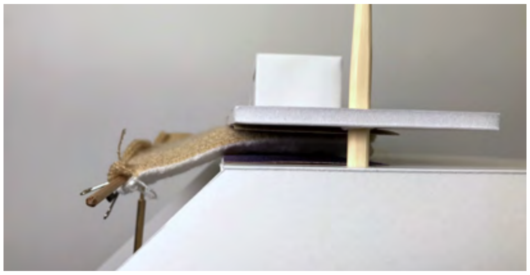


Figure 6.8 *For stiffness testing, each swatch was lightly clamped to its center line between two sandpaper surfaces. The free edge was successively loaded until it bent below a 45° reference line.*

I tested each swatch for bending stiffness in both horizontal and vertical directions. Each swatch was positioned in a high-friction (sandpaper) rig and clamped with a light (150g) weight, Figure 6.8. Successive force was applied until the bottom edge of the swatch was level with a 45° line from the clamped edge. I chart results in Figure 6.9. Overall, both thickness and row density indeed affect stiffness. Because comparable stiffnesses might be achieved with either row density or tuck distance manipulations (compare the swatch with 1/4 row density at tuck spacing 5 to the swatch at 2/1 density with tuck spacing 2) spacer fabrics offer some aesthetic freedom in how stiffness differentials might be arranged.

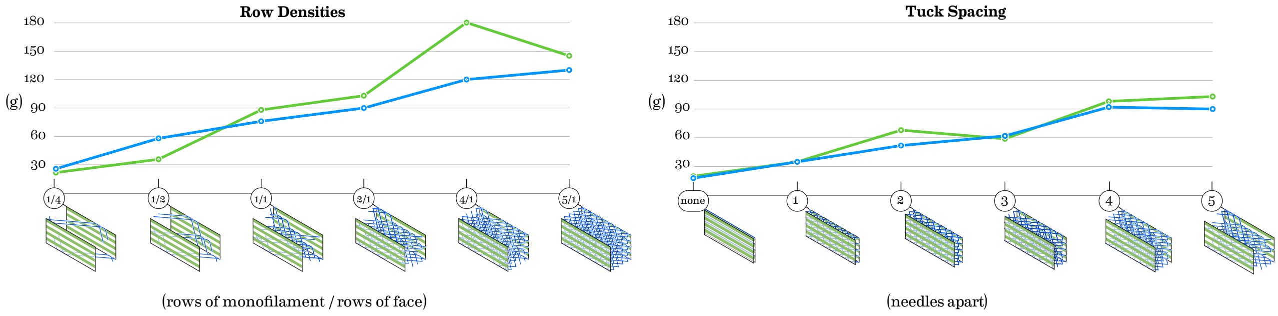


Figure 6.9 *Mass required to bend samples to 45° in the bending rig. Blue lines chart bending in the horizontal direction (along a row); green lines chart bending in the vertical direction (along a column of stitches). Left: Varied row densities at a constant five needle tuck spacing. Right: Varied tuck spacing at a constant 2/1 row density.*

6.4.3. Shearing Bias

Spacer fabrics can have a slight but noticeable mechanical bias in the vertical direction: under pressure, the fabric will tend to shear to the same direction.

This bias can be produced by altering the order in which the face fabrics are formed near the filler row, Figure 6.10. The face which is knit first will tend to shear downward. The bias can be minimized by alternating between face-knitting order, and the opposite bias can be introduced after a gap of face rows without filler rows joining them.

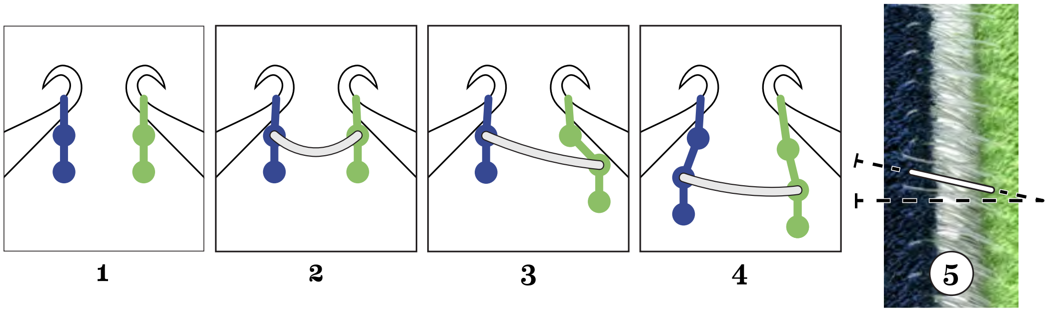


Figure 6.10 *Bias formation in spacer fabrics. 1: The knitting machine beds shown from the side, with two rows of face fabric on each bed. 2: A row of filler yarn has been added; the springy filler yarn attempts to push the cloth apart, but cannot (owing to the close spacing of the needles). 3: Adding the next row of one face (shown in green) allows the filler yarn to spring outward, skewing the connection in one direction. 4: When the next row is added to the other face, the skew remains and reinforces the skew of future rows. 5: This skew bias is visible in the final fabric, shown from the side. In this swatch, the right (green) face will shear downward.*

This effect is reasonably repeatable with the acrylic yarns I used, though I have two important notes for other practitioners: (1) having unequal numbers of front and back tucks in the first spacer course (which can happen for spacer regions of certain widths) seems to override the effect of face order entirely; (2) occasionally (10-20% of the time), the shear bias of a spacer patch will be flipped from what one would predict otherwise; re-knitting the same pattern will often produce the expected bias. (One can also clamp a sample in the desired orientation to re-set its bias; though no examples shown in this paper have been clamped.)

6.4.4. Soft Linkages

The bias in spacer fabrics produces a directional shearing motion under pressure. I demonstrate this effect in Figure 6.11: sections of different bias can be arranged, and the fabric folded, to produce a mechanism based on shearing linkages. In one folding configuration, squeezing the fabric tends to shear the outer layer away from the inner, causing the overall fabric to bend.

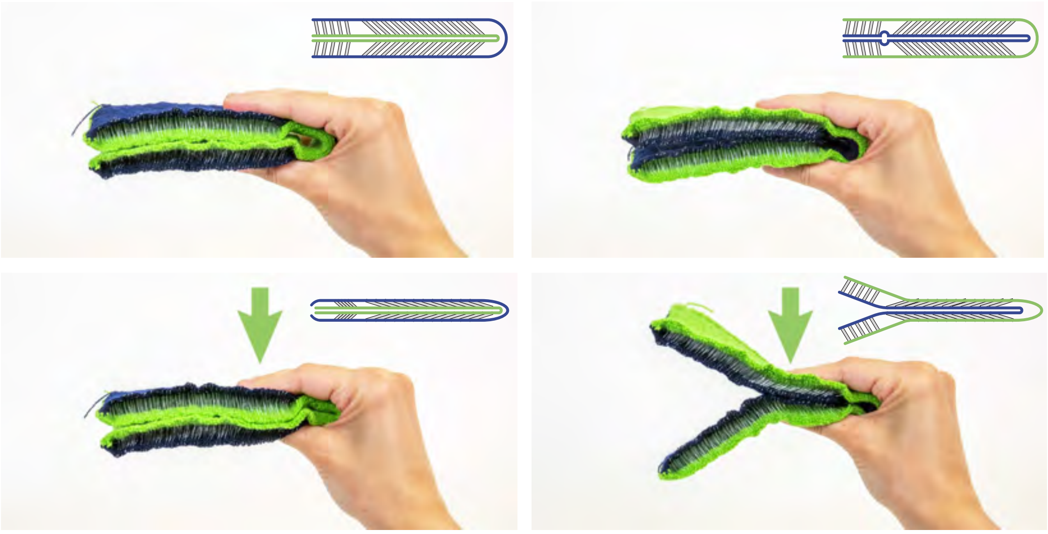


Figure 6.11 *Bias in spacer fabrics can be used to make anisotropic shearing linkages.*

6.5. Robotic Skins

The "bristle" structure is formed the same way as the spacer fabric, but one face is unraveled after knitting. The resulting structure is very similar to a "plush" or "terry" knit, but with much longer loops than are typically produced in these processes. Knitting a sacrificial face helps ensure that the bristles are formed properly; without it, the monofilament may not drop cleanly off the needles after each row. The most predictable results are achieved if the "sacrificial" unraveled face was knit in the same kind of yarn as the remaining one.

Inspired by kirigami-clad pneumatic mechanisms, [315], I applied a biased bristle skin to a fiber-reinforced elastomeric extension actuator [136] Figure 6.12. When the actuator is pressurized, it stretches uniformly; when it is allowed to relax, the bristle structure acts as a ratchet and keeps the front part of the actuator in place while the back slides forward. This "caterpillar" robot is soft and lightweight.

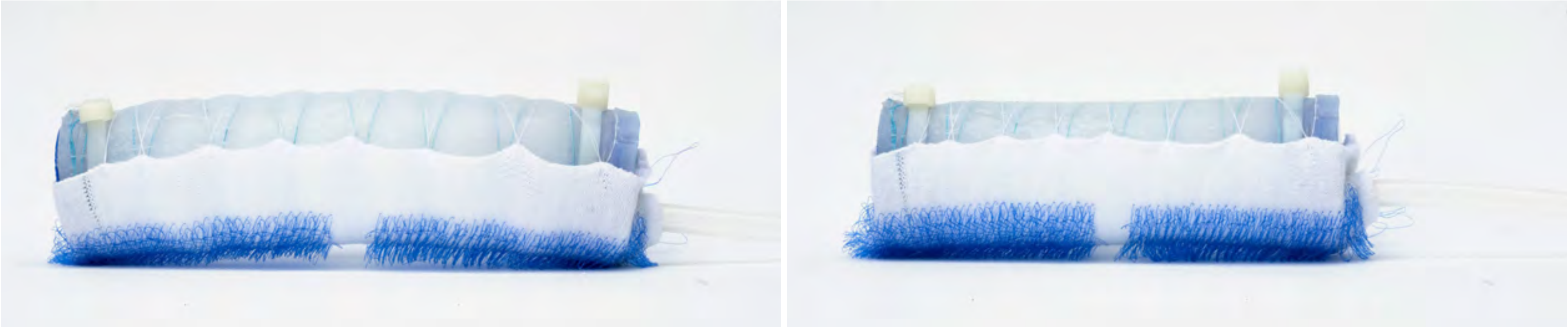


Figure 6.12 *Biased bristles as cladding on a pneumatic extension actuator.*

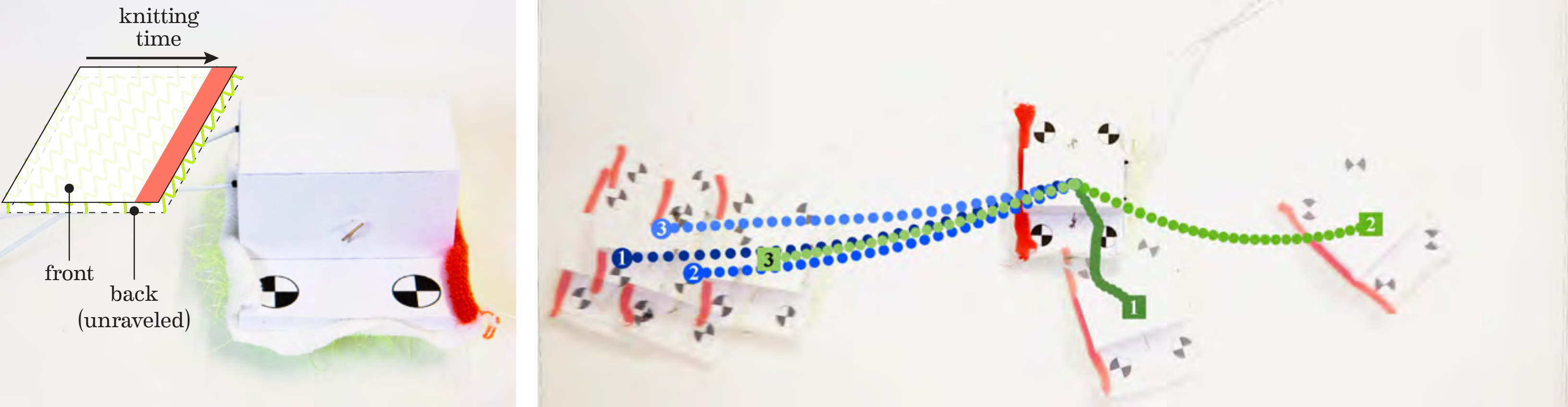


Figure 6.13 *Varying the face knitting order and bristle height can produce bristlebots with different travel directions. Paths are plotted over the first 1 second of travel. Dots are placed every 30th of a second. Blue paths marked with a circle indicate swatches knit with alternating face knitting order; green paths marked with a square indicate swatches with entirely front face priority knitting order. Within each color, the bristle length is shown as shades from dark (shortest bristles, tuck distance 1) to light (longest bristles, tuck distance 3).*

"Bristlebots" use springy bristles to turn vibration forces into net motion [281, 282]. Because the "bristle" structure can have knit-in bias, fabrics can be produced which encourage linear motion to a greater or lesser extent (Figure 6.13). As noted in Subsection 6.4.3, this behavior shows some variation: the behavior of the front-face-priority bristlebots is somewhat inconsistent.

6.6. Sensing and Soft Switches

Knitting can incorporate areas of conductive yarns to support sensing [399]. For example, simple capacitive touch sensors [118] can be constructed from knitted-in conductive patches [15]. As with any soft sensor, care must be taken to interface with rigid circuit boards [307]; weft knitting simplifies this task because the traces themselves can be placed precisely within a knit, increasing the options for component placement.

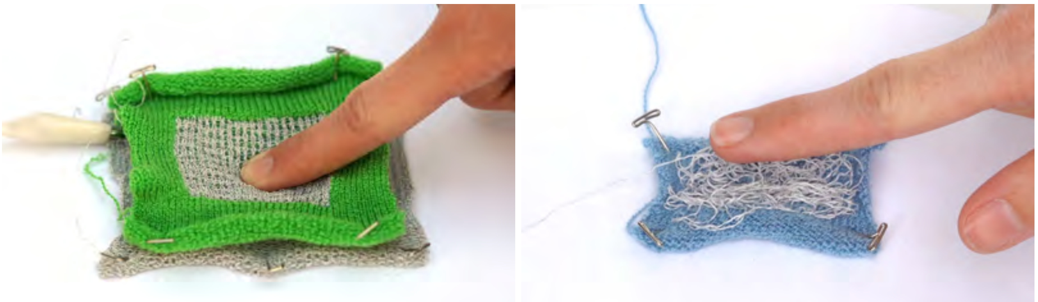


Figure 6.14 *The face or filler yarns can be conductive, for e-textile sensing. Left: each face is knit with a separate conductive yarn; capacitive sensing can detect pressure as the faces are pressed closer together. Right: a bristle fabric with conductive bristles is a soft resistive sensor.*

Spacer fabrics lend themselves to several useful sensing configurations that can take advantage of their unique physical properties.

First: if both faces of the spacer fabric are knit with conductive yarns, and the filler yarn is non-conductive, the spacer structure can act as a force-sensitive capacitive touch sensor, Figure 6.14(a). I knit conductive yarn into areas of both faces of the spacer fabric. One face has edges knit with non-conductive yarn (shown in green) to prevent short circuits at the edges of the faces, where they curl and make contact. Using an off-the-shelf NXP Semiconductors MPR121 capacitive touch sensor chip attached to a microcontroller through an I2C interface, I was able to detect hover of a finger just above the surface of the face, touch of the surface, and most notably, movement of the two faces towards each other as pressure was applied to the top surface. This final capability is able to take advantage of the mechanical properties of the spacer fabric to create a type of soft compression sensor.

Second: 3D printed bristle patches have been shown to work as as a soft symmetric attachment similar to a hook and loop fastener [282]. Knit bristles can also achieve this effect if the filler yarn has high enough friction, such as a "felting" (not "superwash treated") wool yarn. By mixing a conductive yarn into the bristle structure, a soft switch can be made, Figure 6.15: the bristles are knit with alternating rows of conductive and pure wool yarn as the filler, making the loops both conductive and self-sticky. The wool loops provide adhesion through high contact area friction, and the conductive loops carry electrical signal. The same swatch is very pleasant to the touch and can be used as a soft tactile "stroking sensor" inspired by [182].

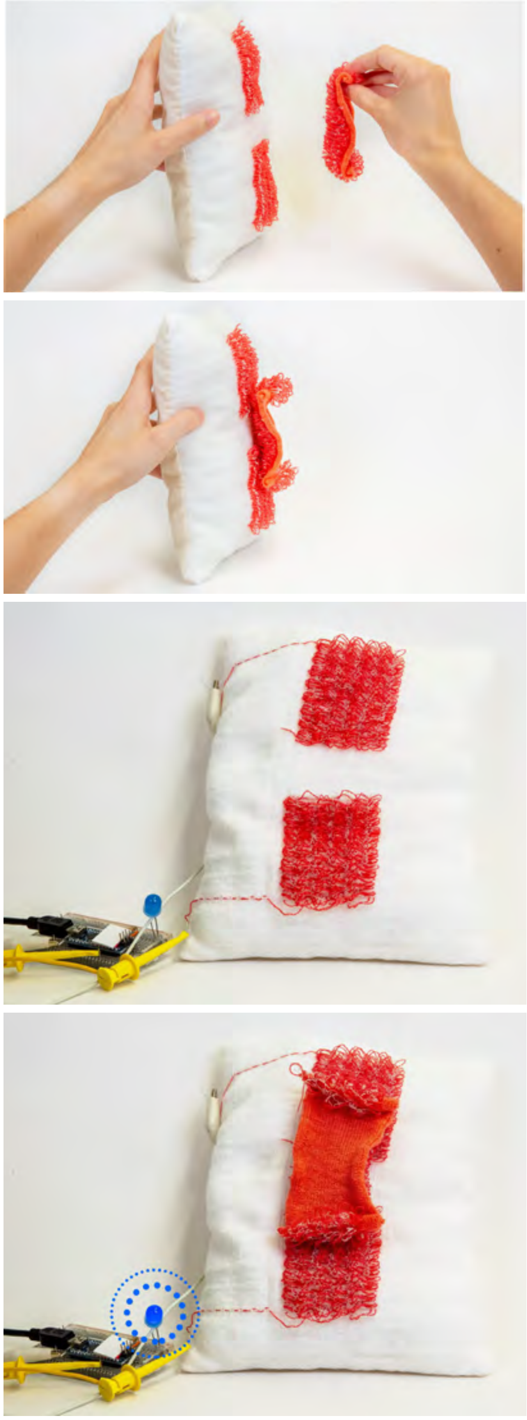


Figure 6.15 *A "hook and loop"-style switch.*

Figure 6.16 *A soft control pad. Different thickness provide tactile cues to the location of capacitive button traces.*

Third: by combining capacitive sensing with tunable thickness, a button pad can be made which highlights interface areas with visible, tactile thickness differences. These differentially raised areas can be used to form custom collections of soft controls–in this case the geometry of a soft button pad has been formed directly within the fabric of an armband, Figure 6.16. For this application as a button pad, I used simple capacitive touch sensing on the top surface only. The conductive yarn was plated alongside the main face yarn in the contact areas.

In the time since the work in this chapter was originally published, Aigner et al. built on it to produce robust compressive resistive sensors for a variety of input modalities [9].

All of the e-textile examples in this chapter use Bekaert VN35X4/150POLY/350 (a stainless-steel and polyester yarn) as the conductive yarn.

6.7. Integration into Knitted Objects

Lastly, the flexibility of v-bed knitting as a process enables combining several knitting techniques within a knit object. Incorporating the tendon-embedding technique from the previous chapter, I produced a "pre-stuffed" knit tendon assembly suitable for soft gripping, Figure 6.17. The full assembly has a "palm" with multiple fingers, each with rounded fingertips. The assembly is knit sideways, in the direction parallel to the fingers, and uses a horizontal inlay for the tendons. After knitting, the structure is fully bound-off, stable, and ready to attach to a 3D-printed snap-together mount.

Figure 6.17 *Fully-knit fingers for a soft gripper. Left: The assembly as-knit. Right: The gripper shown open and closed.*

6.8. Summary

In this chapter, I investigated weft-knit spacer fabrics as a compliant mechanism with complexly emergent material properties. By understanding the interplay of the underlying yarns and the loop-by-loop operation of the knitting machine, it is possible to characterize local material effects in the overall fabric, including thickness, stiffness, and shearing bias. These can be combined with features like conductivity and overall shaping to produce computationally-knit functional materials suitable for advanced engineering applications. These capabilities are possible not *in spite of* but *because of* the underlying **soft properties** of friction, elasticity, and tension.

II.

Malleable Contexts

7. Background and Related Work: Soft Situation and Forms

This Part focuses on how the *context* of a material technology shapes its design and development. Many technologies are tightly designed for a specific operating environment or task; in computational fabrication, this is often assumed to be a kind of mini-version of industrial production, with the specific goal of producing a known outcome efficiently and repeatably. Outside this context, it can be difficult to adapt or re-use fabrication technologies toward any other purposes; for people other than enthusiasts, it can be nonobvious why they would even want to do so. A *soft* context is less sharply focused. It may encompass varying priorities, or scenarios that spread out over time or participants, and it taps into subjective factors like culture and aesthetics.

Contextual approaches consider situatedness: the participants, locations, relations, and histories that shape how something is used; "the ultimate particular" [264]. Designing for a soft context is no less specific, but it is a specificity that may shift over time, or be blended with other specifics. In this dissertation and particular in this Part, I blend **the context of computational control and its unique modalities**, such as networked communication and the transmutability of data, with **the context of textile production as both a highly technological industry and a deeply cultural practice**. I am particularly interested in what it means to have a *personal practice*—an ongoing relationship to a material technology—which must inherently adapt over time, and how such a practice can be seen as a collaboration between current and future selves. In this, I draw on theories of procedural design, in which a creator both designs a tunable generative system of related outputs and uses it to produce new works.

7.1. Personal and Site-Specific Fabrication

Because of its historically high cost, complexity, and historical ties to military funding [50], computational fabrication is not typically seen as personal. We often imagine robotic fabrication machines operating in spotless factories to produce large quantities of consumer goods; these machines are large, dangerous, and inscrutable. In addition to being fairly unlike real factories, this vision discounts the possibilities of computational fabrication in a wide range of *other* contexts. Machines which are less expensive, less dangerous, and simpler to control can be suitable for a classroom, an artist workshop, or a small research lab. Such machines can be used by individual creators for everyday tasks like modification and repair [190].

In 2016, Baudisch discussed the implications of computational fabrication transitioning from an industrial technology, through being an enthusiast technology, to being used by "consumers." He identifies six "challenges" facing the widespread adoption of such fabrication system, with two in each of the three categories of society, usability, and hardware (which I find resonant with my materials/workflows/contexts framing), but nonetheless foresees a "massive" shift in the "user base," with implications for "new and relevant contents, as well as new forms of artistic expressions" [28].

Work in portable and modular fabrication machines has greatly expanded the possibility of physically bringing computational fabrication into new contexts. Jacobs and Zoran observed that digital equipment which was *mobile*, *robust*, and *open-ended* enabled their collaboration with Ju/'Hoan artisans in Namibia [159]. The Jubilee system by Vasquez et al from Peek's lab, Machine Agency, is designed with modular tool-changing to support a wide variety of tasks, while itself being relatively easy and inexpensive to build, with demonstrated applications including lab automation, microscopic imaging, and digital art [395]. An earlier work by Peek and Moyer debuts "the laptop of digital fabrication," a portable fused-filament 3D printer that can be folded down into its briefcase, to speculate on fabrication as a site-specific and personal process [295]. Quitmeyer and Perner-Wilson prototyped "wearable studios" for digital practice that are arguably even more personal (being literally wearable) and proposed for anywhere-in-the-world deployment but particularly tuned to being used while hiking in Panama and Madagascar [313].

Other work has focused on making digital technologies accessible and highly contextual by prioritizing materials and processes that are already available to a maker, such as cardboard [147, 294], paper [311], DIY conductive paint and oven-bake clay [301]. In Chapter 9, I discuss a Jacquard loom that is specifically designed to be portable and modular, and to support unusual materials and outcomes.

7.2. Production *vs* Practice

Any interest in fabrication outside the context of industrial production, and a generally broadened idea of where and when tools might be deployed, necessarily encounters the long-running and contentious conversation in the arts and humanities on the relationship between automation and creativity. Especially since the Industrial Revolution introduced new levels of automation, including the mechanical reproduction of decorative artifacts [33], various schools of art philosophy have venerated "hand" work and, by extension, have regarded overly automatic tools with suspicion or disdain (naturally, without a particularly rigorous definition of "overly"). These conversations have complex overlap with analyses of power and the control of production, with industrial automation developed and cloistered as a capital asset that has historically disenfranchised independent creators and devalued creative labor. However, of course, almost all creative processes make use of some kind of tools. Indeed, expert practice is often distinguished by the expert's use of highly-refined personal tools, and partial automation—everything from woodworking jigs to stencils to the use of gridded drafting paper—is frequently and quite uncontroversially used to allow creators to focus on the part of the creation process that is most meaningful to them. In an attempt to reconcile these threads, David Pye [310] establishes a key difference between:

- *Production* work, in which, at the time an object is created, the processes, materials, and aesthetic decisions used to make it are pre-determined to as great an extent as possible and
- *Artisan* (Pye uses "workman") work, in which elements of the creation process and its output can be decided and altered during the creation process by the creator.

Pye explicitly does not assign moral or aesthetic superiority to either of these modes, nor does he draw a sharp line between them. However, he states that there are material outcomes can be produced in each mode that could not be produced in the other. I note that this framing does not distinguish between different kinds of creative ability and goals, nor (because Pye was writing before the wide adoption of computerized production machinery) does it address the possibility of temporal fluidity, meta- and under-determination, and branching *enabled* by computational tools. However, I find production vs artisanship an intriguing framework to motivate the roles of interactivity and creator agency in creative processes, and to help identify "production"-oriented priorities (such as efficiency and precise replicability) which may be less relevant in "artisan" or other contexts.

7.3. Computational Craft

In including contexts for computational fabrication that go beyond "production" needs, and particularly in focusing on textiles, I overlap with work done under the banner of "computational craft." "Craft" is a contentious and overloaded term [268]: at some times, it evokes a mysticism or ineffability; at others, it shades into dismissive or even derision (especially when applied to female-coded techniques like sewing). In the "computational craft" context I summarize it as the set of fabrication processes available to and practiced by not-necessarily-expert hobbyists, with an emphasis on applied arts such as sewing, woodworking, origami, and metalwork, and with especial emphasis on those commonly practiced by women and girls, such as hand-knitting, quilting, and beadwork.

In 2000, Blauvelt et al [35] grouped "computational craft" approaches into three stated categories: 1) software applications for planning craft objects; 2) craft media which integrate electronics; and 3) systems for designing with heterogeneous collections of materials, such as a tool for building electronic crank automata. The authors also note some ways in which computational craft brings computing *concepts* into craft domains; for example, by applying the lens of programming language analysis to craft notations such as origami folding instructions. Much of the work in this dissertation could be grouped into a broad version of their first category—computational concepts and tools for reasoning about and producing objects—with some e-textiles results (category two), such as in Section 5.9, Section 6.6, and Subsection 13.3.2.

However, these categories are fundamentally one-directional, with the frankly patronizing assumption that the "craft" aspects will be improved by the "computational" ones. Buechley [46] and Pearce [293] note that the reverse is quite possible: concepts and norms from the crafting context, including social identity, open collaboration, political activism and subversion, and sustainability, might be usefully brought into computing practice. For example, hybrid fabrication tools can center the role of artisanal knowledge [437], and craft-based activities can give resonance to a digital narrative [13, 368].

My work is aligned with HCI research which explicitly centers crafting concerns, such as hybrid woodworking systems [223, 331, 436], augmented hand tools [423], and, in textiles: the AdaCAD system for designing woven e-textiles [100], the eLoominate system for scaffolded hand-knitting [114], Smith's work on generative quilting and embroidery patterns [357], and Igarashi & Igarashi's line of inquiry into tools for specific handcraft practices like plush toy design [148], crochet [150], and dimensional beadwork [149]. Notably, these works seek to augment creative craft practice but not to obviate it; especially in the Igarashis' research, the value of engaging in a handcraft practice is assumed. In a computational crafting work on hand-sewn smocking embroidery, Efrat et al describe a "hybrid bricolage"—a process of assembling conceptual modules into a desired final output [80]. Users of their system browse a "catalog" of smocking patterns and recombine them in a "bricolage" that is deliberately based on trial-and-error instead of predictive simulation. These computationally mediated practices are not simply about efficiency or precision, or even necessarily about supporting "novice" creators, but are about developing qualitatively different experiences and outcomes from unmediated craft practice.

7.3.1. Social Computational craft

Social connection is a deeply entangled context of craft practices. Physical crafts such as sewing or building miniatures are often taught and practiced socially—for example, from parent to child, between children at a summer camp, or in hobby "guilds" like quilting circles [340]—and online craft communities such as the knitting and crochet site Ravelry are well-populated [94] despite the challenges of representing physical hobbies in virtual space [189]. Social activities in crafting include working adjacently [128] and collaboratively [323], gifting crafted artifacts [196], aligning with community aesthetics [128], and exchanging feedback on craft processes and outcomes [383]. These activities are particularly relevant in the context of long-term engagement with learning a skill or maintaining a practice.

Within HCI research, the social context of crafting is the basis of Spyn [325], which allows hand-knitters to tag their knitting with machine-readable metadata; users of the system attached a range of media from audio messages to photographs of their surroundings as they worked. The narratives of the works ranged from travelogues to music "mix-tapes," and they encoded social hopes, mundane thoughts, and personal reflections on their knitting design—all reflections of the knitter's personal context during the process.

In this dissertation, the craft and craft-like practices I reflect include the value of hands-on labor, the possibility of an ongoing personal relationship with a material process, and the mutually beneficial entanglement of fabrication and social interaction (in this Part, as well as in the discussions with creators in Part III).

7.4. Textiles and the Fiber Arts

While "crafts" (as discussed above) and "the fiber arts"—felting, weaving, knitting, sewing, knotting, braiding, and an enormous abundance of related textile techniques—are heavily associated in the popular imagination, they are not synonyms. Additionally, most textiles are created industrially, at massive scale. (Textiles had an approximately trillion-dollar global market size in 2023 [113, 375].) Creating with fabric can be an expressive outlet for a teen fashion designer, and it can also be a means of producing highly engineered material at a range of scales from tiny braided surgical stents to geotextiles which cover entire hillsides. The production of textiles is an ancient form of technology which significantly predates recorded history. For example, physical evidence has been uncovered that points to the use of twisted fibers even by Neanderthals [125], and other direct evidence suggests that weaving dates to at least 25,000 years ago [359]. Sophisticated textile techniques have been refined across human culture; specific technologies can be a cornerstone of a cultural identity, from the syncretic print designs of the west coast of Africa [85] to the constructed mythology of aran knit sweaters [332]. The "fiber arts" can be associated with deep artisanship, including complex tooling and dense jargon, and can be an expressive medium for artistic expression.

7.4.1. Fiber Arts and Computation

While it is not particularly accurate to say that "the first computer was a loom" this popular memetic idea [21, 161] reflects that historically, influences between the computation and fiber arts, especially weaving, have been complex and bidirectional [235, 326]. Ada Lovelace famously used a weaving metaphor when describing the complex possibilities of mechanical computation: "We may say most aptly that the Analytical Engine weaves algebraical patterns just as the Jacquard-loom weaves flowers and leaves" [217]. The use of punched cards as program storage for weaving, as initially developed by Bouchon in 1725 and popularized by Jacquard in 1804 [306], influenced Hollerith's design of census-tabulating equipment [273] and thereafter the use of punched cards in general-purpose electronic stored-memory computing [61].

Program storage can itself be textile. Inca khipu, at "the height of their development" by the 1400s CE, encoded data through knot topology and yarn characteristics [23]. Much more recently, magnetic-core memory was developed that stores non-volatile random access memory in magnetic cores at the interlacements of a woven plane, and core rope memory encodes its read-only data directly in its carefully-crafted physical braid [352]. These were simultaneously cutting-edge computational technology, e.g. in their role as part of the US space program, and also often explicitly associated with the hand-weaving needed to produce them [326], with the core rope memory casually referred to as "Little Old Lady" memory (regardless, of course, of the age, gender, or size of its maker [352]).

Beyond pure function, the forms and aesthetics of fiber arts can shape computation. Susan Kare drew on her experience with counted-stitch embroidery to design the literally iconic pixel art of the Apple Macintosh [180]. Irene Posch's Embroidered Computer implements a deeply material-driven and stunningly beautiful functioning computer entirely from relays constructed with hand-embroidery and magnetic beads [305]. Smaller-scale e-textile works have investigated how computation can be stitched with heritage fiber arts like goldwork embroidery [164], Turkish needle lace [410], or honeycomb smocking [10, 127].

In the other direction, computation has of course had major effects on the industrial production of textiles, while individual fiber artists have incorporated computation and computational thought into a wide range of projects. Algorithmic approaches to knitting and crochet include embedded ciphers [291] and procedural hyperboloid surfaces [230, 239]; within weaving, contemporary practices incorporate laser-cutting [11, 45, 413], glitch [231], and the use of Photoshop to process pixel imagery into complex weaving patterns [338]. The introduction of weave planning software and even home electronic dobby looms in the 1980's spurred new capabilities [411] and even new aesthetics of handweaving [337]. (I discuss the landscape of computational weaving tools in Section 8.3.)

All of this comes together as an incredibly multifaceted context for textiles, and especially for weaving, which is the focus of this Part. I play with the cultural associations of weaving, juxtaposed against the context of online streaming, in Section 10.6, and I draw on specific technical practices in weaving throughout, notably in translating Schlein's Photoshop-based digitization technique [338] to a real-time photographic shader in Section 10.4 and supporting double-cloth structures in Subsection 9.3.1.

7.5. Aesthetics of Computation

In this Part, I consider not just the materiality of woven fabric, but also the materiality of computing. While theoretically any forms could be represented in computation, a subset of these have been particularly developed by artists and practitioners. The digital humanities include everything from computer music [129, 224] to poetry generators and hypertext [402] to algorithmic visual art; the inherent fungibility of computation has made discussions across and between these areas possible and influential. In this work, I primarily focus on visual culture and patterning, but the related work does not necessarily fall neatly into this category: e.g. it includes applications of live-coding — typically applied to performance-based work like music — to weave pattern generation [57] and conversely, the use of textile objects as a basis for generative music [321].

Computational media often builds on conceptual groundwork laid by instructional artists, such as Sol LeWitt [104] and Yoko Ono [276], who specified visual or performance works textually to be interpreted by gallery installers or directly by the audience. Early computer graphics pioneers such as Vera Molnar [243] developed methodological tactics, such as variability and multiplicity [320], that continue to influence computational visual culture. (Amusingly, Molnar characterizes her work with an IBM 370 computer as serving "to minimize the effort required during the preparatory phase of making a picture," which may seem to contradict the production/practice dichotomy I discuss in Section 7.2. However, it is clear from the surrounding text that she means that the advantage of the computer is to work through a multitude of possibilities, in "not only the stepwise approach toward the envisioned goal but also sometimes the transformation of an indifferent version into one that I find aesthetically appealing" [243]—in other words, as a tool to aid the exploration of a full space instead of a single output.)

In addition to computational methodologies, computational *aesthetics* have arisen, often from algorithmic models of natural phenomena such as Voronoi and reaction diffusion algorithms [216, 288]. The method for synthesizing Perlin noise was published in 1985 [300] and has become ubiquitous These visual algorithms are recognizable not necessarily in their specific forms, which are inherently malleable, but in their generational logic and visual style; they are instantly perceivable as "computery" (despite the technical possibility of making them physically, in the case of Voronoi and reaction diffusion figures).

In this dissertation, I particularly relate to two aspects of computational media practice: procedural and emergent generation, and transcoding and glitch.

7.5.1. Procedural Generation

Procedural generation could encompass any production methodology in which the output is designed indirectly, by designing the higher-level production processes. Such a methodology is not necessarily uncertain—for example, a fully deterministic generator could be used solely to save disk space in a computer game [169]—but it is often referred to as "emergent" when the results are perceived as "greater than the sum of the inputs": when there is unpredictability or complexity beyond what the designer explicitly encoded. (While mathematically "true" randomness cannot arise purely from computation, perceptual randomness is an altogether different quality [314].)

Art and artisanal processes have a long history of working emergently [356]. A unifying motivation in many of these methods is the possibility of producing results not directly intended by the creator, often pushing the creator to take the work in different directions than originally envisioned. Modern art movements such as Dada and Fluxus looked to mechanics like random instruction cards and improvisational games to provoke and complicate their work [70, 197]. Western interpretations of tie-dye and raku pottery glazing center surprise, and flexibility in interpreting a result [360].

Procedural generation is currently deployed broadly, in domains ranging from architecture to online product descriptions (though these are increasingly delegated to "generative AI," which is outside the scope of this discussion of procedural aesthetics because it relies less on the algorithmic process and more on the dataset); for an example in textiles design, Knit Yak produces machine-knit scarves with patterning based on mathematical rules of elementary cellular automata [347]. Crafted parametric spaces in digital fabrication tools often seek to computationally optimize structures for robustness or ease of production, e.g. Forte [420].

The topics of possibility spaces and parameterization are fundamental to the field of procedural generation research, which has long been concerned with shaping the possible outcomes in systems which are both highly automated and highly chaotic. Particularly in Chapter 10, I draw from Karth's overview of the dimensions of procedural generation, which primarily cites examples from game design and net art [169]. Karth documents the *poetics* of generative systems—that is, "what it means when we use a particular form of generation and what effect it has on the player"—and classifies generative systems along several properties. Particularly relevant to the fabrication context are the properties of *form* (in fabrication, likely to be an artifact or part of one) and *locus* (the user's interactions with and perceptions of the system itself), each with its own gestalt aesthetics. Karth's writing provides a helpful frame in thinking about artifact and experience in interactive fabrication, as he highlights that it can be difficult to draw a line around "a single output" in any generative system. Aspects like temporal scale and the "surface" of the generator can affect this: for example, a Twitter bot might be encountered either as a single daily message in one's feed or as a collection of tweets in a book; the latter form encourages understanding the output not as individual messages but as an oeuvre. Drawing these lines is even more complex for an interactive system, where the "output" could either be a single system contribution—one modification of the in-progess item—or the final collaborative result. I discuss this framing, and the need to consider of interactive fabrication at multiple physical and temporal scales, in more depth in Subsection 10.7.1, and it directly inspired the improvisational "Paths" interface in Chapter 13.

7.5.2. Transcoding and Glitch

In his writings on computation and creativity, Hofstadter circles around the idea of parameterization as a lens for thinking of creativity [135]. (Hofstadter calls a possibility space an "implicosphere": an implicit counterfactual sphere.) He describes a series of metaphorical knobs, with the slight tweak of a knob—"slippage"—revealing not only a new outcome, but some insight on how the knobs themselves might be defined. This idea can be akin to "glitch": an act of formal destructuring, or minor destabilization, which surfaces an underlying structure [238]. Technically, most glitch practice is formed around the process of slightly modifying a digital file in a way unintended by its software specification. For example, the bits of an image file can be edited with a text editor, with results that can be difficult to predict [237]. On a higher level, glitch relies on a fundamentally digital process of transcoding: the bits of a file can be interpreted as either pixels or text, and the process leverages the gap between these interpretations. Transcoding can also occur across domains. For example, artist Robin Sloan describes what he calls "the flip-flop": a process, either by an individual or at a broader cultural level, of transcoding physical artifacts to digital data and back again, which results in creative explorations that could not be conceptualized otherwise—for example, dance moves inspired by video footage that skips or reverses [354].

Manon notes that glitch is by definition digital, software-based, based on copies, and therefore only "simulated risk," remaining "low-stakes" despite its "un-tame" appearance [225]. However, glitch can be thought of more broadly as a deliberate introduction of errors, which is of course very possible in fabricating physical objects. In 3D printing, deliberate over- and under-extrusion has been used to produce semi-predictable tactile objects in filament [204] and clay [351] printing. Glitch visuals can also be generated in the usual way and applied any medium which can represent pixels, such as weaving [231, 367] and knitting [339].

I explore specific processes of transcoding and glitch in Chapter 10 and Chapter 14.

8. Fabrication Technique: Weaving

Simply defined, weaving is the process of making a fabric out of interlaced sets of threads, Figure 8.1; typically a *weft* thread is passed over and under a tensioned *warp*. This process, with its myriad variations, is one of humanity's oldest and most widespread technologies.

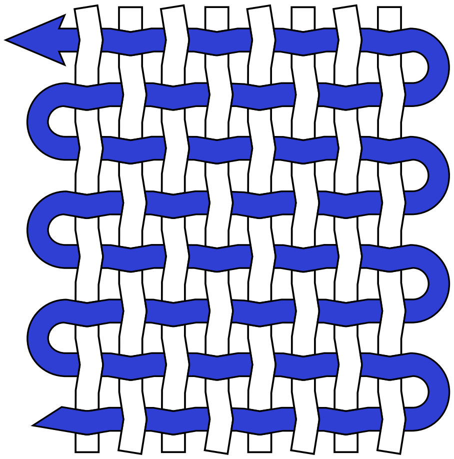


Figure 8.1 *A basic woven swatch: a weft yarn (shown in blue) interlaces with warp yarns (white).*

Woven fabrics include the denim in jeans, wool and linen suiting fabrics, the taffeta of a ballgown, the ripstop materials used in sporting gear such as tents, the fiberglass, carbon fiber, and kevlar fabrics used in resin composite materials, and many furnishing fabrics such as most upholstery, carpeting, and curtains. Most woven fabrics have two axes of tensile strength (along the warp and along the weft); unless the underlying yarn is stretchy, a woven fabric will resistant to strain along these axes.

8.1. Looms and Jacquard Weaving

As a fabrication technology, weaving is remarkable for its range of possible tool setups, from the simplest hand-held peg or backstrap loom to multi-story industrial powerlooms. A loom is any device that simplifies the weaving process, from very low-tech pin looms such as one a child might use at summer camp, to highly automated, high-speed computerized production looms used industrially. As Bauhaus weaver Anni Albers describes, "any weaving, even the most elaborate, can be done, given time, with a minimum of equipment. The main incentive, therefore, for perfecting the weaving implements has always been that of saving time." [17].

A loom supports weaving by, at a minimum, holding a set of threads (the *warp* threads) parallel and at tension. This task can be quite straightforward; minimal solutions include the peg frames common in tapestry handweaving as well as simply tying the warp between the weaver's body and a nearby stationary object (a "backstrap loom"). Most production looms include rotatable beams for holding a longer warp and managing the fabric as it is woven.

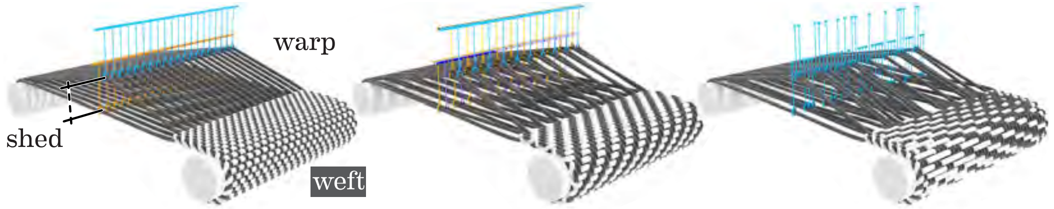


Figure 8.2 *Looms can be categorized by their range of patterning complexity. Left: an abstract loom with two selectable subsets of warp threads; every warp thread is either allocated to the first frame (shown in blue) or the second one (orange). Without manually overriding the pattern, this is just enough to produce the "plain" (checkerboard) weave. Center: with four frames, more patterning variants are possible. In this case, the warps are assigned in a repeating incrememeenting sequence to the four frames (weavers would call this a "straight draw"); two of the frames have been raised for each frame of the weaving, again in a repeating incrementing sequence, to produce a "twill" pattern. Right: a fully Jacquard loom allows any warp thread to be selected independently of the others. In this case, a "shaded satin" pattern has been woven, with a more weft-dominant pattern one one side of the loom subtly shifting to a warp-dominant one on the other.*

Second, the major category of looms that is relevant to this work can selectively raise or lower some of those warp threads, forming a gap known as the *shed*, allowing the *weft* to interlace quickly without the weaver manually threading it over and under each warp. This task can be accomplished by a great variety of mechanisms. Simpler looms, like most hobbyist looms, can only select amongst a number of sets of warp threads determined by the quantity of "frames" (or "harnesses"/"shafts" depending on the specific mechanism) available on the loom. For example, a very basic loom might support only two sets, Figure 8.2. A weaver might allocate all the even-numbered threads to one set, and all the odd-numbered ones to the other; at weaving time, that weaver can only alternate selection between those two sets, producing the "plain weave" structure shown in Figure 8.1. A more typical number of frames is four or eight. Frame-style looms occasionally (but rarely) have up to twenty-four frames, at this high end often requiring the support of an analog or electronic "dobby" mechanism to keep track of which sets should be selected in a row. Designing a weaving pattern for frame looms is an expert task. Notably, the patterns which can be woven are constrained by the loom's setup; changing the setup is labor-intensive and is typically only done when an entirely new fabric is begun.

In contrast, a "full Jacquard" loom is one in which every warp thread can be selected individually. This style of loom is commonly attributed to Joseph Marie Jacquard, whose 1804 model used punched cards to control the selection of each pattern row [86]; related concepts were considered even earlier, including inventions by Basile Bouchon in 1725, Jean Baptiste Falcon in 1728, and Jacques Vaucanson in 1740 [306]. These increasingly automated weaving machines formed the functional and symbolic heart of the early stages of the Industrial Revolution. The Jacquard loom in particular holds an important place in the history of computing by including both a stored numerical representation of a task, and a mechanism for interpreting that representation to automatically carry out a sequence of actions [86, 316]. Electronic versions of the Jacquard loom, with computer-controlled solenoids standing in for the punched cards, are used heavily in today's global textiles industry. Industrially, the flexibility of Jacquard warp selection is valuable because a different weaving pattern can be deployed without the setup overhead that would be required to change patterns on a less-complex loom. (This setup time can be measured in days or even weeks for an industrial machine.)

More broadly, this fundamentally computational approach allowed the Jacquard loom to escape from the confines of a limited set of actions (in this case a pre-determined set of shed patterns) to a more general and complete capability, dramatically expanding the range of expression to include all possible shed patterns.

8.2. Production and Handweaving

"Jacquard" simply refers to support for computational patterning in warp selection, but Jacquard looms are often used in industrial production and are thus automated in other ways as well. Industrial looms are expensive, large, and as fully-automatic as possible; in addition to Jacquard selection and mechanically-driven shedding, an industrial loom inserts and compresses the weft, manages spooling and un-spooling the warp and the cloth as it is formed, and may even post-process the fabric in specialized applications (e.g. cutting apart velvet [392]). Contemporary industrial Jacquard looms have been refined over centuries and under a tremendous amount of economic pressure to produce consistent fabric as quickly as possible. Such looms are capable of truly wild weaving speeds, such as 1200 picks (selection and weft insertion cycles) per minute [275].

At the other end of the scale of automation, a *handloom* requires a weaver to insert and compress the weft, as well as to attend to other matters of craft such as managing tension (to a greater or lesser extent depending on the specific loom). A handloom can produce a much wider range of fabric types than a specialized industrial loom, and the interactive process of weaving offers opportunities for creative improvisation and learning. However, handlooms that are suitable for prototyping, experimentation, and playful or meditative use may not support complex patterning, or may require a great deal of expertise or effort to achieve these effects.

Striking a balance between the patterning complexity of an industrial loom and the material flexibility of an artisanal one, Digital Norway's TC2 Jacquard Loom [78] and AVL's Jacq3G [7] are commercially available looms which combine computerized Jacquard selection with otherwise mostly-manual control. (They do both have optional features for assisting with tensioning and fabric take-up comparable to other large handlooms.) These are used in "production prototyping" contexts, as well as by textile design students and very serious hobbyists.

8.3. Software Tools for Weaving

Handweavers have embraced computer-aided design tools since the dawn of personal computing [338], and the contemporary handweaving ecosystem includes social networking sites [280], mobile apps [335], and the use of Photoshop as a weave planning tool [338]. Professional weaving design software such as Pointcarre [99] and the Arahne suite [302] includes features such as generating weaving patterns from photographs, parametric design for standard patterns like plaids and stripes, rendering weave swatches into product mock-ups for internal use or advertising, and interfacing with industrial computer-controlled looms. These tools primarily position weaving design as a graphical composition task, with support for tiling, repeats, and colorway variants. The conversion to woven structures is largely done through applying sub-structures from libraries of possibilities, though they allow thread-level changes for expert users. Lastly, the company WOVNS [68] offers a weave-on-demand service scaffolded by their own weave design platform, which uses a similar approach but with a much smaller, curated set of structure palettes to reduce the complexity for the designer and lower production overhead.

Many of these tools are either based on Photoshop-like pixel manipulation or a traditional weaving notation called a "draft." As a representation of the relationship between a surface of fabric, the allocation of warp threads to the loom's frames, and the row-by-row instructions for producing a fabric [394], drafts are particularly suited to hand weaving on a frame loom; they are less helpful for fully manual or fully jaquard weaving, or for fabric structures with interlocking layers or complex routing requirements. Pixel manipulation is well-suited to converting graphic elements for Jacquard weaving, but it requires constraining the available design space in ways that may be difficult for non-experts to understand.

Contemporary artisans working in "complex weaving" (a loose term that covers a variety of technically intricate woven structures, including multilayer and shape-changing fabrics) have highly developed personal notations and workflows for reasoning about and producing their patterns [77]. These may draw on historical ways of thinking about woven structures (e.g. named types of fabric), but are also often experimental and idiosyncratic.

8.4. Weaving in Human-Computer Interaction Research

Within HCI, weaving has been explored in many contexts [73], including as a technocultural phenomenon [324], as a site for personal expression and reflection [37, 277], and as a fabrication technique for functional materials. In the last category, weaving has particularly been explored for e-textile applications such as sensors [145, 188, 370] and displays [34, 74, 174]. The AdaCAD system provides a structure-based representation of weaving, allowing the user to track specific yarn connections and separate woven layers, which is particularly important in e-textiles work [100]. In response to the possible wastefulness of e-textile materials, the Unfabricate system supports the production of woven artifacts designed to unravel for material reclamation [417] Researchers have also studied the fabrication possibilities of "3D weaving"—the construction of dense objects as stacked interlaced layers—and created special-purpose tools to help users manage the complexity of designing for this fabrication method [126, 416].

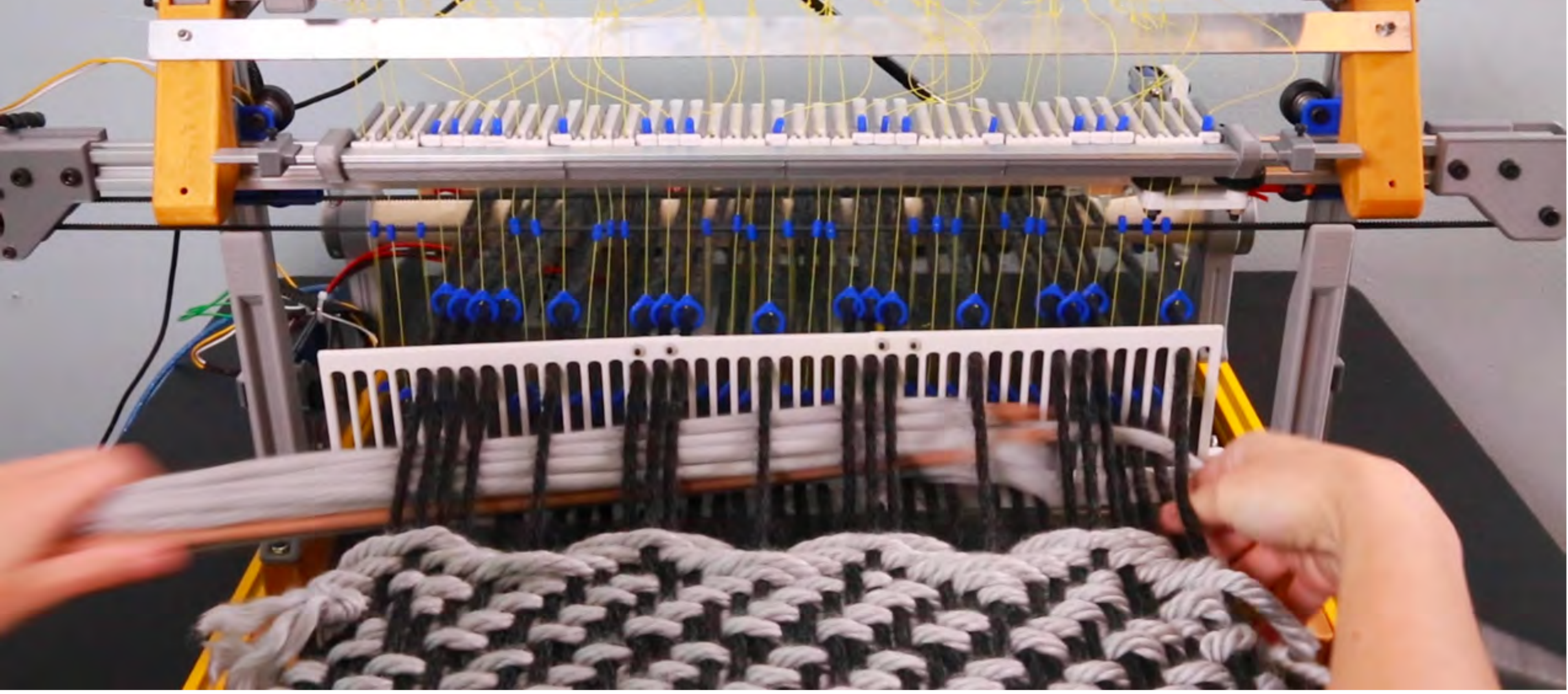
The wide-ranging research interest in weaving motivates the development of tools and infrastructure for experimental weaving, especially with complex structures and/or unusual materials.

Personal Practice in Computational Handweaving

In Chapter 9 and Chapter 10, I present two explorations of blended computational and handcraft practice.

Each describes an engineering intervention that underpins a set of “sketches” combining computational modalities with handweaving. In Chapter 9, a novel portable and inexpensive Jacquard handloom becomes a site of interactive, networked, and ongoing day-to-day weaving. In Chapter 10, I show how reverse-engineering the control protocol of a powerful Jacquard handloom enabled flexible integration with video processing, computer vision, and the Internet and prompted questions about procedural generation under the resource constraint of manual weaving.

This work offers shifting roles for a creator within a computational fabrication system: as the designer of a procedural system, as a participant in the weaving process, even as a remote collaborator. The work is situated in the **blended contexts** of the fiber arts, of computation, and of personal practice. Each contributes tendencies and opportunities, from the computational logics of woven interlacements, such as double-cloth or shaded satin weaves, to the modular malleability of a data stream.



9. A Personal Jacquard Loom

In this chapter, I discuss the design and engineering of a loom that was conceptualized as a *personal* loom, with the intention of serving a range of end users including artisans and researchers, whether for exploring expressive weaving or prototyping e-textile or composite material applications. This framing offers a set of constraints and goals that are quite different from industrial weaving: it almost entirely de-emphasizes high speeds and autonomous operation, instead prioritizing a low price point, portability, safe operation, and a novice-friendly interface.

For maximum flexibility, the loom is a Jacquard handloom. As I explained in Section 8.2, "Jacquard handweaving" blends computational patterning with manual control of other aspects of the weaving process; the two commercially-available models are Digital Norway's TC2 Jacquard Loom [78] and AVL's Jacq3G [7], and they are popular for prototyping, learning, and artisanal weaving. However, these looms largely replicate the mechanical logic of industrial weaving. As a result, they are expensive and large—the TC2 has a baseline cost of approximately $30,000 USD for their smallest model, which is 61 inches tall with a 58 inch by 48 inch footprint and additionally requires an air compressor—and thus impractical outside of dedicated spaces such as fashion schools and textiles-specific laboratories. Additionally, even these relatively simpler looms are still complex and difficult to maintain, and their mechanical complexity scales with the width of the fabric (counted in number of threads).

In a home, laboratory, or workshop, however, factors like speed and tightly-tuned repeatability are much less important than storage space, setup time, and cost. Indeed, handweaving is not just a concession made to allow less expensive build; it's an approach that makes it possible to modify a project on the fly, incorporate unusual materials, and even learn about weaving.

Loom	Selection Type	Size
AVL Jac3g [7]	parallel Jacquard (solenoids)	floor
TC2 [78]	parallel Jacquard (solenoids)	floor
OSLoom [32]	parallel Jacquard (muscle wire)	floor
Little Weaver [5]	18-24 frames	tabletop
Liou [208]	parallel Jacquard (motors)	tabletop
Moyer [249]	parallel Jacquard (solenoids)	tabletop
Schaefer [336]	serial Jacquard (cams)	tabletop
Nicholls [266]	serial Jacquard (Lego™)	tabletop
Ruta [278]	manual Jacquard pins	tabletop
SPEERLoom [361]	parallel Jacquard (linear stepper motors)	tabletop

Table 9.1 *Contemporary computational handlooms. Entries in bold are commercially available; other entries are personal, research, or incomplete/deprecated projects.*

9.1. Contemporary Handlooms

Table 9.1 summarizes the landscape of contemporary computational handlooms. "Floor" looms are large, standalone machines typically intended to produce full-width fabric, whereas tabletop looms are small and deployable. The form factor has implications for both price and intended use.

Two full Jacquard handlooms are currently commercially available: the Jacq3g loom by AVL Looms [7] and the TC2 loom by Digital Weaving Norway [78]. Both are "prosumer" floor looms intended for either production prototyping or serious artisanal practice, and both use a parallel selection mechanism similar to industrial looms. The Jacq3g uses solenoids to displace hooks onto a lifting bar; the baseline model has a 45 inch by 84 inch footprint (90 inches tall) and costs approximately $28,000 USD. The TC2 uses electronic valves to direct air from a compressor; the baseline model has a 58 inch by 48 inch footprint (61 inches tall) and costs approximately $30,000 USD. The OSLoom project was Kickstarted in 2010 with the intention to build a floor loom using parallel muscle wire selection. While a full OSLoom was never built, the approach has similar power, size, and electrical complexity drawbacks for home or casual use.

AVL's smaller "Little Weaver" approaches ours in intent, as it is a portable (tabletop) computational loom which is designed to easily integrate with a home computer setup, such as by allowing Wi-Fi connectivity [5]. However, the Little Weaver is a frame loom with eight, sixteen, or twenty-four frames, not a full Jacquard loom. At the time this Personal Jacquard Loom work was published, in 2021 [16], AVL had stopped making Little Weavers; I said at the time that "when they were last sold, Little Weaver looms cost $2,000-3,000, depending on the number of frames [6]." In 2024, a Little Weaver 2 costs $3,600-5,400 for eight to twenty-four harnesses, respectively [8].

Many personal hobby projects have approached loom design and construction. These have been greatly successful in the area of looms which are *not* computationally controlled, such as DIY pocket tapestry looms [87] and small frame looms [45, 413]. Hobby Jacquard looms have been periodically explored, but these projects typically still use parallel selection strategies [208, 249]. Of particular note are two designs which use a serial actuation strategy similar to ours: a device by Schaefer which uses set of circular cams actuated by a shaft moved along the heddles [336], and the Weav3r loom constructed with Lego™ blocks[266].

The Ruta loom by Ooms et al [278] is an interesting byway in contemporary loom design which centers hands-on sensemaking for novice weavers. It is fully Jacquard in the sense that it can select any warp set, but it is entirely non-electronic: the weaver programs each set by manually arranging pins in a frame. We see the manual override capability of our loom as potentially offering similar opportunities for hands-on engagement.

Lastly, the SPEERLoom, built by a team including friends in the Carnegie Mellon Textiles Lab, uses linear stepper motors for parallel selection. It was developed specifically for classroom use and has an individually-tensioned warp system to be as easy as possible to warp, its stepper-motor drive is likely quieter than our solenoid, and its warp thread density (ends per inch) is thrice that of the loom in this chapter. Intriguingly, it also has the potential to do non-binary warp selection, though this potential was not explored in currently-published work. Due to its parallel selection design, the SPEERLoom's cost and electronic complexity would scale with the number of warp ends it supports; the 40-end version in the paper, which is the same number as the loom in this chapter, costs roughly five times as much. It is also worth noting that the SPEERLoom was developed after the work in this chapter.

9.2. Technical Implementation

The main unique design feature of the Personal Jacquard Loom is the mechanism by which it forms its *shed*: the gap between upper and lower warp threads through which the weft passes to create interlacing. Typical electronic Jacquard-style looms form a shed in two stages: *selecting* a number of warp threads, then *raising* (or, depending on other factors of the loom's geometry, lowering) the selected threads. Because they are designed for industrial production weaving, these looms prioritize very high production speeds; to achieve such speeds, industrial looms must do the entire row's selection at the same time, requiring an electronic actuator (typically a solenoid) for each individual warp thread. The number of solenoids, as well as the power requirements and the complexity of the control circuitry, must therefore grow with the number of warp threads.

In contrast, our loom differs from these by performing the electronic selection *serially* using a passive bistable shedding mechanism. With this design, a single solenoid mounted on a drive belt can perform selection on any number of warp threads. This arrangement greatly lowers the initial parts cost, maintenance difficulty and costs, the complexity of electronic wiring and power requirements, and the overall size and weight of the loom. With access to a hobbyist FDM 3D printer, the entire loom can be made for under US$200, and its "table loom" form factor fits on a desk or countertop.

9.2.1. Heddle Design

The heart of any warp selection mechanism is the *heddle*. Each heddle guides the position of a single warp thread; heddles can be selected and raised (or for some looms lowered), taking their warps along with them to form the shed. Heddles are commonly made of wire or string (or thin wood/plastic, in the case of more limited "rigid heddle" looms which can only actuate a fixed pair of shed patterns).

The serial selection design uses an array of string heddles which are fixed at the top and hold their warp threads at the bottom, Figure 9.1.

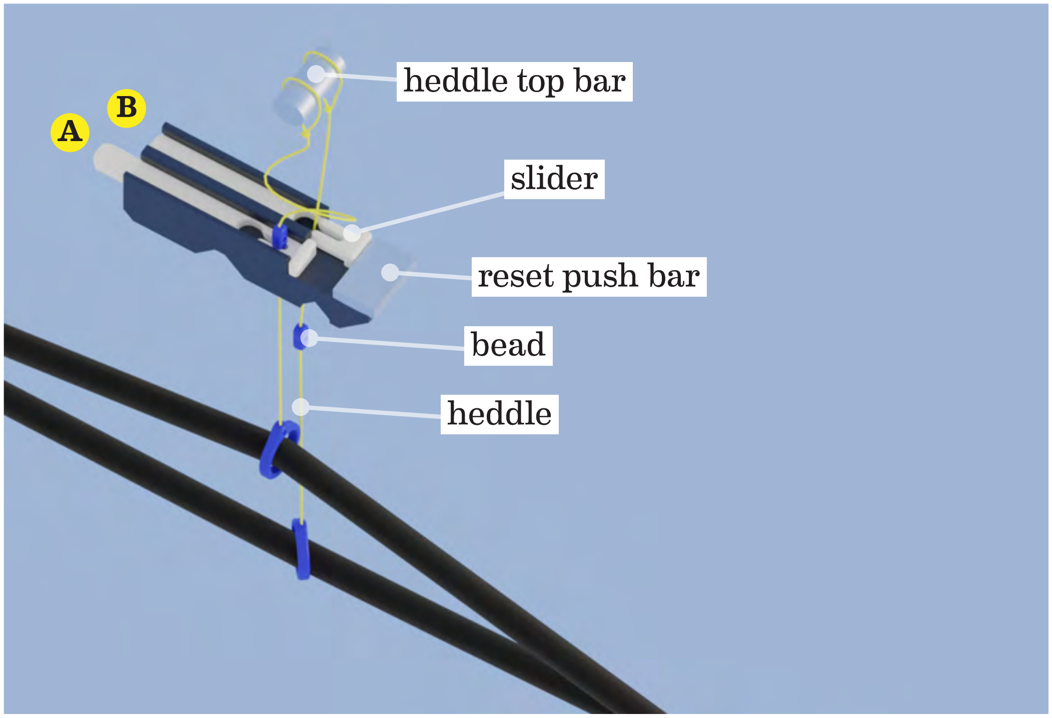


Figure 9.1 *The heddle mechanism. Slider A is in the reset position, with its heddle raised. Slider B has been pushed forward, and its heddle has dropped.*

During the selection process, all heddles are initially raised, lifting each corresponding warp thread and providing a small amount of tension on the heddle. Each heddle has a "bead" which is 3D printed onto it, and each passes through a keyhole-shaped hole in its own "selection slider." At each slider's home position, its heddle's bead is held captive in the narrow slot part of the keyhole. When a slider is pushed forward, gravity and warp tension pull the bead down through the hole, allowing the heddle and its warp thread to drop.

When selection is complete, the warp threads corresponding to each actuated slider will have dropped down, while all remaining warps will remain pulled high, thus forming the desired shed.

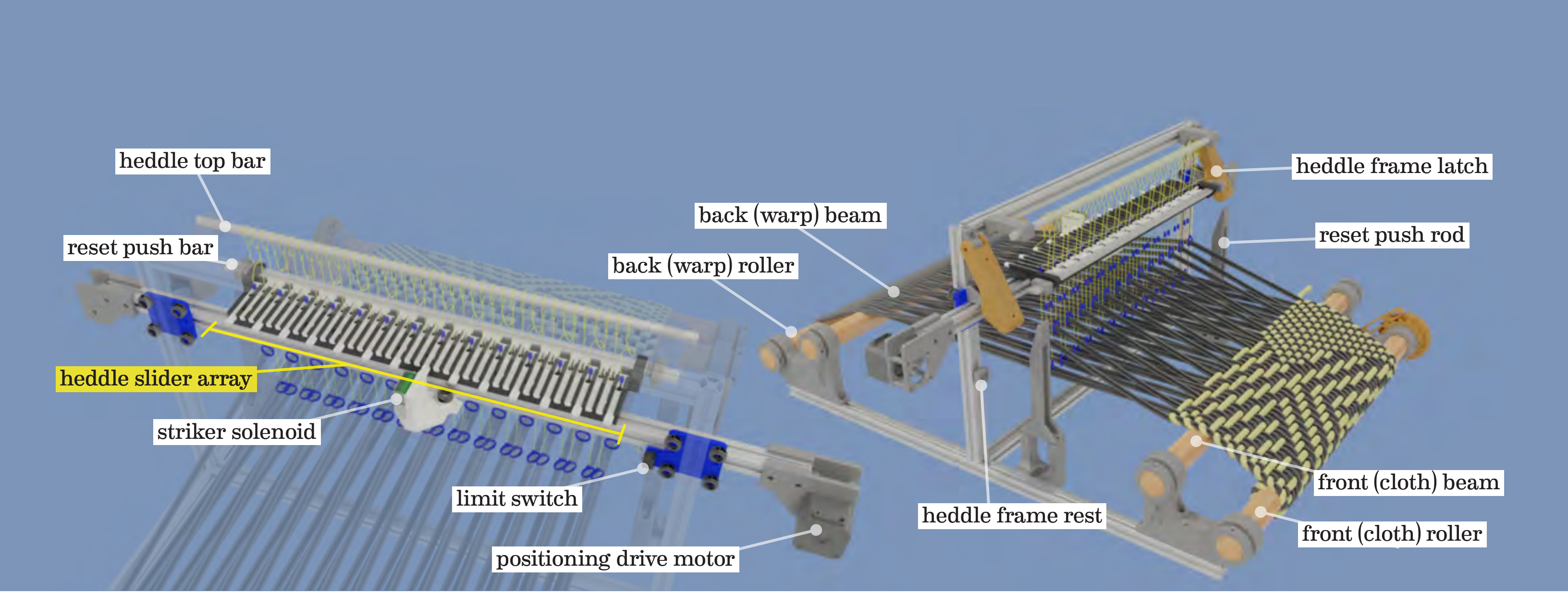


Figure 9.2 *The main components of the loom.*

9.2.2. Frame and Tensioning System

These heddle mechanism can easily be operated by hand, but the *heddle frame* incorporates a stepper motor and belt assembly which positions a solenoid sequentially to strike each selected heddle, Figure 9.2.

The loom's motor and belt assembly are common in 3D printer construction and were chosen to ensure the availability of parts. The loom is designed to be flexible about specific components; e.g. any push solenoid with approximately one centimeter of throw would be compatible, as would any NEMA 17 stepper motor. We specifically used a NEMA 17 bipolar stepper motor with 1.8° step size rated at 0.59 Nm holding torque (2.0A per phase), with a GT2 (2mm pitch) timing belt and a 12-tooth GT2 drive pulley. The belt drives a linear motion platform consisting of 24mm OD Delrin v-wheels with #625 (5 x 16 x 5mm) bearings. The platform rides on the same 20mm x 20mm v-slot aluminum extrusion that the heddle mechanism array is mounted to. The solenoid is an XRN-0530 pull-type solenoid with 10mm throw (rated at 12v 0.3A).

The frame additionally includes a *reset push rod* which, acting with the *reset push bars,* can reset the entire array of heddle sliders. The heddle frame is either held in the upper position by a pair of *frame latches*, or rests in the lower position on *frame rest brackets*.

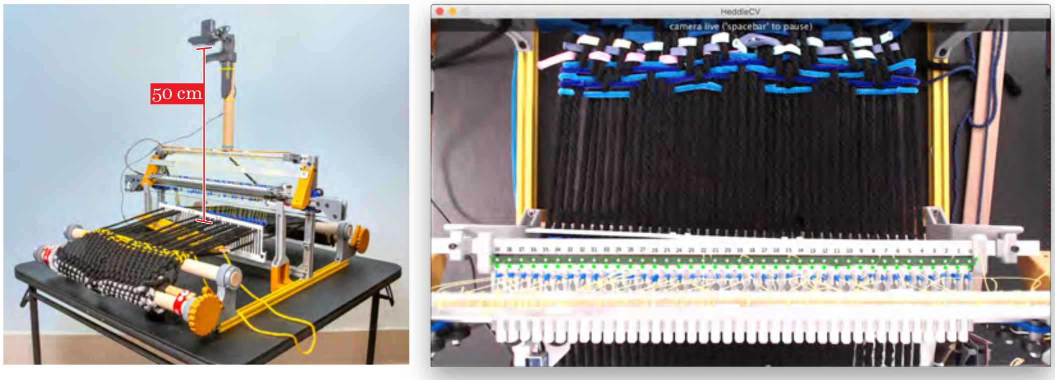
The rest of the loom handles warp and cloth storage and tension, and it closely mirrors established home loom designs. (It may be possible to retrofit an existing loom to reduce build complexity.) Four wooden dowels comprise the *rollers* and *beams*. Warp threads are wound onto a *warp roller* at the back of the loom, pass over the *warp beam*, through the heddles, over the *cloth beam* and are tied onto the *cloth roller*. As it is woven, fabric can be progressively wound onto the cloth roller as warps are unwound from the warp roller. Ratchets on the warp and cloth rollers maintain tension on the warp. Between the heddles and the cloth beam, warp threads pass through a *reed*, which the weaver uses to compress each weft against the fabric as it is woven.

9.2.3. Control Electronics

The control electronics for the loom are provided by the RepRap Arduino Mega Pololu Shield (*RAMPS*) platform [160], a popular and inexpensive choice for hobby projects involving dimensional motor control, such as 3D printers and pen plotters. The RAMPS platform incorporates a programmable micro-controller, power management, stepper controllers, and MOSFET switched high current drives. This board controls a single bipolar stepper motor of the type used in most inexpensive 3D printers, along with a small solenoid (a small "flyback" diode across this inductive load is also added to protect the MOSFET drive transistor from transients). Finally, an endstop switch is connected to the controller board to allow the loom to place the heddle selection head at a known "home" position. All other positions are determined in low-level firmware by counting steps of the stepper motor with respect to this position. Off-the-shelf open source firmware (a specially configured, but unmodified copy of the Repetier v0.92 3D printer control firmware [141, 142]) is loaded into the RAMPS board and accepts G-Code commands over a serial connection to drive the motor and solenoid. We use this system to reliably direct the selection head to exact heddle locations, and selectively power the solenoid to push forward selected sliders.

9.2.4. Input: Computer Vision

An advantage of the slider-based heddle design is that it can be easily overridden: the weaver can manually drop or lift heddles to tinker with the pattern directly on the loom. As a computational system, this tinkering is especially valuable as input: the weaver's choices can be recorded, transmitted, or act as a basis for future patterning. To avoid adding electrical complexity to the loom hardware, we used a webcam and computer vision techniques to sense the heddle positions. We mounted a Logitech C922 webcam to the main loom frame such that the camera lens was 50 cm above the warp, Figure 9.3(a). This position gives the camera a field of view which extends to the front of the loom. (An additional benefit of this input approach is that the weaving area can be used for other visual input as well. I demonstrate this capability in the "On-Loom Input" example described in Subsection 9.3.1.)



(a)

(b)

Figure 9.3 *a) The camera mount. b) The view from the camera. Green lines show the computer vision's region of interest for each heddle slider.*

I use OpenCV to determine the heddles' positions. Within the camera's view, the region of interest is the area just in front of the heddles' home positions, Figure 9.3(b), which will show the color of the bed if the slider is in its home position, or the color of the slider if it is moved forward. To calibrate the system, the weaver clicks on the four corners of this area in a live camera view. (This calibration is persistent across weaving sessions as long as the camera isn't moved.) The system divides the area into a number of perspective-corrected cells corresponding to the number of heddles, then calculates an average pixel value in each cell. These averages are then thresholded with Otsu's method. A cell that is mostly white indicates that the slider is forward, and therefore its heddle is dropped.

The heddle sliders were printed in white and the slider bed was colored black for maximum visual contrast, and I additionally enabled the built-in lighting of the webcam to reduce shadows. These measures, along with the adaptive thresholding, ensure that the system is robust in typical room-lighting conditions.

9.2.5. Control Software and Operation

The main control software consists of several modules: communication over a USB serial port with the RAMPS control board, webcam capture and computer vision, and networking to connect to server-based design tools (such as the "Remote Collaboration" example). For interoperability, weaving patterns are consistently passed as lists of pattern rows, with "1" indicating a raised warp and "0" indicating a dropped one.

For each row of weaving, a Processing sketch fetches the next row of the weaving pattern from either local or networked design tools. (I describe specific example tools in Section 9.3.) This pattern row is converted to G-code commands which sequentially position the selector solenoid at each heddle which should be dropped, then strike with the solenoid to hit the slider and drop the heddle. The weaver may then modify the shed as desired, and the computer vision system can record these alterations and communicate them to the design tool. Then, the weaver passes the weft of their choice through the open shed, and uses the reed to "beat" the fabric, pushing the latest weft into the fabric being formed.

To reset all the heddles, the weaver lowers the heddle frame. After the sliders have passed below any dropped heddle beads, the reset push-rods engage the reset bar to push all sliders back into their home positions, trapping each bead above its slider. At this stage, the weaver also periodically takes up the completed fabric onto a roller at the front of the loom and releases an equivalent length of warp threads from a similar roller at the back of the loom. Finally, the weaver raises the heddle frame again to prepare for another row of weaving.

Figure 9.4 *Weaving can incorporate many materials: pipe cleaners, balloons, and strips of fabric are shown here as improvisational responses to generated weaving patterns seeded by daily weather conditions (described in Subsection 9.3.3).*

9.3. Personal Weaving Systems

The low cost and small footprint of the loom makes it suitable for personal weaving practice. I describe three systems which make use of specific capabilities of tabletop Jacquard handweaving. The software interactions align with the principles guiding the loom itself: flexibility and extensibility. This approach aims to serve a range of end users including artisans and researchers, whether for exploring expressive weaving or prototyping e-textile or composite material applications.

9.3.1. On-Loom Input

An advantage of handweaving is the ability to incorporate disparate materials; Jacquard handweaving in particular can use “double cloth” patterns to form two layers of fabric which can act as a pocket.

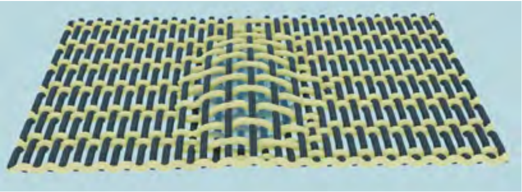


Figure 9.5 *Double cloth weaving.*

Double cloth is a pattern type in which an area of the fabric has two separate faces, Figure 9.7. In this area, half of the warp threads are allocated to one face, and the other half are allocated to the other; the weft only interlaces with one face per pass. Double cloth patterns are a popular way to produce areas of solid color [246], as well as, in the hands of a skilled weaver, a way of producing complex intersecting topologies [272]. However, designing a double cloth pattern can be difficult even for intermediate-level weavers, especially if each fabric face itself has complicated patterning. Additionally, when woven on a frame loom, double cloth patterns require twice as many frames at the same complexity of face patterning; the frame quantity requirement rises if the doubled area is not rectangular. Thus double cloth is an ideal demonstration of the flexibility of computational Jacquard capabilities.

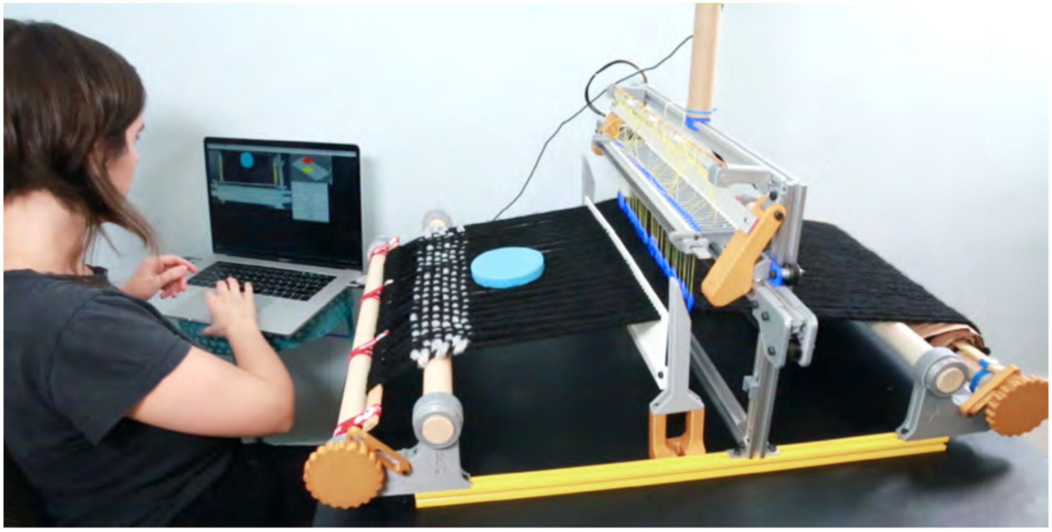


Figure 9.6 *The weaver places an item on the loom. The system captures and rectifies the outline of the item. The weaver allocates diferent weaving patterns to the two faces.*

We show a system in which an object to be incorporated (e.g. a PCB for an e-textiles project) is laid directly on the not-yet-woven part of the warp, Figure 9.6. The image is captured by the same camera that handles the heddle input, and it is rectified to represent the real-world size and shape of the object. The weaver can then apply weaving patterns to all three areas of the fabric: the single-layer area, and the front and back faces of the pocket, Figure 9.7(a). These patterns are then computationally composited for weavability to create the weaving pattern, Figure 9.7(b). In addition to the elaborate structural patterning enabled by Jacquard weaving, this system takes advantage of the loom's integrated calibrated camera system for capturing real-world input to the design process. In this case, this input allows the weaver to incorporate physical objects without the need for measuring, a hallmark strength of on-machine interaction systems.

(a) (b)

Figure 9.7 *a) A screenshot showing the camera's view of the weaving as well as the pattern allocator. The weaver has assigned a twill pattern to the upper layer and a plain weave to the lower. b) A diagram of double cloth weaving showing the "pocket" formed in one area of the cloth. When actually woven, each weft line would be compressed against the previous ones, resulting in the outcome shown in Figure 9.5.*

9.3.2. Collaborative Editing

Computational mediation allows a creative process to have remotely-accessed components. We show a system in which the current weaving plan can be accessed in real time via a web server. The weaver and any number of remote collaborators can edit the plan using a web-based editing tool, Figure 9.8; edits are shared in real time, as in other online collaborative editing tools such as Google Docs.

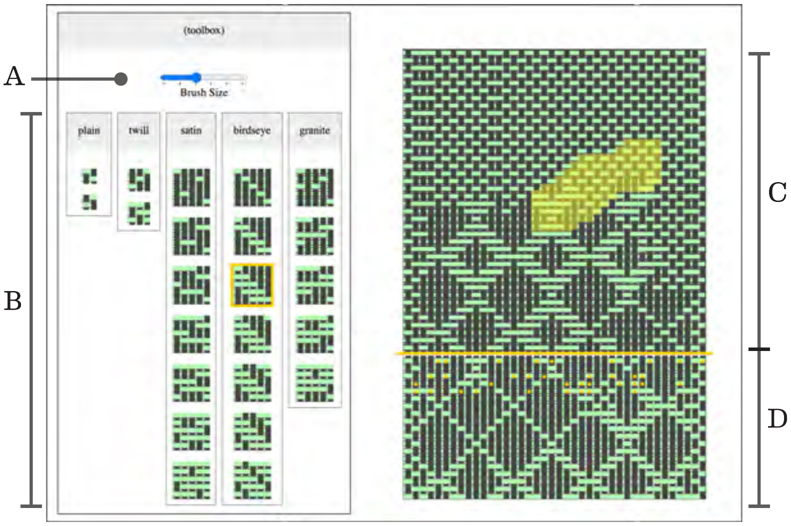


Figure 9.8 *The editor is a browser-based application which allows multiple users to simultaneously edit. A: "brush size" allows users to edit larger or smaller areas of the fabric. B: "brushes" are loaded from a pattern repository. C: the editable weaving area. A collaborator's edit is shown highlighted in yellow. D: Below the yellow line, the rows have been physically woven and can no longer be edited. Highlighted interlacements show where the weaver manually overrode the pattern.*

The editing interface includes "brushes" of different weaving patterns at different sizes; any collaborator can "paint" with these and a viable weaving pattern will result. The weaver can choose which brushes to make available to collaborators; for example, they could focus on just a series of tonally gradated satin patterns. During the weaving process, the weaver can manipulate the heddles directly to alter the pattern. The weaver can choose to capture these changes using the heddle-tracking system and send them to the server to share with their collaborators.

This system highlights how computational mediation can enable remote collaboration, as well as how simplified pattern design tools like "brush" systems can make expert tasks accessible.

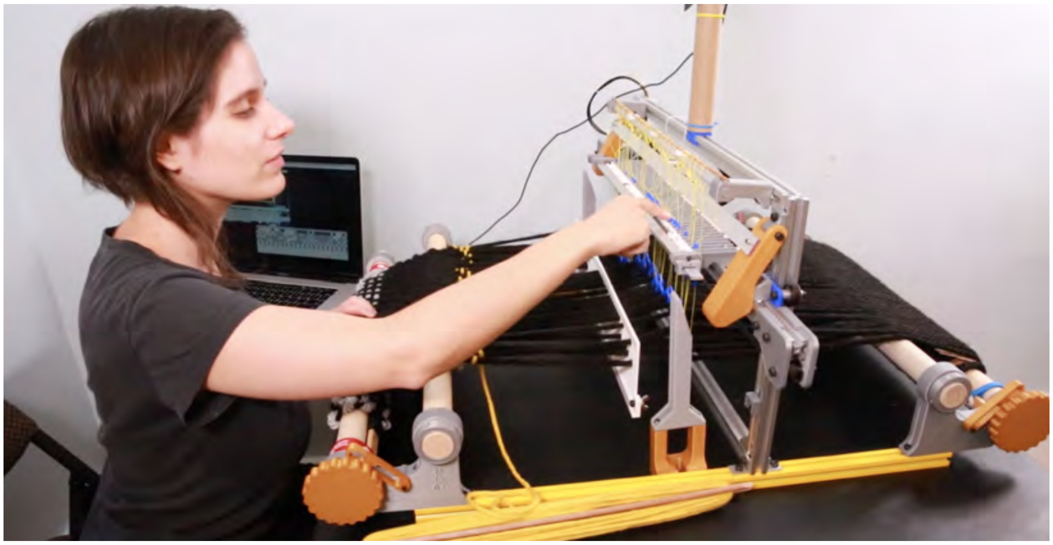


Figure 9.9 *The weaver overrides an area of weaving by manually changing some heddle positions.*

9.3.3. At-home Weaving

While our loom has many applications for use in a lab or makerspace, we consider its size and cost particularly appealing for home and daily use.

As an example of the potential of integrating computational weaving into a personal weaving practice, we show a system which encodes a daily weather report into the weaving pattern. The day's temperature relative to the previous day's is shown as the shading of the pattern—more warp-dominant patterns indicate cooler, whereas more weft-dominant patterns indicate warmer—and specific weather features are indicated with different styles of pattern, such as slanting twills for rain, dappled satins for clouds, and wavy broken twills for wind, Figure 9.10.

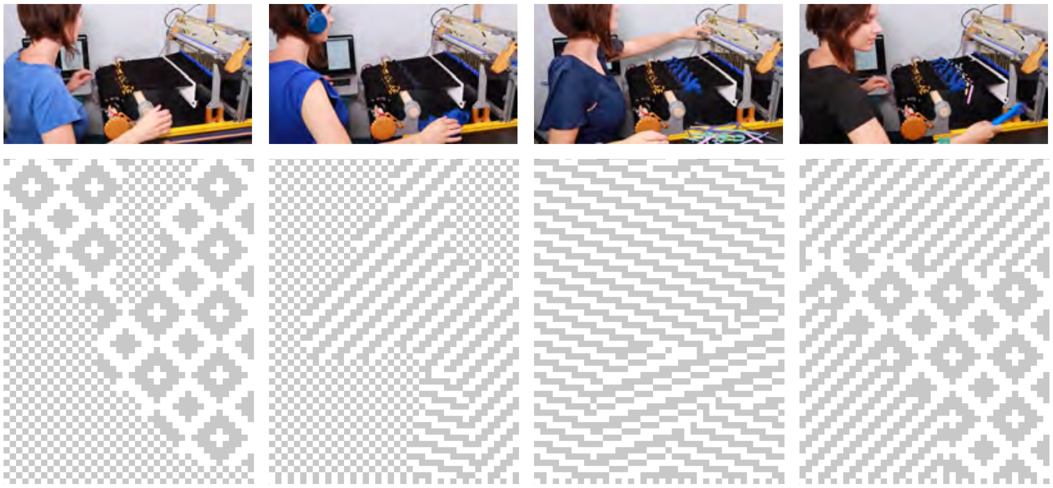


Figure 9.10 *Sample weaving patterns.*

Of course, the weaver retains control over weft yarn color choices, as well as how many rows to weave, so the resulting fabric incorporates both formal weather data as well as the weaver's own expectations and desire to spend time weaving on that day, Figure 9.4. This system demonstrates a way that the flexibility of computational Jacquard weaving can extend beyond weaver-directed changes, to incorporate sources which may surprise even the weaver, or spur their creative practice.

9.4. Limitations and Future Work

As a Jacquard loom, the serial selection design supports *any* sequence of warp and weft interlacements. However, some fabrics require other loom features. For example, the fixed heddle spacing presents a slight limitation in being decreasingly appropriate for higher numbers of woven layers (triple cloth, quadruple cloth, etc); however, these are typically woven on looms with other purpose-built features such as extra warp beams.

Additionally, the 10mm warp thread spacing (center to center) is large compared to most other looms. This is because of the overall priorities of low-precision construction and easy manipulability for hands-on hybrid weaving. Combined with the loom's small overall size, chosen for convenience and portability, this spacing results in only 40 warp threads. (Of course, the serial design means that adding more warps would only require more width, not any further electronic complexity.) However, many hand looms have a much smaller spacing, such as 2.5 mm for a "rug loom" (typically a four- or eight-set loom as described in Section 8.1). Future iterations might decrease the warp spacing in several ways: miniaturization with more-precise printing or other manufacturing approaches; additional heddle arrays above or behind, with constant mechanical complexity per frame; to an extent, tilting the heddle frames as in AVL's "dial a sett" system [7]. Each has tradeoffs but is potentially viable, and they might also be combined.

9.5. Summary

In this chapter, I have described an inexpensive personal weaving system which uses serial selection of a passive bistable heddle mechanism to avoid the electronic complexity of industrial Jacquard looms which actuate all heddles in parallel. This allows for a dramatic reduction in both acquisition and maintenance costs, as well as a much smaller "tabletop" form factor convenient for personal use. This portable, inexpensive loom makes Jacquard weaving accessible outside of industry and specialized textiles scholarship, with implications for e-textiles prototyping, fiber arts education, and personal craft.

The loom is explicitly designed to support and extend the activities of handweaving, which is a **soft context** quite different from the industrial production of cloth. As a ***soft technology***, the personal Jacquard loom flexibly supports non-standard materials and manipulations of the warp or weft, as well as improvisational changes to patterns and on-the-fly decisions. The ability to work with programmatically-designed double cloth structures, to record and potentially embellish improvisations, and to incorporate information from remote collaborators or data feeds demonstrate the **malleability of computation** as well as the **emergent expressivity of handweaving**.

10. Underdetermined Computational Handweaving

In the previous chapter, I documented a new loom design that enables personal Jacquard weaving, and I highlighted computational *hand*weaving as an opportunity for interactive creative intervention. I suggested several applications for computational patterning, while noting that patterning can be highly complex even for experienced weavers. In this chapter, I investigate computational patterning itself as a complex context, drawing on both computational and fiber arts practices. Specifically, I discuss how exploratory creative fabrication in a complex domain might be structured through the lens of *underdetermination*: the quality of a system which is set up to guide, but not fully determine, final outcomes.

In non-computational systems, highly automated production processes might be thought of as "certain"—any deviation from a set outcome could only arise from a flaw in the system—and opposed to more "risky" freeform manual processes ([310], as discussed in Section 7.2). Computational systems, however, can both highly embody precision and automation as well as highly allow for variability. Underdetermination can arise from the system itself: computational approaches can automate risk, magnify or elaborate upon it, or even inject it where it may not have previously existed [76, 81, 435]. Interactive systems expose the user to fluctuations in risk and resolution along the whole trajectory of making, mediating sources of risk and certainty according to generative logics.

A computer-controlled handloom is a hybrid fabrication tool: while the loom greatly speeds the process of weaving by precisely selecting threads for a pattern, a human weaver must manually throw the shuttle and beat the warp. The weaver is therefore present and involved for the entire production time, which may be a series of hours, depending on the size of the fabric; however, at present, weavers using computer-controlled systems determine the weaving pattern in advance, as the existing production systems do not support real-time interaction or design modifications to the pattern. This chapter describes technical intervention and a set of design propositions that reintroduce underdetermination into the computational system, channeling factors such as the weaver's intent and posture or external sources of disruption into the handweaving process.

Specifically, I describe a related but divergent family of three “sketches”: *Slit-Scan Self Portrait*, *Blobs*, and *Twitch Plays Loom*. *Slit-Scan* takes inspiration from durational, slit-scan photography. It translates a series of photographic captures during weaving time into the weave pattern itself. *Blobs* takes inspiration from other direct manipulation systems in fabrication by allowing a weaver to design at the time of weaving by using paper cutouts. As the weaver places cutouts on the loom, a camera reads their position and translates them into the weaving pattern. *Twitch Plays Loom* invites a group of remote spectator-users to specify thread-level design decisions during weaving time; these are enacted by the loom and weaver, and shared back to the remote users in real-time.

10.1. Design Process and Principles

My process consisted of designing and using three variations of real-time computational weaving systems in an effort to understand, from a first-person perspective, how the dual contexts of computation and weaving influenced real-time experiential processes in handweaving, and thus suggest how computational fabrication systems can best support personal creative practice. Inspired by the principles of procedural generation, I explored variable configurations of the balance between risk/certainty or underdetermination/constraint.

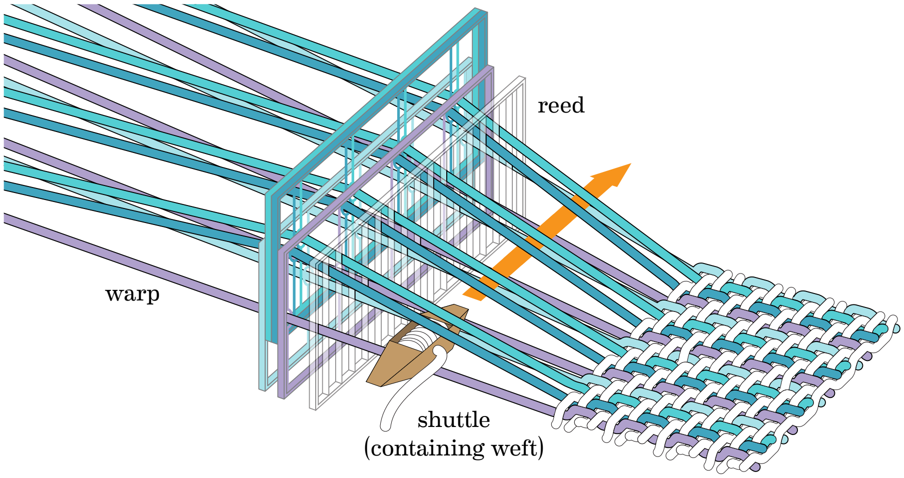


Figure 10.1 *A woven swatch, showing interlacement between warp (shown in four colors) and weft (white). In this fabric, four subsets of warp threads have been allocated, corresponding to four frames; in each row of the weaving, two frames were raised.*

10.1.1. The Design Space of Weaving

The process of weaving has two distinct phases. **First**, the weaver sets up the loom by measuring out the warp, tensioning it, and choosing which subset each warp thread belongs to. In a typical handloom, these warp threads are allocated to *heddle frames*, each of which can lift the threads belonging to it, Figure 10.1 (and previously described in Figure 8.2). **Then**, at weaving time, the weaver cycles through these actions: raising the warp subsets indicated in the pattern by pressing foot pedals (*treadling*) to raise the corresponding frames; *throwing* the shuttle containing weft thread through the space between the raised and lower threads (the *shed*); using the *reed* to *beat* the new row of thread against the existing fabric; possibly taking up the newly-formed fabric onto a collection beam; repeating the cycle.

Creating a fabric on such a system can be thought of as a procedural design task: weave-time changes in the treadling pattern can change the woven pattern only within the parameters of the warp threading. Traditional handweaving has a low number of heddle frames: often four or eight, and somewhat limited by the weaver's ability to select multiple simultaneous foot pedals. It's worth noting that non-obvious outcomes are possible even with fairly basic loom setups: a pixel of draft notation (Section 8.3) is not the same as an actual thread interlacement, and the interactions between warp and weft can be unintuitive, as in the double cloth shown in Figure 9.5. At higher numbers of shafts, it is increasingly difficult for even accomplished weavers to predict the outcome of a particular treadling sequence. This is particularly exemplified the algorithmic patterning approach called "network drafting" that blossomed in response to the availability of 18-24 shaft computational "dobby" looms to home handweavers beginning in the 1980's. The algorithm enforces structural viability of the woven output, but network-drafted designs are considered unpredictable and surprising. It is, of course, possible to pre-render the output of a network-drafted design, but in practice this style is designed by stylistic tendencies (with colorful names: "turtles," "snails," and "fleas") and treadling meta-rhythms [337].

When using a "fully Jacquard" loom such as the TC2 I used for this work (shown in the image at the beginning of this chapter), the patterning constraints, at least in terms of what can physically be done, are entirely removed. The weaver's work is simplified in two ways: because each warp thread is individually addressable, the weaver does not need to plan their work in advance and designate pre-determined subsets of warps; because the computerized system actuates the threads, the weaver presses just one foot pedal to advance to the next row. The pattern is not limited by the constraints of machine configuration or the weaver's ability to remember and execute a pattern.

Jacquard weaving therefore has a deceptively simple set of constraints:

- As an upper bound, the number of possible weaves can be enumerated by a binary choice of either "up" or "down" for each warp thread in a given row.
- However, to be a viable woven structure, there must be interlacement between warps and wefts. In the extreme case where all of the warp threads are selected to the same position, the weft does not interlace at all; more practically, for structural soundness, it is common to limit the distance without interlacement (the *float length*) of both the warp and the weft. (For example, the maximum float length of the fabric shown in Figure 10.1 is two.) The interlacement constraint makes it clear that each yarn crossing cannot be determined entirely in isolation.
- While the warp material is chosen when setting up the loom, the weft material can be chosen per row. Different weft yarns can result in very different appearances and material properties even with the same interlacement structure.

By decoupling mechanical and logistic constraints from weaving, it's simple to construct patterns that are highly chaotic, to the point of producing structures which are unviable, nonsensical, or simply uninteresting: a computational Jacquard system can just as easily emulate a basic plain weave as it can pattern the fabric based on cosmic background radiation or a bowl of Compton's oatmeal [314]. One way to state the task of system design for Jacquard weaving, then, is that goal is to *re*-introduce procedural constraints into the patterning process, recovering the underdetermined generative logics of handweaving but with the additional malleability of computation.

10.1.2. Methods

The methods I followed in this inquiry are were based on the precedents of Research Through Design [434], autobiographical design [69, 265], and reflective design [346]: each sketch served as a probe into my creative process; through the creation and use of these systems, I aimed to understand the hybrid soft context of computational handweaving, toward guiding other designers in designing interactive fabrication tools for handweaving and beyond. I engaged in reflective documentation in the form of pre- and post-experience journaling as well as semi-structured discussion with my co-authors of this work.

I chose to use an autobiographical approach to acknowledge that individual creative practice is inherently unique, especially in the case of work that actively encourages emergent outcomes and material experimentation. As such, I designed, used, and report on my systems as both designer and user of the systems described: When designing, I targeted myself as the user; when using, my experiences were necessarily inflected by my intimate knowledge of the systems' underlying structures. In a sense, using and documenting such systems is a collaboration between past and present selves.

Of course, this is a shift from how creativity support tools are typically studied in HCI. However, this time-based self-collaboration is rooted in the context of weaving as a procedural art, with patterning emerging from the interplay of decisions made at the time the loom is warped and those made at the time of weaving. Additionally, within the constraints of my access to the loom, choosing one user (and specifically, one whose schedule I had complete control over) was an opportunity to spend the time necessary to reflect on the nuances *between* systems, instead of shorter and necessarily more superficial exposures of multiple users to a single system. Following Höök [139], I prioritized understanding my own experiences—of the full-body performance of mechanically complex weaving [88]—as a necessary prerequisite to extending or considering how I might design for others with different preferences.

The complexity of these overlapping personas is reflected in the way I talk about my roles. When the work in this chapter was originally published ([14]), I used the "academic" third-person and first-person plural voices throughout. In keeping with the rest of this thesis document, I have largely switched to first-person singular where accurate. However, in discussing the outcomes of the two systems that I position as "performances" (*Slit-Scan* and *Twitch Plays Loom*) referring to "the weaver" in the third-person voice felt more appropriate, so I retained it.

10.2. Systems, Artifacts, and Findings

I designed, developed, and used the three systems in order to speculate on a personal computational handweaving practice, generate insights on underdetermination in the context of fabrication, and, on a practical level, produce technical infrastructure for future experiments in real-time control over Jacquard handweaving.

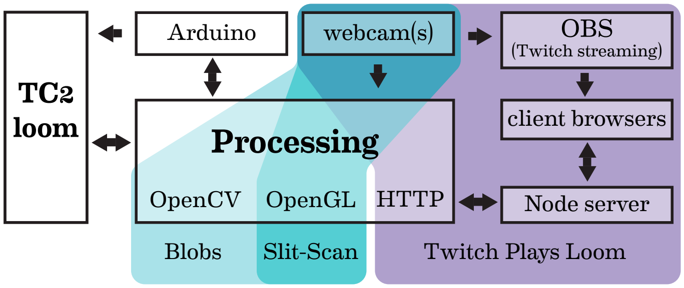


Figure 10.2 *Technical diagram of the TC2/Processing system.*

10.3. Technical Implementation

I developed the work in this chapter as a visitor to the Unstable Design Lab of the ATLAS Institute at University of Colorado Boulder. I reverse-engineered the control protocol for my gracious hosts' Digital Weaving Norway TC2, a popular computerized Jacquard handloom. The TC2's assumed workflow involves uploading a bitmap image representing the warp thread positions for each row of a complete fabric. The loom requests the next row of this static data from the control software each time the weaver presses a foot pedal. Data is sent from the TC2's control software to the loom itself using TCP over WiFi. I used Wireshark [412] to sniff this data; by comparing the transmitted data to a known sequence of selected threads, I was able to isolate the commands to establish a connection to the loom, control its air compressor, receive requests for row data, and send row data in response. I additionally developed an Arduino-powered replacement for the foot pedal, allowing us to issue "next row" requests programmatically.

I encapsulated my row-by-row protocol as a library for Processing, enabling it to be used with a variety of other input and output modalities. These modalities included live video processing with OpenCV, text-to-speech, live many-to-many internet-enabled communication, and custom additional hardware buttons, Figure 10.2.

I quickly iterated many interaction "sketches" based on these capabilities, and then chose three main sketches to develop more fully. My selected sketches were unified by their use of a video feed modality (albeit in different roles: as a literal image, as a composition input, and as an entertainment medium) but otherwise mutually differing in tone, extent of pre-determination, additional role for the weaver, and similarity to existing works; e.g. "Blobs" was inspired by existing on-machine fabrication [251], whereas "Twitch Plays Loom" explores territory that is less familiar within fabrication.

While all handweaving is in one sense a performance—the enacting of a repetitive task to call something into being—the selected sketches additionally provide distinct additional roles for the weaver to engage in while enacting the weaving. All three sketches were then refined to be suitable for a single weaver and 2-3 hours of weaving time, which is a maximum session length for the loom hardware as well as for my own comfort.

10.4. "Blobs": Designing With Paper Cutouts

This sketch was an on-machine design interface inspired by tangible remixing interfaces [91] and on-device specification [251, 256, 297]. The designer could arrange scraps of brightly-colored paper directly on the unwoven warp of the loom (shown in the image at the beginning of this chapter), and take a "snapshot" to generate a weaving pattern at 1:1 scale with the cutouts rendered in a palette of weft-dominant (light-colored) and warp-dominant (dark-colored) diamond twill weaves. In addition to enacting the weaving, the weaver could therefore tinker and disrupt, allowing composition, remixing, and non-linear sampling to intervene into the rhythm of the weaving, Figure 10.3.

To implement this sketch, I first processed the camera feed with OpenCV: I used color detection to isolate the blobs, then image rectification to map the camera input onto the real size and shape of the woven fabric. I then used a custom OpenGL shader to assign weave structures to areas of this cleaned and rectified image, essentially implementing a real-time image processing version of the technique described in *The Woven Pixel* [338]. This sketch was therefore constrained to produce viable weave structures. I chose a family of "birdseye twill" weaving structures for their unique appearance and wide range of tonal values, Figure 10.4.

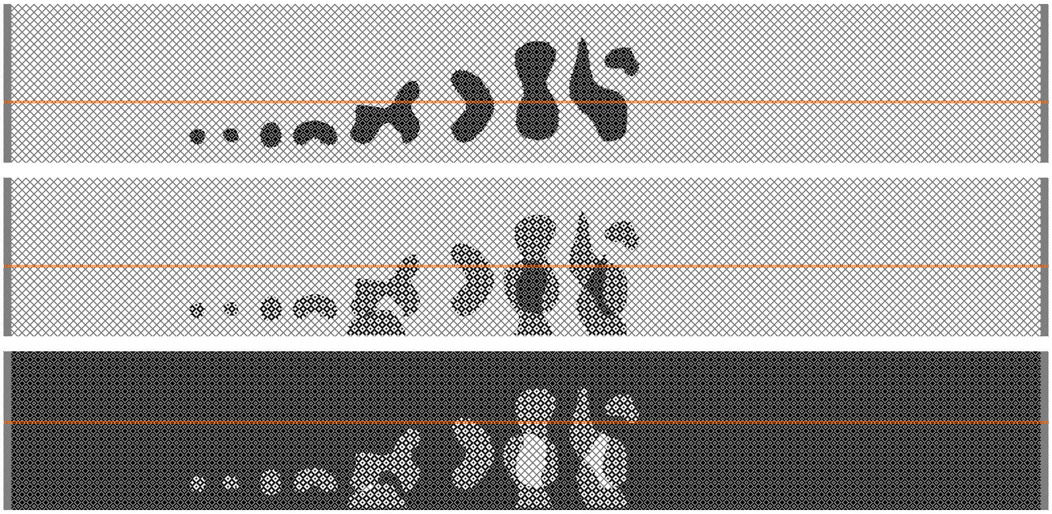


Figure 10.3 *A sequence of snapshots generated with the second version of the "Blobs" interface. The orange line indicates the row of weaving at the time of the snapshot. Top: a new snapshot in high-contrast tones. Middle: a second arrangement of blobs is overlaid onto the first at 50% opacity, resulting in mid-tones. Bottom: the composition is color-inverted.*

I ran two versions of this interface. In the first, the only input to the system is the placement of the paper cut-outs and the choice of when to take a new snapshot. At the extreme end, the designer might choose to sample every row, leading to slit-scan-like effects. I wove this first version for forty-five minutes and discovered that, in practice, there was little motivation to take another snapshot before the first was completely woven. The pattern produced was therefore a faithful reproduction of the cutouts, but it left very little space for designerly interaction in mid-weave. Additionally, I found that I was concentrating on the on-screen preview of the weave pattern, instead of on the fabric itself.

In the second version, three capabilities were added: 1) the designer could composite a new cutout snapshot with the previous one. The new one would be additively blended at 50% opacity with the existing snapshot, allowing mid-tones to be introduced. 2) The designer could choose to invert the dark and light tones. 3) The designer could skip to a different line of the composition to re-mix the line order. Each of these decisions could be made at any point during weaving time.

These additional capabilities gave me more opportunities for manipulation during the weaving process. Additionally, “jumping around” in the pattern broke the direct correspondence between the on-screen representation and the resulting weave, allowing me to focus more directly on the woven output. The capabilities also provided intervention possibilities on several time scales: the “invert” capability produced a relatively immediate effect, whereas the “composite” capability caused changes that took a greater number of rows to reveal.

I wove the second version for two and a half hours and produced a composition with seven snapshots introduced within the process, Figure 10.4. While this was the same amount of time spent weaving the Slit-Scan sketch (described in the next section), I perceived the Blobs interface as faster and less exhausting.

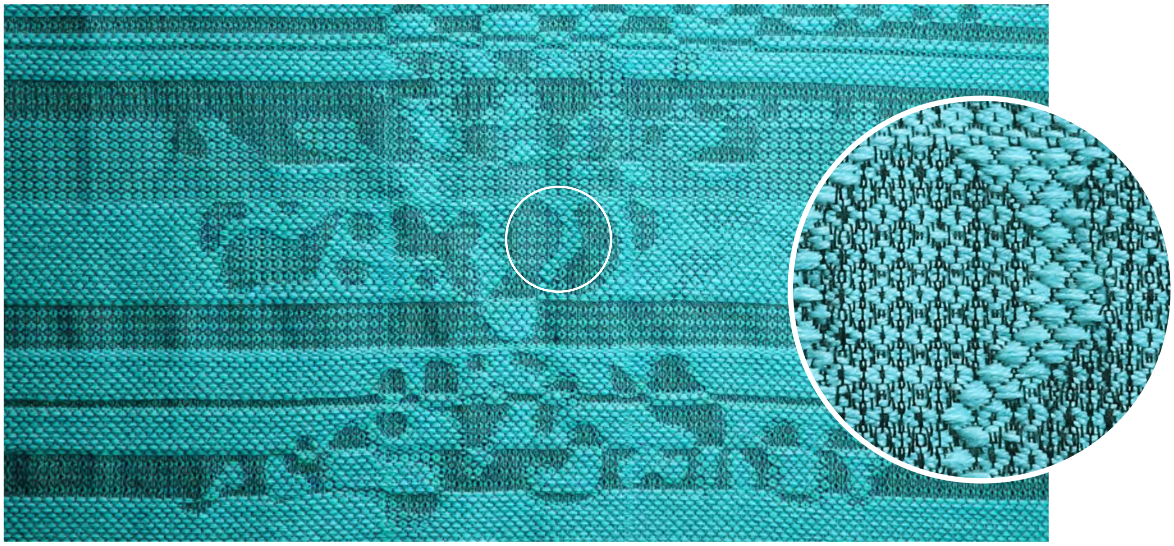


Figure 10.4 *The composition generated with the second version of the “Blobs” interface. The detail shows how a family of related “birdseye twill” patterns provides tonal variation.*

10.5. Slit-Scan Self-Portrait

A second sketch, Figure 10.5, was influenced by two conceptual threads: glitch aesthetics in textiles [231] and objects which visualize their own creation process [209]. "Slit-scanning" is a photographic technique in which a scene is sampled through a narrow, moving window over time [200]. The Slit-Scan Self-Portrait sketch positions its user as both weaver and subject, requiring the two interleaved tasks of weaving and of posing for the camera. The weaver strove to maintain a similar pose for the camera samples, Figure 10.6, and later noted that this task felt sportlike, like a gymnastics task judged on both emotional display and technical precision. Prior to the actual weaving session, the weaver altered their appearance with high-contrast makeup with the hopes of enhancing the quality of the output image.

I implemented the sketch as a window corresponding to the progress of woven production, with the image data converted to a tonally dithered "shaded satin" structure via the same OpenGL shader technique as in the "Blobs" sketch, Figure 10.7. The webcam was pointed at the weaver and I planned to sample the webcam image from the bottom of the frame to the top, corresponding to the direction of weaving from the weaver's point of view.

Figure 10.5 *A screenshot of the view the weaver could see during the weaving process, with a small live video feed showing the portrait crop area and a preview of the woven structure in progress. The aspect ratio of the image is distorted to account for the non-square "pixels" of woven interlacement, which vary according to relative warp/weft thickness and other factors.*

Because the image is of the weaver, it is guaranteed to be disrupted as the weaver must move around in the very process of weaving. Slit-scan data can become incoherent depending on the design constraints: the sampling rate and the height of the sample. A very narrow sample (such as just one pixel row tall) sampling a chaotic source very slowly may cease entirely to look like its input. Because the warp is very wide, I planned to composite together four different sampling rates side-by-side. The most frequent sampling rate was every eight rows and the least frequent was every thirty-two. The "not-yet-decided" portion of each panel was shown as live (satin-dithered) video in each panel.

Figure 10.6 *Representative examples of face images captured during sampling.*

However, during the weaving session it was discovered that an error in the code meant that the actual woven lines were sampled from the top of the image to the bottom, thereby resulting in an image that was functionally sampled every row for the top half of the image. I report on this because it resulted in several arcs of expectation and surprise during the weaving process: first, when the bug was undiagnosed, it seemed that the woven results had no relation to the image input, that predicting the outcome would be impossible, and that the weaver would simply have to surrender expectations; second, when the bug was discovered, a moment of relief—the results were indeed coherent, just inverted—was followed quickly by disappointment, because it meant that the pattern was fully determined at that point and there was no reason to continue to pose for the camera. However, the net result was, in fact, delight: this error was in a sense a genuine glitch within an engineered glitch-like system: an accidental swap of axis direction in a system which deliberately transcoded time as the y-axis. The trajectory of this experience shifted as uncertainties and stakes came more or less in focus. The full weaving experience lasted two and a half hours.

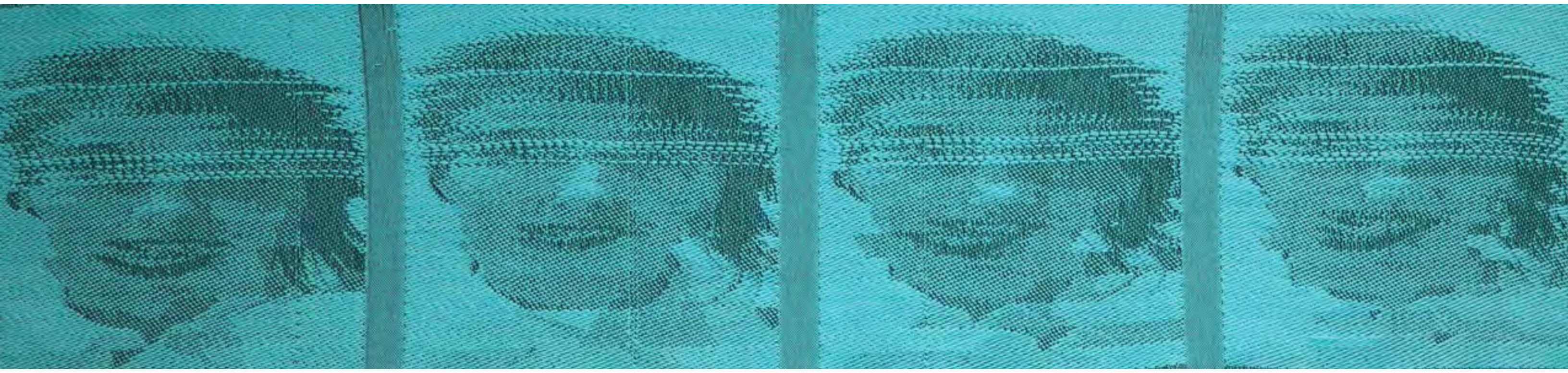
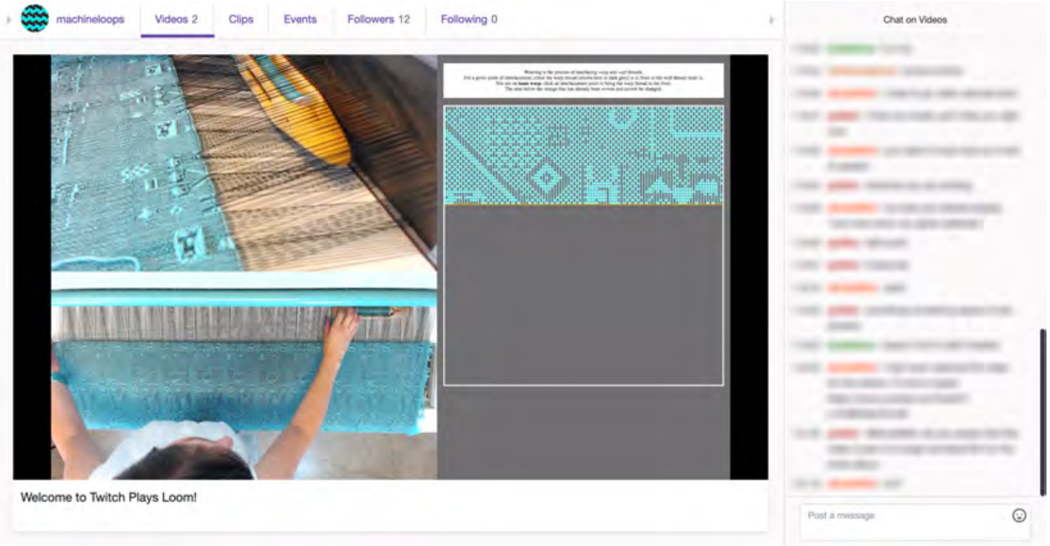


Figure 10.7 *The final woven fabric.*

10.6. "Twitch Plays Loom": Anonymous Networked Editing

The third sketch, formally named "Twitch Plays Loom (An Actual Loom, not the 1990 Graphical Adventure Game)" was influenced by "playful fabrication" [13, 368] and spectator-based interactions [345]. This sketch opens the editing of the weaving draft to internet spectators, who may additionally observe the weaving process streamed as video on the popular live-streaming site Twitch, Figure 10.8.



Welcome to Twitch Plays Loom!

Figure 10.8 *A screen-capture of the Twitch stream, showing composited overhead and close-up video sources, a feed of the live draft, and viewer chat.*

The interface was implemented as a client-side JavaScript browser application communicating over websockets to a Node server. A local Processing sketch requested interlacement data from the server, formatted the repeated layout across the width of the warp, and passed the data on to the loom. Spectators could view the weaving in real-time through two cameras; one provided an overhead view of the full warp, and the other provided a close-up. In the browser application, Figure 10.9, the spectators could directly edit a limited area of one hundred warps wide and one hundred wefts tall.

Weaving is the process of interlacing *warp* and *weft* threads.
For a given point of interlacement, either the warp thread (shown here in dark grey) is in front or the weft thread (teal) is.
You are on **team warp**: click an interlacement point to bring the warp thread to the front.
The area below the orange line has already been woven and cannot be changed.

Figure 10.9 *The draft-editing interface presented to the remote users*

This limited width was repeated across the full weaving width of the loom, Figure 10.10. After one hundred wefts were woven, a new plain weave draft was generated for the spectators to edit. This process was repeated three times for about two hours of weaving. In the the fourth draft, the spectators began seeking ways to circumvent the repetition of their task by directly scripting their interlacement swaps in their browsers' JavaScript console. (One managed to re-boot the server, so the final number of woven rows was not an integral multiple of one hundred.) Thus some obviously computational aesthetics, including random noise and a Sierpinksi triangle, emerged in the last part of the session.

The asymmetries of streaming were evident: while the Twitch chat stream was lively and the spectators found the experience "fun" and even "calming," as the weaver I found the experience awkward and alienating. Online streaming is subject to the pacing of network lag, which can be roughly twenty seconds [106], and the weaver could only catch snippets of the chat while close enough to the computer screen. As a result, there was a clear performer/audience divide that made this sketch feel even more "like a performance" than "Slit-Scan" did.

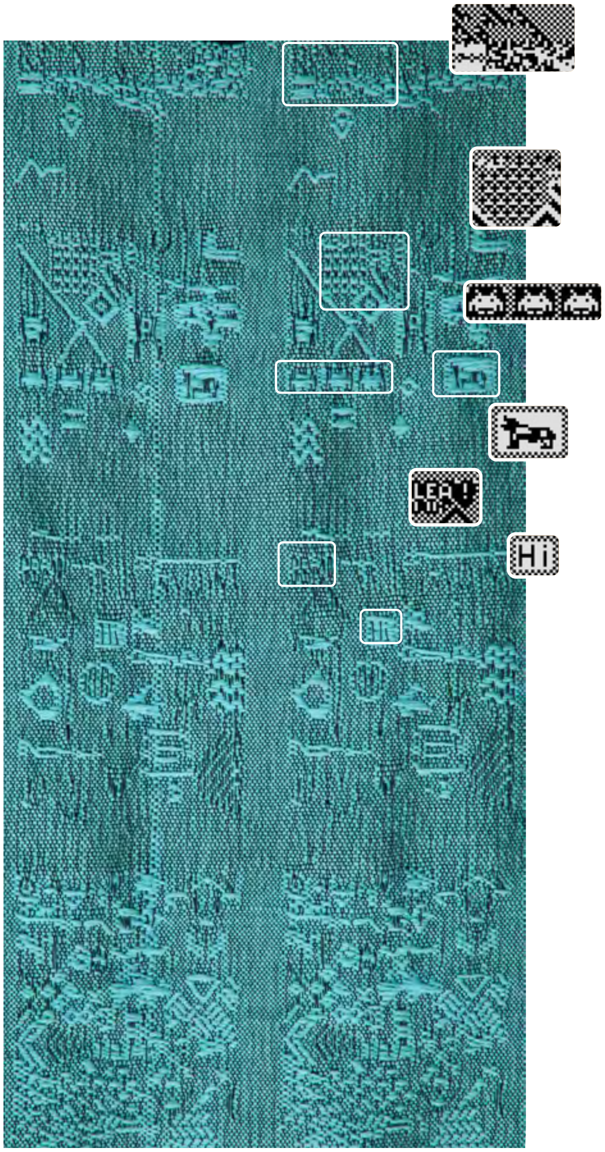


Figure 10.10 *The resulting fabric. Two of the eleven repetitions across the width of the fabric are shown.*

The spectators were not specifically weavers, and indeed several commented with surprise on aspects of the weaving process: noting how hands-on the process was, finding difficulty in the task of creating viable weave structures, and marveling that their seemingly inconsequential clicks were being physically manifested. They were reluctant to overwrite others' contributions, suggesting a strong awareness of their fellows' presence and recognizing their labor.

10.7. Discussion: Procedural Generation Under Risk

When focusing on the design of underdetermined systems, there are almost infinite amounts of variation possible. In my studies, I show that a myriad of outcomes can exist according to subtle variations within a single tool, or even within the act of simply surrendering to a tool that might appear to be not working. I believe that these parameters are ultimately ones that will be shaped less by the available tools and more by the desire an individual has for their own practice.

The procedural generation lens clarifies a possible role of output in a generative system: as one way to understand and appreciate the underlying system, but neither as its entire goal nor as a by-product. This view contrasts with fabrication research, which has long prioritized *either* a singular output *or*, in more-recent work under alternative value systems, an experience of making [253, 382, 409]. Instead, material and compositional concerns (risk and viability, material scales) can be aligned with procedural generation ideas of *form*, and experiential concerns (agency and role, suspense and temporal scales) with those of *locus*, "a balance between the Structure of the generator's processes, the Locus Gestalt of the generator's output, and the Surface of the immediate experience of individual generated artifacts" [169]. These concerns can then be balanced in mutually supportive ways within the generative logic of a system.

For example, slower temporalities of making become not just a reaction to production-oriented values, but a system quality chosen to complement particular material scales: by literalizing the weaving speed through sampled video capture, the material output celebrates the experience's specific temporality. Internet spectators can act as a source of disruptive data, but that data is necessarily filtered through a cultural milieu which encourages particular kinds of social actions. When applied to other computational fabrication domains, the nature of these interventions will inevitably vary to suit different material forms and machine loci. For example, the role of heat in thermoplastic 3D printing is both highly technical and potentially a source of emotional resonance.

I offer the following themes to help organize approaches to underdetermined systems in digital fabrication. I discuss each theme within the context of computational handweaving, illustrate the factors with examples from my sketches, and offer broader implications for both fabrication research and procedural generation work.

10.7.1. Immediacy and Gestalt: Frequencies in Time and Space

Gestalt aesthetics are particularly suited to procedural generation—consider a Twitter bot whose animating principles are understood best when its output is viewed in aggregate. Repetition with variation can clarify essential vs inessential qualities: which elements are integral to the underlying logic, and which are embellishments, echoes, or stochastic variation.

In weaving, the output could be considered to be an entire fabric, or a section of weaving following a particular decision by the weaver, or one weft pick, or even just a single interlacement. Traditional frame weaving requires a generative logic to be determined at loom set-up time, which then applies to all the fabric woven with the entire warp.

Because the fabric is built up row by row, the pacing at which uncertainties are introduced and resolved is both a spatial one (in the y-axis of the fabric) and a temporal one (over weaving time). The larger the system output, the more data it can contain but the longer it will take the weaver to encounter it. Additionally, limited resources (e.g. yarn) may impose a limit on total output size. Because an "output" can also be seen as a horizon of results on which the system will make no more decisions, very large output sample sizes might become difficult to distinguish from predetermined patterns. I observed in the differences between the two versions of the Blobs interface that the "interactive feel" of the system (and consequently my desire to continue engaging with the system) relied on the possibility of making meaningful changes "in real-time," which, at least for me on that particular day, meant no more than a few minutes per decision-chunk.

The gestalt of a temporal interaction can also be understood through the narrative concept of "suspense": anticipation, or a sequence of uncertainty and resolution about something with emotional stakes. Higher sampling frequency increases risk through compounding the possible uncertainty but can potentially decrease it by lowering the stakes in terms of material or time costs per decision. Additionally, very high sampling rates can lead to effective incoherence, and thus a breakdown in the emotional stakes of the process.

My three realized sketches primarily focus on steady, real-time pacings—that is, disruptions arise and are resolved during a continuous session of weaving at the weaver's natural speed. However, other mappings are possible: one of my imagined sketches positioned the act of weaving as a daily ritual of care for a virtual entity: "loom as virtual pet." The process of weaving might then extend over months, with relatively little fabric generated.

Broader Implications: Within the fabrication landscape, I see this factor as pointing toward the necessity of tuning generative systems to specific fabrication contexts. This close link between time and material scale is particularly pronounced in weaving; many digital fabrication processes do share a linear progression (e.g. 3D printing typically uses a layer-based approach) but this is not universal. Subtractive processes like milling may progress from rougher to finer detail, or 4D printing includes transitions between shape states after fabrication is otherwise complete [380]. Extending ideas of generative system design to other physical media may prompt re-consideration of the time scales involved (e.g. the short time scales of glass-blowing vs the long scales of gardening), as well as how to support rhythms of production in less-linear media.

Within procedural generation research, the timescale of operator engagement in these woven systems is unusual for generative systems—even bots whose output unfurls over weeks or years do not typically require ongoing labor from their observers. "[S]ustained, deep engagement with a single, gradually evolving generated artifact" has been proposed as a partial solution to the problem of player desire for endless fresh content in games [186]; labor and material risk underscore these. I see craft attitudes to difficulty and embodied value as a possible antidote to novelty churn in procedural generation.

10.7.2. Roles and Sources of Disruption

Most procedural generation systems are either fully independent of user input (creator sets them in action), or are systems to elaborate upon or "complete" a user input. My weaving systems inherently require weaver action to enact the woven result; the weaver therefore has at least one role within the system, and my sketches all introduce others. A "role" here is a manner in which an element of the system holds power, and it can be conceptually underpinned by analogy to roles in other systems (e.g. the role of "subject" in "Slit-Scan," or "host" in "Twitch Plays Loom"). Roles can entail responsibilities and priorities, and can delimit acceptable inputs to the system.

In underdetermined systems, an important role is that of disruption. Sources of disruption can be within the structure itself, as in a glitch system, within the weaver as in my Slit-Scan and Blobs sketches, or from the environment as in my Twitch example or data visualization weavings [406]. Sources can be poetic or meaningful, or deliberately in tension with the production process (as in the slit-scan example, in which the weaving process itself is guaranteed to disrupt the image input).

The dynamic balance of these scales and sources determines what or who "matters" to the experience and output. Twitch Plays Loom is an example where both the disruption and the stakes, and therefore also the locus of importance, come from the live spectators; the weaver acts mostly as a conduit for these, enacting the weaving for the enjoyment of the audience. The "float length" constraint (described in Subsection 10.1.1) affects the viability of the woven structure. While I used shaded satin structures to impose fairly tight float length constraints in the Slit-Scan and Blobs sketches, I did not enforce any weavability constraints in the Twitch sketch. Insteand, as the weaver, I periodically reminded the spectators to consider floats (and occasionally the spectators reminded each other) but did not overrule any potential problems. In addition to taking the focus off viability, relaxing this constraint gave the spectators rein to be mischievous or even subversive.

Broader Implications: Within fabrication, I often see discussion on the distribution of power between users and systems: machine systems as co-creators, as familiars, as apprentices [19, 168, 220]. I envision opportunities to examine not just the relative extents and positions of power, but the manner and social templates of how it is deployed in ways that go beyond humanlike characters: systems as parties, as fortune-tellers, as camping shelter.

In procedural generation, intriguingly, I primarily see extended user roles specifically within analog systems [356]. However, many digital systems have the implicit user role of interpreting the output; consider a Twitter bot that generates short murder mysteries [59], or a generator of instructional artworks [58]. Integrating complex roles alongside a generative process may deepen a generative experience, or extend the possibility of circular, iterative, or reflective interactions.

10.7.3. Order, Disruption, and Effective Complexity

While sources of disruption give uniqueness and meaning to a system, they must be balanced against order to be legible. Karth notes that "perhaps the central tension is between the randomness that generators use for aleatoric novelty and their need for ordered structure to give that novelty the context for it to have any meaning" and cites Galanter [103] to point out that "effective complexity recognizes that highly disordered systems are nevertheless conceptually simple."

Weave structure viability is one form of order in a generative weaving system, along with factors like semantic content and visual organization (e.g. symmetry) in the output and regular pacing by the weaver.

Two possible mechanisms for ordering disruption are repetition and multiplicity. In the “Twitch” sketch, the editable canvas available to the spectators is only one hundred interlacements wide, allowing it to be repeated eleven times across the width of the fabric. In addition to focusing the participants’ editing efforts into a less overwhelming space, this copy/paste repetition in the final artifact points to the fabric’s computer-mediated origins, despite its chaotic aspects within a given repeat. In contrast, the “Slit-Scan” sketch also generated repeated frames within the fabric, but the different sampling rates generated a multiplicity of specific outcomes from a unified underlying input (the video feed).

Broader Implications: Under material risk, factors like repetition or multiplicity must be balanced against cost. Repetition may even be seen as contrary to the ideals of digital fabrication. However, viewing these tactics as part of a system of meaning can surface and support the system’s underdetermination while also celebrating an artifact’s computational origin.

For procedural generation, physical viability can be a rich source of order. Generators that produce bio-inspired imagery may be considered to be indirectly constrained by physical viability—e.g. a “leaf” generator seeks to replicate or expand upon forms that were initially produced under viability constraints. Material craft processes have embedded vocabularies that may serve as inspiration or goal structures for computational processes.

10.8. Limitations and Future Work

I sampled the space of real-time computational handweaving systems in three places, demonstrating effects in each of my main themes. I offer suggestions for how to tweak or slide these examples for individual experience, but such a sampling is by its nature specific and personal.

10.8.1. Other Computational Modalities

As mentioned, I developed several other technical implementations of input modalities before refining my sketches. Each could interact with my design factors in a multiplicity of ways. My modified foot pedal could allow us to override or shift the pacing of each row request; it could be placed at great distance from the loom, shifting the weaver’s role to athlete; it could use voice recognition to behave petulantly, refusing to progress to the next row unless soothed with song. Gestural input could be used to disrupt or smooth a pattern, or to suggest other roles for the loom itself: as a musical instrument, or as a garden bed. I view these modalities as essentially compatible with my focus on their effects on pacing frequencies and as sources of disruption or order, but I trust that their specific outcomes must be discovered through experimentation.

10.8.2. Manifesting Community Around Digital Fabrication

An individual weaver is only one of the possible participants that could be involved in a weaving process. The social aspects of analog fabrication are well-documented and have found opportunities in online/networked space [36, 128, 189, 280]. There are also online communities for digital fabrication enthusiasts [242]. My "Twitch Plays Loom" sketch integrates a social aspect; however, while it established a community space for the spectators, it was less successsful in holding that space for the weaver. The complexities of human social interaction could offer a rich source of variability and meaning in digital fabrication contexts.

10.8.3. Tuning for Production

While designing my systems, I deliberately set my scope outside of purely productive outputs. Indeed, many analog fabrication techniques are engaged in as enjoyable pursuits or aesthetic experiences in their own right, and the crafting community even considers some versions of productive challenge as semiformal games [369]. However, I observe that changing some parameters to individual preferences can be enough to tilt a system from "playful" to "serious"—e.g. by using a double-cloth weave structure instead of satin, the "Blobs" interface could be used as a tool for manipulating functional e-textile layouts and integrating component pockets at a 1:1 scale; the livestreaming "Twitch" interface could be used with an audience of expert weavers to harness their expertise as a learning tool for the weaver.

10.9. Summary

I presented computational handweaving as a site for exploring the experience of real-time, interactive, and underdetermined fabrication. I developed three novel generative systems for interacting with computer-controlled weaving equipment and used them within my own practice to reflect on the felt and embodied experiences they brought forth. As a way to think through the connections between system design and experience, I identified temporal and material factors that shape interactive fabrication systems. I suggest that these factors help readers understand that playful, exploratory and otherwise reflective engagements in real-time fabrication might need to be "tuned" to invididual users/tasks within this parameter space.

As ***soft technologies***, underdetermined weaving systems are adaptably structured, and are necessarily and joyously shaped by their contexts. I aim for the work in this chapter to inspire inquiry beyond the particular site of weaving to consider how one might traverse, creatively ideate, and play simultaneously within the manipulable material world and within the fluid space of computational possibilities.

III.

Liminal Workflows

11. Background and Related Work: Soft Abstractions and Tools

Workflows are how we bridge between physical operations and contextual goals. One popular form of workflow is conceptualized as a pipeline: a linear sequence of processing steps from an idea through intermediate representations to a final result. Traditional digital fabrication pipelines have their roots in industrial hierarchical distinctions between engineers, draftspeople, and machinists, separating Computer Aided Design (CAD) tools for producing dimensional geometric representations of forms from Computer Aided Manufacturing (CAM) tools for producing fabrication machine instructions such as toolpaths [215]. Loops in this overall pipeline may well be expected, as when digital fabrication is used for "rapid prototyping," but the processing sequence remains more or less fixed.

However, workflows in practice, as used by practitioners from artists to researchers, are much more fluid and hybrid [201, 388]. Soft workflows are *liminal*, meaning transitional or indeterminate, both in their role as bridges and in the possibility for the workflows themselves to be underdetermined and adaptable. Tools may present not just a single result, but a space of possible outcomes; processing might be less of a direct translation between pipeline stages, and more of a transmutation. In computational fabrication, workflows can benefit from both the "soft" of "software"—the parts of the machine that can get remapped—as well as fuzzy inputs from and outputs to the physical world.

11.1. Artifact and Experience

In designing workflows, there is a tension between the metaphorical "space" of possible outcomes and the physical space and time in which creation occurs. The creator's *experience* may be formed over varying amounts of time and feature glimpses of diverse parts of a possiblity space, but ultimately a single point is resolved as a physical enduring *artifact*.

Colloquially, artifact and experience may be treated as fundamentally at odds, with artifact "quality" possibly sacrificed in favor of a more satisfying experience. For example, a casual tie-dyeing process might be undertaken for the sheer delight of seeing what happens; the resulting t-shirt might be more of a souvenir than a cherished item in itself. (Some notable extremes of this are "destruction" experiences [81] which are entirely disinterested in the resulting artifact.) However, in a computational system, the lines between experience and artifact are not always clear. In any interactive system, the "outcome" could entirely be the experiential memories of the user [185]; in a procedural fabrication system, an "outcome" could be a single artifact, or it could be a series of artifacts understood to be sampling a possibility space in some way [169] and therefore resolving in a more experiential way. The computational abilities to repeat and "undo" (or, in the case of fabrication, "fork and incrementally modify" [251]) may mean that a particular output is *an* artifact but not *the* artifact.

11.2. Interactive Fabrication

In a strict CAD/CAM workflow, the fabrication process itself is almost entirely non-experiential, often with long periods of time that are, ideally, uneventful. Interactive fabrication is an area within fabrication research that focuses on enabling a design process to occur *while* an artifact is being constructed, shifting the role of the fabrication device from being the endpoint of a pipeline to being the site of design and material exploration.

In "Interacting with Personal Fabrication Devices" [251], Mueller defines interactive fabrication as having "four main characteristics: 1. the physical environment is the workspace, not a digital editor; 2. users work hands-on on the physical workpiece through physical tools as known from traditional crafting; 3. each physical action results in immediate physical change, which can also be reversed; 4. in contrast to traditional crafting, users receive support from a computer system that helps to achieve precision."

While I diverge from this approach slightly in de-emphasizing precision as the value added by computation, I find hands-on, real-time intervention with a fabrication machine pleasingly aligned with both the importance of material as a locus of creativity and Pye's ideas about interaction during the act of physical creation as a fundamentally unique paradigm (as described in Section 7.2). I will discuss the implications of Muellerian and similar interactive fabrication systems along two axes: *real-space* (corresponding to Mueller's characteristics #1 and #2) and *real-time* (corresponding to her characteristic #3).

11.2.1. Real Space

One key component of Mueller's style of interactive fabrication is the idea of working directly on "the workspace": the material, physically located on the fabrication machine. For example, Constructable [254] allows a user to "draw" directly with a laser pointer onto the cutting area of a laser cutter with any of several tools to indicate a range of material effects, from cutting and scoring to producing complex gears and live hinges.

Such systems can integrate data about real-world objects placed on top of the in-progress model, allowing direct, measurement-free communication of constraints. Sketch&Stitch [124] allows a user to designate embroidery paths for e-textile circuits by placing component modules directly on the embroidery machine's sewable area. (I used a similar in several of the weaving systems in Part II.) Similar affordances have begun to be designed into commercially-available fabrication equipment, such as the AutoPilot software for Innova Longarm quilting machines [343] which can use the operator-positionable sewing head as a cursor in the task layout mode, and the camera view mode of a Glowforge tabletop laser cutter [107], which overlays a rectified view of the real physical material onto the design canvas.

Understanding real-world dimensions is a key problem in fabrication — arguably, seeing something at the right size is one of the primary drivers of computational fabrication as a prototyping tool for designers, and users often struggle even with linear measurements [175]. By handling these aspects of fabrication implicitly, on-machine fabrication greatly reduces the friction of incorporating existing physical objects, either as inspiration [90] or as components of the final object [418]. This is a particular benefit when the user is working with irregular materials, such as in Larsson et al's system for creating structures out of naturally-shaped tree branches [195, 404]. When a fabrication machine is small and/or maneuverable, such as a Shaper Origin portable CNC router [350] or a handheld filament extruder [327], "the workspace" can be anywhere the machine can be deployed, broadening "on-machine design" to include "in-situ design" and making it available for complex contexts.

On-machine interaction might also minimize or entirely remove screen-based representation, which particularly supports context-specific fabrication paradigms that do not translate as well to a screen. Mueller's FormFab system supports shaping plastic sheets with heat and air pressure (as in glassblowing or "billow" thermoforming); its gestural input system is well matched to the continuous physics-influenced forms it can create.

11.2.2. Augmentation for Interactivity

As a variant of hybrid fabrication, I am especially interested in *adding* computational capabilities to a not-otherwise-computerized machine. There exists a wealth of existing fabrication technologies and communities of practice that are currently seen as outside the scope of computational fabrication research. However, systems like "Drill Sergeant" [341] and "Adroid" [378] have shown how computational systems can augment a manual fabrication task with real-time guidance, and augmented reality research has long shown how helpful information can be overlaid onto physical surfaces [407], boosting users' ability to navigate complex, domain-specific, and critical tasks [355]. Reflecting on the ways that interactive fabrication research has meaningfully extended fabrication machines past the original bounds of the technologies to include the operator as "part of" the machine, I do not consider it a stretch to include computationally-*augmented* fabrication as an additional and meaningful form of computational fabrication.

11.2.3. Real Time

The other component of "interactive fabrication" is its situation during the process of fabrication: changes can be chosen and implemented while an object is being made [409].

An arguable nascent form of interactive fabrication is the act of slightly intervening in the fabrication process—for example, briefly pausing a 3D printer to insert an external material into the print [319], or manually changing a plotter's pen color [388]. Such turn-taking between fabrication equipment and user can be used to accomplish tasks that neither could accomplish alone [223]. Interleaving fabrication and design processes might also be thought of as a way of extending historical ideas of physical icons or widgets as *input* [117] by producing physical *output* as well.

From another angle, beginning with the idea of computational fabrication as a "prototyping" technology, concepts like "bidirectional fabrication" [405] push toward shorter and more fluid iteration cycles—if a user wants to slightly modify a design-in-progress, they can do so directly on a fabricated version of it, with the computational system working to keep the physical and digital representations synchronized [299].

Interactive fabrication might thus be seen as "hyper-rapid prototyping." Mueller identifies "a tight feedback loop"—the difference between a "turn-taking" interaction and a "direct manipulation" one—as a key part of making computational fabrication accessible to novice users [254]. As such, the technical aspects of this work often focus on reducing the length of iteration time. For example, the WirePrint system for filament deposition printers reduces production time by up to 10x by printing only a wireframe-like outline of a modeled object [252] and applying active cooling. RoMA ("Robotic Modeling Assistant") [297] combines Wireprint's high-speed low-resolution printing with an augmented reality (AR) head-up display to allow a user to modify their design in real-time with gestural input; they will see their changes in the AR overlay immediately, and the physical object soon afterward.

3D printing in particular can be framed as ideally a kind of "matter replicator"; for example, a panel discussion at CHI 2022 on the topic of digital fabrication and augmented reality stated that "both aim for a single end goal: creating 'objects' instantly" [79].

I note that these systems confront the experience/artifact tension by centering the knowledge gained through designing as the locus of value: to reduce iteration time, fabrication processes are often made "lower resolution," implying that they are merely simulacra, and underscoring their role as "prototype." Such interactive fabrication systems can particularly excel as "time saving systems," designed to embody as much expert knowledge as possible and delegating repetitive or time-consuming work to the machine.

11.3. Experimental, Experiential, and Hybrid Fabrication

On the other hand, other creative interactive fabrication systems might be better conceptualized as "time deepening systems" [123], where users are approaching interactivity for the intentional purposes of "disrupting" or dehabituating an otherwise familiar practice [76, 436] and supporting creative reflection. Because these systems make space for other agencies to confront, challenge, or otherwise disrupt their users, their experience may even be characterized as difficult or frustrating [269]; such "productive frustration" can itself be a positive experiential outcome. Hirsch deliberately confronts the temporal aspects of fabrication, with a range of approaches including both technical work in super-rapid fabrication with rapid crystallization and eutectic alloys, as well dance-based experiential fabrication performances [130].

"Hybrid" fabrication systems split creative work between multiple entities, possibly including software and hardware machine systems, creator-users and/or other humans [198], the materials themselves [152], and even non-human living entities [72, 210].

Hybrid systems which delegate part of the fabrication process to a human user can be seen as collaborative systems [176]; research in other areas of human-robot collaboration highlights that work-sharing arrangements can be designed to productively combine human strengths such as flexibility and context-awareness with machine system strengths such as precision and repeatability [111]. For example, in woodworking, small areas of digitally-assisted joinery can combine simple shapes into larger, complex structures. Tian et al's MatchSticks system [331] features a portable CNC router and a library of joints that can be accomplished with it. Magrisso et al's "Digital Joinery for Hybrid Carpentry" system uses optimized plastic joint components formed by selective laser sintering to connect basic wooden members. Leen et al's "JigFab" produces custom jigs to guide the user in more-traditional carpentry; by modularizing the design space of jigs, it allows for very fast construction of new jigs from standard components with small custom additions [199].

Pushing hybrid carpentry further, the FreeD system [436, 437] consists of a fully handheld milling tool and a positional tracking system. The user may move the milling tool anywhere in the working volume, and the default mode of the system is to carve only *around* the user-provided target shape—when the tool would cut into the target shape, the system turns off the cutting bit. However, the user can selectively override this scaffolding, as well as digitally mix the input shape alongside the carving process. In writing about this system, its authors emphasize experiential aspects of its use, and how the blend of machine and human agency is perceived by various users of the system. A hybrid system like FreeD is particular suitable for creators with deep expertise, who may want a high degree of real-time agency. Turn By Wire [379] describes a system which is explicitly computationally *mediated* while retaining a high degree of agency for the human user. Its interface is deliberately highly similar to a typical manual lathe — opaque to a novice, but intuitive for an expert — with an overlay of computational capabilities like repetition with variation and digitally-enforced endstops and guides.

However, such hybrid systems can also be designed for non-expert users, especially in support of a learning process. The 3D-printed "proxies" in the ProxyPrint system [382] encode skilled wirebending practices: stencils suggest the deconstruction of a shape into primitives, jigs enforce bending in torsional forces, and forging proxies provide guides to judge appropriate amounts of flattening. These proxies offer varying levels of support to novices versus experts, but each had the result of altering its user's construction approach. "Ground truth" proxies like jigs helped provide confidence and cognitive offloading for novices, which allowed them to grow their own practice.

Lastly, because of its perceived immediacy, experiential fabrication can also enable playful or otherwise alternative engagement with fabrication equipment. "Destructive Games" explores using a laser cutter as a component of a series of games which automatically destroy real-world objects as the stakes of playing [81]. "Loominary" uses a handloom as an input device for a choice-based adventure game [368]. My own "Threadsteading" situates a two-player territory defense game within machine sewing equipment (initially a quilting machine, then ported to an embroidery machine), which adds both strategy-deepening "waiting time" to each move, as well as a tangible trophy for the winner to keep. These works expand our ideas of what a fabrication machine can be, and for whom.

11.4. Representations and Modular Workflows

Even where it is not desirable or possible to directly intervene with a fabrication process—for example, on an industrial knitting machine with a mechanical complexity and operating speed that would make interacting with it during fabrication time dangerous for both machine and operator—the connection between creator specification and eventual output is a site of inquiry.

Many of the experiential fabrication systems discussed above operate on a basis of "what is done" (by the machine and, in the case of hybrid fabrication, by the creator) instead of "what will result." This differs from a classic CAD/CAM workflow, in which a user specifies their desired result in terms of its geometry, typically either using constructive solid geometry or with the point-mesh or boundary surface representations historically used in computer animation. From there, the operations needed to produce this result on a target fabrication machine are produced via automatic or user-assisted toolpath planning. For extrusion-based 3D printing, this can be quite automatic; it typically involves slicing the geometry into horizontal layers, generating contours, determining inner and outer areas and generating infill patterns, and converting contours and infill paths into GCODE instructions [214]. (For machine knitting: while working from a 3D point mesh representation is not at all the typical industrial workflow, it nonetheless is possible and might be seen as one way to make machine knitting "as easy to use as 3D printing" [234].)

Separating "what" (overall geometry) from "how" (specific machine operations) makes it possible to generalize to other machine types with similar output affordances: for example, from a cartesian 3D printer to a delta printer, or more drastically from additive 3D printing to subtractive milling. However, one tradeoff is that such pipelines necessarily reduce access to atypical capabilities of a particular machine and constrain outcomes to the specific affordances of the representation. For example, filament 3D printing techniques have been developed that use under-extrusion [95], over-extrusion [204], or unsupported drooping filament [296]; these are tuned to particular kinds of machines, and they are not well described by a solid geometry representation. Emergent material properties, extra functionalities of a fabrication machine, and experimental effects arising between the two—what I will describe as "grain" in Chapter 14—are typically not specifiable in an unmodified CAD/CAM workflow.

While the terms are highly related and are not always used in exactly these ways, within this work I use "representation" to mean a computational data structure of an outcome (as opposed to a sequence of actions), and "notation" for a user-facing manipulable form (likely by not necessarily either visual or textual). Often, a notation is built with a *visualization* of some aspect of the underlying representation; for example, three-dimensional geometry is typically visualized on computer screens through orthographic or perspective projection; different materials may show as different colors or patterns.

11.4.1. Representations and Notations

Alternative or additional representations for fabrication can be found in many projects that go beyond typical material outcomes. For example, an engineer might want to annotate a model with desired functional affordances to ensure that later processing steps do not prevent the outcome from performing as intended [133]. These can be represented as different "materials" within a solid geometry or point mesh model. Or a complex material like a magnetophoretic display might be designed as a parametric overlay onto a 3D mesh [424]. Extended representations such as these may be able to similarly use modified production pipelines. For example, the Geodesy system [119, 120] implements a specialized slicer to produce shape-changing artifacts which use a shape-memory effect tuned by the layer thickness and printing direction of a fused filament print, which would otherwise be secondary aesthetic concerns. By modifying just part of a typical 3D printing pipeline, this project benefits from existing tooling for the other parts, such as many possible tools for generating and modifying the point mesh representation format.

However, an outcome that is less about an overall geometry, or is otherwise complex in a way that is not well-supported by existing models, might require a less typical representation. Crucuially, any representation for fabrication must ultimately be transformed into the sequence of actions needed to produce it. This can be thought of as a *compiler*: "a program that can read a program in one language—the *source* language—and translate it into an equivalent program in another language—the *target* language" [4]. For complex domains like computational knitting, compilation can be quite complex; indeed, whether or not a particular output can be produced on a given knitting machine can be non-obvious [233], and very low-level machine operation considerations can propagate in surprising ways to the output object. Thus a machine-knitter, particularly one who is attempting an unusual or advanced structure, may wish to work with representations of knitting at *various* levels of abstraction—perhaps stepping through simulated machine instructions directly when debugging a knitting problem, perhaps treating segments of a surface as composable patches [132], or perhaps generating realistic predictive renderings of the overall object [166, 430]. I view these representations as complementary, each suited to particular parts of the possibility space of knitting, and I view approaches which can handle a multiplicity of such representations, and respective compilers where appropriate, as necessary to support creative exploration.

Once a representation exists, it must be presented to, and manipulated by, creators: it must have an accompanying notation. The success of a notation hinges on which aspects of the problem are relevant to the user, so notations are necessarily domain- and task-specific. They can be quite personal [221]. The Penrose system aims to translate between notations—written mathematical notations and conceptually appropriate diagrams—with an emphasis on extensibility to unique or even idiosyncratic domains [427]. (I discuss existing notations for weaving in Chapter 8 and knitting in Chapter 12.) Visual notations—diagrams—have a robust history in HCI for sense-making and information visualization. A prevailing argument for visual notations, as summarized in 1987 by Larkin and Simon is that "diagrams automatically support a large number of perceptual inferences, which are extremely easy for humans" [193]. *Interactive* notations can make complex domains tractable for casual readers; for example, Nicky Case's Explorable Explanations website hosts explanations of topics including large-scale social dynamics [267] and epidemiology [353].

Notations necessarily encode a level of abstraction and boundaries of what can be notated [396]. A notation makes it possible to document a possibility; a good notation makes it possible to compare possibilities in ways that are meaningful to the user. Iverson lists *suggestivity* as a hallmark property of a good notation: "A notation will be said to be suggestive if the forms of the expressions arising in one set of problems suggest related expressions which find application in other problems." [157] An "opinionated" representation — one that draws a tight boundary around what can be represented — can establish and enforce messier, softer aspects like "style." In Chapter 14, I discuss how developing an opinionated/stylized representation can be a generative act of tool-building.

11.4.2. Toolpath and CAM-based Workflows

Instead of manipulating a representation of an outcome, fabrication systems can center the actual sequence of fabrication actions as the locus of manipulation. Arguably, working in this way is much more like working with non-computational tools. Such systems might be thought of as speculative, as "alternative" to the dominant paradigm in computational fabrication [248], and can be referred to as "toolpath" [38] or "CAM-based" [67] design.

In a series of works centered on extruded clay printing, the Expressive Computing Lab has explored several CAM-based fabrication interaction systems [38, 67, 101, 248]. In one such system, a real-time/real-space hybrid pottery system consists of an extrusion printer directly mounted on an electronically-controllable pottery wheel; the potter can choose in real time to use the system as if it were fully un-augmented pottery wheel, or as if it were fully a 3D printer, or in an on-the-fly, hybrid manner with modular hardware control akin to musician foot pedals [248]. In another, a creator can directly doodle an extrusion path in a top-down view on a computer or tablet screen [101]. These two systems in particular are at the extremes of that lab's work in terms of how much physical hardware and real-time/real-space control is involved, but they share a powerful central workflow paradigm of using the machine "directly"; representations of the outcome per se are within the creator's head.

In Chapter 13, the computational intent and instructions are intrinsically operation-based. While the system does provide a lightweight predictive preview, the underlying model is that of the machine itself, not the yarn.

11.4.3. Compositional and Modular Workflows

Modular workflows are particularly relevant to softness. As Twigg-Smith points out, many creators switch between several modalities, likely developing their own unique blend of "direct specification tools, parametric tools, and translational tools" [388], and therefore grappling with how to "glue" their workflow together.

"Compositional fabrication" [177] describes an approach of using an assortment of compatible middleware to allow a user to steer the design of an output in ad hoc and fluid ways. Tran O'Leary's work aims to formalize composability for digital fabrication using programming language design [385] and visualizations underpinned by programmatic modularity [384]. Building on these, several systems by Tran O'Leary and Twigg-Smith propose flexible visual programming workflows for fabrication. Machine-O-Matic is positioned as a way to prototype workflows [386], whereas Dynamic Toolchains are presented as infrastructure to be used throughout a creative process [389]. Tandem emphasizes interoperability with other complex software, such as Fusion360, and shareability with collaborators [387] In weaving, AdaCAD is a full-featured weaving design system intended to support everything from simple patterns to complex weaving concerns like yarn connectivity for e-textiles [100] and multi-layer fabrics for artisan practice [77]. It therefore confronts difficult problems of representation and composition, from overall parametricity to control over specific interlacements. Dynamic Toolchains and AdaCAD use a "dataflow" paradigm, in which functional modifiers are represented in a graphical interface, typically with boxes representing functions, sockets representing the arguments and return values of those functions, and "wires" or "noodles" that can be drawn to connect them. Such interfaces have been used in several creative programming contexts such as in the music and video effects generating languages Max [433] and Puredata [309] and in the modeling and animation software Blender [98].

Together, these modular fabrication workflows combine the advantages of pipelines — broad support for various specific hardware systems — with on-the-fly reconfigurability and extensibility. While I have focused on less screen-based interaction in this dissertation, modularity and the potential for composition are at the heart of how I have engineered every system in this dissertation. Particularly, the "brioche format" in Chapter 14 is designed to be flexibly modifiable.

12. Fabrication Technique: Knitting Again

In Part II, I centered weaving for its rich histories and range of possible contexts. Here in Part III, I revisit knitting, which I described in terms of low-level operations in Chapter 4. In this chapter, I provide background on tools and workflows for knitting; because of the tremendous possible complexity of knitting, these are numerous, and they range from general to highly task-specific.

Knitting can be done in various ways: by hand on straight needles or on knitting frames, and on knitting machines of a wide range of complexity. While the resulting structures can be quite similar, the resources required for each production technique can be very different and this is reflected in their typical outcomes. For example, a hand-knitting pattern should ideally be straightforward to understand and remember (e.g. a set of operations is repeated with minor variations), and it may minimize maneuvers that are physically tricky for beginners. Conversely, in machine-knitting, some operations are downright impossible that a hand-knitter would find trivial. Hand-knit structures typically have larger loops than machine-knit ones, and therefore may prioritize decorative manipulations at the scale of small numbers of loops (e.g. “cables,” in which columns of loops cross each other) which would be illegibly small in most machine-knitting; machine knitting may instead tend toward all-over textures that require many loops per pattern element.

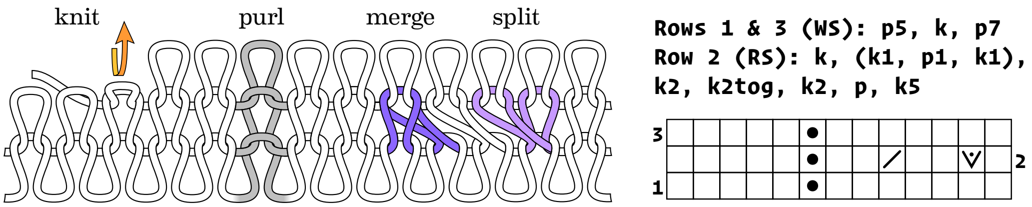
Not all knitting machines are as automatic as the industrial v-bed machine I described in Part I. Indeed, many knitting machines are fully or semi-manual, requiring a human operator to execute the knitting process. Some have electronic patterning control, similar to the Jacquard handlooms in Part II, while some support mechanical patterning via cams or punchcards [62] and some rely entirely on the operator to produce any patterning beyond a single-color rectangle. Depending on the machine, tools for this kind of knitting are likely to include some of the concerns of hand-knitting, such as making the patterning legible to the knitter, as well as some of the concerns of automatic machine knitting, such as low-level needle allocation and electronically interfacing with the machine.

In representing knitting textures computationally, it is particularly tempting to reach for “pixel” representations of knitting. Just as with woven interlacements, though, a pixel is an incomplete proxy for a knit loop. While some knit textures are indeed “colorwork”—patterns in which the color of each stitch is the main design element, as in pixel art—many others are not, and a “pixel” approach to representing them can obscure their rich design spaces.

12.1. Representations and Tools for Hand Knitting

All of the knit work in this dissertation was produced on either an industrial or semi-manual knitting machine. However, the notations of hand knitting are unavoidably influential in machine knitting, so I will summarize them here.

Hand-knitting patterns have traditionally been disseminated in a textual form often called “knitspeak” [132], which is a sequence of abbreviated terms directly representing specific loop-level operations for a hand-knitter to enact. Knitspeak was developed in the late 1800's and, being textual, was easy to typeset into the printed handbooks and magazines that were the primary way of disseminating knitting knowledge at the time [332]. However, it is inherently sequential, not 2- or 3-dimensional like the outcome, and not particularly amenable to direct creative manipulation. Instead, knitters typically plan, and increasingly frequently disseminate, their projects using a combination of sewing-pattern-like sketches of overall shaping alongside a grid-based notation called a *chart*. Charts are particularly often used to represent a graphic motif which will be repeated within the overall shaping. While many knitters will simply use graph paper or even spreadsheet software to plan charts, purpose-built applications for knit charting [2, 44, 47, 241, 397] are typically lightweight tools for correcting aspect ratios, previewing repeats/tilings, planning color combinations and yarn quantity requirements, and exporting charts to be followed by other knitters; some include modes for other chart-based techniques like counted stitch embroidery. The popular DesignAKnit software [41] builds on these simpler knit charting applications by also including tools for overall shaping (which are similar to the same company's "Fittingly Sew" tools for sewn garment design), as well as machine control capabilities for home knitting machines and real-time audio instructions for hand- and home-machine-knitting.



```
Rows 1 & 3 (WS): p5, k, p7
Row 2 (RS): k, (k1, p1, k1),
k2, k2tog, k2, p, k5
```

(a)

(b)

Figure 12.1 *Basic knit loops and notations. a) A small area of knitting showing the default "knit" (a loop pulled from the back to the front), "purl" (a loop pulled from front to back), and how columns of the grid can be merged and split. b) Two typical hand-knitting notations for this knitting—textual "knitspeak" above, and a chart below.*

Although in many cases there is a simple correspondence, a gridded chart fundamentally notates operations, not necessarily outcomes. For example, a single operation resulting in two loops would be represented as single chart cell. Additionally, knitting easily supports overall fabric shaping, such as non-rectangular boundaries and double curvature. Unlike a weaving draft, which can represent one surface of weaving fairly literally because fabric woven on a frame or Jacquard loom is (at least while it is being woven) flat and rectangular, a rectangular knitting chart which may result in a non-rectangular surface is inherently somewhat abstract. The charting software Stitch Maps [43] is designed to offset this by combining typical charting symbols with a non-rectangular grid, though Stitch Maps requires valid textual knitspeak instructions as input. (A similar non-rectangular charting format is much more common in the adjacent handcraft domain of crochet, which particularly excels at overall shaping and topology.)

Regardless of notation system, hand-knitting instructions often prioritize practicality, e.g. by regularly repeating a short sequence to allow the knitter to work from memory. Tools that support hand-knitting may include functionality like the ability to highlight the currently in-progress row [43] or audio cues [41]. Most of these tools are software-only, but the eLoominate project augments a purpose-build peg knitting loom—a type of jig for hand-knitting, often used for teaching novices—with LED indicators to guide users through simple two-color patterns designed in advance on an accompanying application [114]. This use of real-time record-keeping assistance, as opposed to automation, was influential to the augmented knitting machine in Chapter 13.

12.2. Representations and Tools for Industrial Machine Knitting

In industrial knitting, each of the major two knitting machine manufacturers, Stoll and Shima Seiki, provide a proprietary design software to interface with their machines [362]. (The industrial knitting machine used in this work is Carnegie Mellon University Textiles Lab's Shima Seiki SWG091N2. Like most other members of the Textiles Lab, I typically use the open Knitout format [232] as a intermediary between my knitting systems and Shima's Knitpaint format. I occasionally, but very rarely, edit knitting patterns directly in Knitpaint; this is typically either to make a quick fix or, if there is a guest in the lab, for pedagogical or comedic effect.)

Stoll's M1Plus and Shima's SDS-ONE design systems offer similar capabilities, including grid-based charting, templates for shaped knits, and machine operation preview visualizations for debugging, as well as industry-targeted capabilities such as remote machine access and ticketing/management systems. While both companies make *machines* that are capable of fully three-dimensional shaping—Shima Seiki calls this "Wholegarment" and Stoll calls it "knit&wear"—neither *design system* offers general 3D editing capabilities; instead, several template shapes such as "sweater" and "tubular object" can be parametrically modified through a "wizard" interface.

Within the research world, machine knitting has been recognized as a powerful fabrication technology particularly in HCI and Graphics research. This work has primarily focused on fully automatic computational knitting, both as a means to produce technical materials and as a target for improved design and fabrication pipelines. Some of this work is driven by a vision of machine knitting as a "general purpose fabrication machine": from building an improved technical stack at the low level by establishing the Knitout standard [232] and investigating medium-level compilation to that format from shaping primitives [233] to generating shaping patterns from 3D point mesh models [258, 262]. Especially at the level of individual loops and needle operations, mathematical approaches have been proposed to optimizing machine efficiency and reliability [205, 207], especially by solving fundamental representation problems for provable equivalence [206].

Getting more specific to the affordances and contexts of knitting, the "stitch mesh" computational model, which represents knit stitches as nodes connected by yarnwise and loopwise edges, has been established to describe knitting (e.g. for rendering images [430]), as a basis for establishing knittability [415] and constructing composable texture patterns [263], as an exchange format that hand-knitting patterns can be interpreted to [132], and even as the basis of a model for the adjacent technique of crochet [122]. Other mid-level representations include composable and parametric "template" patches [162], and a fabric-native system for representing knit surfaces via flat patterns similar to sewing patterns, to provide a familiar interaction paradigm for garment-design experts [173]. Systems for specifying texture include the KnitPick system which presents a database of textures harvested from hand-knitting patterns and a method for composing and modifiying them [132] and an approach to extracting predicted knitting operations from a photograph of an existing knit swatch using neural network instruction synthesis [172]. Technically, any of these are about knitting broadly, not just machine-knitting (and the earlier stitch mesh projects do not include compilation to the machine), but for the most part they do not produce practical hand-knitting patterns—the complexity of the output is driven by the capabilities of the machine.

12.3. Representations and Tools for Hybrid & Manual Machine Knitting

As a cross between the support for computational complexity of industrial machine knitting and the possibilities for creative intervention (not to mention the much greater accessibility) of hand-knitting, manual knitting machines have been the locus of a wide range of creative work both in academia and outside of it. A tremendous range of knit patterns can be produced on such machines [62] and, at their least expensive, such machines are affordable for casual hobbyists; mid-range machines are frequently used by students in textile design and fashion schools as well as more serious hobbyists. Overall shaping is certainly possible on this type of knitting machine, but it typically requires frequent hands-on intervention. These machines are therefore particularly used for texture knitting, especially that which can be produced with changing yarn carriers and combinations of the “knit”, “tuck”, and “miss” needle operations (i.e. not “transfer”).

Less-automatic machines have primarily been explored in research on the hardware side. All Yarns Are Beautiful (AYAB) [20] is an open-source project which documents an Arduino-based replacement controller for the 1980’s Brother ElectroKnit series of computerized home knitting machines—these machines are manually operated (non-motorized), but the computer controls specific patterning (typically used for two-color “pixel art”-style patterning, but more complex styles are possible too [390]). Another open-source project, OpenKnit [329], sought to make it possible for a hobbyist to build a knitting machine from off-the-shelf and 3D printed parts. Depending on the build, an OpenKnit machine could be fully manual or mostly automatic. (OpenKnit has since become a hardware startup, Kniterate [330], which makes fully-automatic industrial-style machines for non- or less-industrial contexts.) AYAB and OpenKnit are both long-running projects with active communities (as of this writing, the OpenKnit Instructable has over 111,000 views [329], and the AYAB discussion group on Ravelry has 362 members [93]); other, smaller projects include small-run specialty tools for automating color changes [363] and repeating patterns across the width of a knit [187]. These hobbyist-led innovations have supported interactive art [339] as well as experimental architecture research [26]. Together, these projects highlight both a community interest in manual machine knitting as well as opportunities for creative practice.

Manual machine-knitting is a skill that can require a fairly deep knowledge of the knitting machine. The notations for it can be opaque and niche, as they often depend on the affordances of specific brands and models of machine [62]. These include: diagrammatic views of the selected needles and cam settings [63], especially for less-automated machines; straightforward reproductions of the punched card or marked mylar input to semi-automated machines (e.g. [65]); and simple grids for colorwork patterns [64], as in hand-knitting. The ability of even low-feature manual knitting machines to pattern using tuck and miss (often called "slip" in this context) operations—thereby extending individual stitches past the borders of a simple grid cell notation—complicates the correspondence between grid notations and the actual knit output.

In a series of work on creativity support tools particularly targeting the the patterning affordances of a Silver Reed domestic machine with carriage-based selection, Twigg-Smith has developed a semi-simulated notation for tuck- and miss- based patterning [390]. Unlike much of the existing work in fabric simulation, which is driven by graphics researchers and thus aims to produce highly realistic graphical renderings suitable for product imagery or animated films [166, 430], Twigg-Smith's system is diagrammatic: it combines a familiar grid-based input system with a light 2D physical simulation to show the locally-distorting effects of this type of patterning, without introducing visual complications. I similarly used a mass-spring simulation in the work in Chapter 13, for much the same reasons; in my case, I targeted a style of knitting ("racked rib") with loops that overlapped laterally, so I abstracted the specific visual motif of loops to make the distortion effects more legible to novices.

12.4. New Workflows

I describe this thriving ecosystem of tools for hand- and machine-knitting not because I intend to replace or obviate it, but because it's a mature example of how fabrication practices have developed to encompass numerous overlapping concerns, from the material to the contextual. Recurring notations and abstractions can be a strong foundation for new ones, as in the case of the Stitch Maps modified charting system, and existing technical systems can be modified and remixed, as in the case of the AYAB hardware control project. These practices can be inspirational to any domain of fabrication, and creativity broadly.

⁂ Experiential Workflows for Knitting

In Chapter 13 and Chapter 14, I describe two systems for machine knitting that support improvisational, underdetermined creative outcomes in machine knitting. Each builds upon a worked-through computational model of machine knitting at sufficient complexity to unlock creative potential and showcase the rich potential of the fabrication technology, while encapsulating a specific design space within it.

In Chapter 13, I show how an existing non-computational workflow can be computationally augmented. In Chapter 14, I introduce the concept of a *grain space* and use it to develop a modular system for knitting "brioche" fabric. Each system presents a tension between creator agency and control, and I use each as an opportunity for discussions with creators on their own relationship to personal and improvisational workflows.

The work in this Part exemplifies the role of **modular, emergent, and deployable workflows** to channel the tendencies and opportunities of low-level material properties and operations toward experiential and situated creativity.

13. Augmenting a Manual Knitting Machine

Manual knitting machines, as described in Section 12.3, are an opportunity to approach two specific challenges that prevent computational fabrication from being used at its full potential, for all the varied tasks supported by traditional fabrication. One is the rigidity of mono-directional CAD/CAM pipelines, as discussed in Chapter 11. Another is that specialized computational fabrication machines are often highly expensive and fragile, which greatly undercuts their availability for novices or for experimental tasks. Manual machines are much less expensive, and much more robust to human interaction, than automated ones. (Indeed, they do actually require such interaction.) The physical scale of their output is well-suited to hands-on, real-space experimentation, and, as with the looms in Part II, the requisite manual operation of the machine itself lends itself to real-time design. This type of machine is already common in non-production contexts as learning tools for textiles design students, as prototyping equipment, and for hobbyist use.

However, while these machines can certainly produce fabric faster than hand-knitting on pointed needles, they do not necessarily require any less expertise. Indeed, the relationship between the user's operational input and the final object is arguably even even more obscured than in hand-knitting, because the most recently formed fabric is hidden from sight behind the needle beds, preventing users from getting timely feedback on their actions. Additionally, like many well-developed but manual fabrication processes and machines, knitting machines can have complicated, inter-related mechanical settings.

These challenges and opportunities of manual machine knitting are the basis of the work in this chapter: a case study for how a "lower tech" manually operated machine can be *augmented* with new capabilities using lightweight sensing and simulation methods. By combining machine state tracking with domain-aware interaction modules, my system provides immediate feedback about the recent past, current, and potential future states of the machine. This 1) enables creative access to the otherwise opaque fabrication process of manual machine knitting, broadening access to machine knitting overall as a fabrication technique and 2) provides an example of on-machine interaction using the machine as an immediate and embodied input, with implications for experience of working with the machine, especially for novice users.

This work diverges from typical Interactive Fabrication research in that the underlying fabrication machine is not electronic or even electrical; even with augmentation, the system does not autonomously produce physical output. The machine itself was always intended to be operated by hand, so some common concerns in Interactive Fabrication, like reduced iteration time and imposing additional safety features, are less relevant here. Instead, the focus of this work is on how to develop and communicate real-time and real-space interpretive feedback in a lightweight and modular way.

13.1. Operating a Manual Knitting Machine

The work in this chapter is based on a Dubied NHF4 manual knitting machine which is functionally very similar to models made by the same company in the early 1900s, and even fairly similar to its predecessors from the mid-1800's. Like many other mature machine technologies, its operation interface is constructed for reliability and relative power of expression, not for legibility or ease of use. I'll summarize its operation and patterning affordances in this section to form some basis for understanding my interface augmentations, as well as to underscore the difficulty of learning to use a manual knitting machine unassisted.

As with the industrial computer-controlled knitting machines described in Chapter 4, manual knitting machines form fabric on hook-shaped *needles*. These needles are arranged in parallel in individual slots on *beds*. The simplest manual knitting machines might have just one of these beds, in which case the needles run parallel to the floor, with the hook end of each needle facing the user. Our Dubied machine has two beds, arranged in the same inverted "v" as the Shima Seiki machine I used in Part I. (In a domestic machine, the second bed might be sold as an optional attachment, and referred to as a "ribber.")

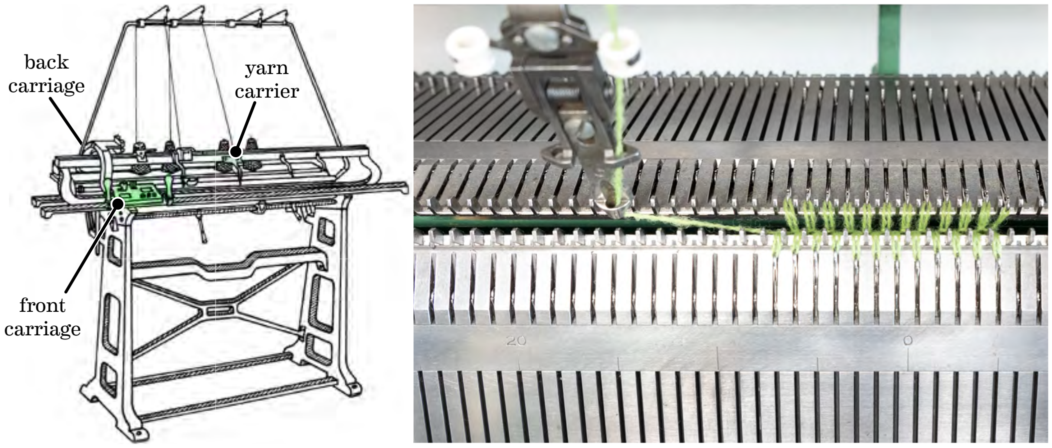


Figure 13.1 *Right: the overall layout of a v-bed manual knitting machine, showing the carriage (Image modified from Wikimedia Commons, [83]). Right: a view of the needle bed and one carrier on my machine.*

The *carriage* is the main point of contact for the user. The carriage has two main roles. First, it controls the *yarn carriers* which position yarn in front of each needle; these may be directly integrated into the carriage body, or, as in the machine used in this work, the carriage may be able to selectively engage separate but passive carriers for multi-color knitting. (Industrial computer-controlled knitting machines coordinate the carriage and carriers either with selective mechanical engagement or with electronic synchronization.)

Second, the carriage contains a set of cams which, when the carriage is slid across the needle bed, push the needles up and down along their slots to carry out the operations of knitting. These operations include the eponymous "knit" operation, in which the needle is pushed forward to catch the yarn from a carrier, then pushed back down to pull a loop of the yarn through any previous loops on that needle, dropping those previous loops in the process (Figure 4.3 in Section 4.1). The other most typical operation is a "tuck," which also grabs a loop of yarn from the carrier, but does not pull it through existing loops, instead incrementing the number of loops on that needle (Figure 4.4 in Section 4.1). A group of controls on the carriage configures the pattern of stitches carried out across a row. All together, these operations determine the loop-to-loop connections of the knitting, with effects in the knit surface's stretchiness, density, and surface patterning.

To operate a manual knitting machine, the user must push the carriage across the needle bed for each row, alternating leftward and rightward passes. This action can require up to 15 lbs of force, and knitters typically stand at the machine to operate it.

The knitter can also transfer stitches (move them from one needle to another, as in Figure 4.5 in Section 4.1), but this is not an automatic operation as it is in industrial knitting; it must be done by hand, in an operation that takes some skill to perform quickly or reliably. (Nonetheless, these hand-manipulated stitches can greatly increase the repertoire of a manual machine-knitter [121].)

13.1.1. Patterning Affordances of Manual Machine Knitting

While operating the machine, a knitter can adjust the various cam settings of the carriage, as well as engage different yarn carriers. When the knitter moves the carriage across the needle bed for each row of knitting, the carriage cams guide the needles up and down in their individual slots for knitting and tucking. The cam settings for a given pass of the carriage will select which needles will knit, which will tuck, and which will be missed (passed by without being actuated). On knitting machines intended to support tubular knitting, as ours is, these carriage cam settings can be allocated independently for each direction of pass, per bed. Therefore, the knobs are repeated for each of leftward on the front bed, rightward on the front bed, leftward on the back bed, and rightward on the back bed.

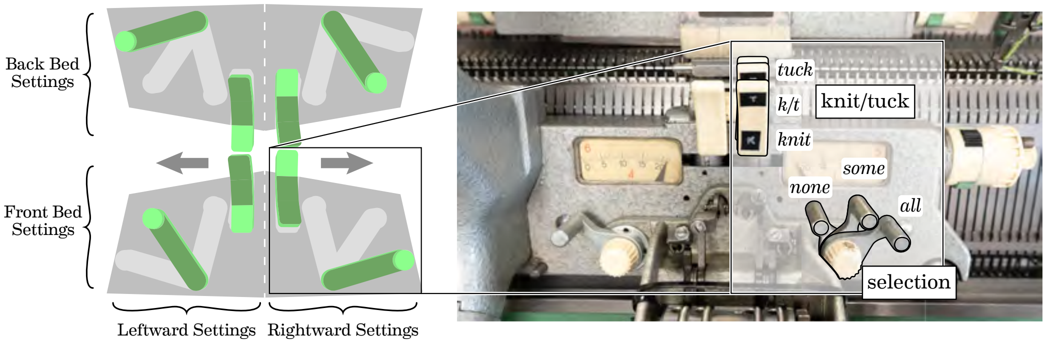


Figure 13.2 *Left: a bird's-eye diagram of the carriages shows how a set of two switches are mirrored for the leftward and rightward directions on each of the front and back carriages. Right: Within each bed/direction set, there are two switches—"knit/tuck" and "selection"—each with three possible positions. The "selection" switch rotates on the face of the carriage, and "knit/tuck" is a rocker switch.*

Within a bed-direction, selection is based on needle type. On the Dubied NHF4, there are two types of needles: "high" needles and "low" needles, referring to the distance the needle's selector tab protrudes from the surface of the bed. For each bed-direction, the cam settings are presented as a set of two switches Figure 13.2 which alter the selection cams' proximity to the needle bed. If a cam is brought close to the bed it will catch and actuate all of the needles (both high and low). It might otherwise be brought away from the bed to miss all of the needles, or it might be adjusted to a distance where it actuates just the "high" ones without the "low." It is not possible to select just the "low" without the "high."

Therefore the three positions of the *selection switch* are: "select all needles," "select high needles only," and "select no needles"; the three positions of the *knit/tuck switch* are "knit all selected needles," "knit the high and tuck the low, assuming low are selected," (abbreviated as "k/t") and "tuck all selected needles."

Figure 13.3 *"High" and "low" needles differ only in the height of the selector tab, which is the part that protrudes out of the needle bed to be caught by the carriage cams.*

The choice of which type of needle should be in each slot on the bed is established before knitting begins, and stays the same throughout the knit job. (Individual needles may additionally be put "out of work"—that is, set to a position where the cams don't actuate it, regardless of their settings—during knitting. Because this can create problems, such as jamming when a yarn loop prevents the needle from being taken fully out of work, my system never suggests it. None of the knit fabrics shown in this work required changing needle allocation during knitting.) However, the combination of selection and knit/tuck settings can still give rise to complex behaviors. The full range of how the switch settings interact with the two needle selection sets is summarized in Table 13.1.

Setting		**Outcome**	
Selection	**Knit/Tuck**	**High Needle**	**Low Needle**
all	knit	knit	knit
all	k/t	knit	tuck
all	tuck	tuck	tuck
high	knit	knit	miss
high	k/t	knit	miss
high	tuck	tuck	miss
none	knit	miss	miss
none	k/t	miss	miss
none	tuck	miss	miss

Table 13.1 *Operations from Switch Settings*

Because these settings are allocated per bed-direction, a basic two-row-long sequence of operations can be performed without changing settings. For example, the front bed carriage might have its leftward settings be "knit all" and its righward settings be "knit none," with the back bed set to "knit none" needles leftward and "knit all" rightward. The result of this would be a tubular knit, in which the knitting proceeds in a spiral, leftwards on the front bed and rightwards on the back, without the knitter needing to change settings between rows.

Lastly, the machine's *rack* lever changes the alignment between the front and back beds. At the neutral position, the two beds are aligned with each back bed needle almost directly across from its corresponding front bed needle, and it can be adjusted rightward or leftward by three needle-widths in each direction. In fully automatic machine knitting, the rack alignment is primarily used in conjunction with "transfer" operations to move stitches around (as in Figure 4.6). On a manual knitting machine, transferring is done manually and does not depend on changes in bed alignment; however, the rack lever enables a unique category of knitting patterns known as "racked rib." In these, the rack position is changed between knitting passes of a fabric formed on both beds per row, as in a "rib" (alternating front and back knits) or "cardigan" (a lofty fabric in which each row knits on one bed and tucks on the other). The changes in rack position entangle the columns of stitches, producing fabrics with puckers, tight zigs-zags, or meandering waves.

Together, this system is fairly powerful, enabling knitting a variety of structures such as tubes, ribbing, and cardigan without frequent settings changes on the part of the knitter. However, it is also highly nonintuitive for a beginner. The potential for frustrating accidents, such as causing tension problems by tucking or missing the same needle too many passes in a row, is high, and recovering from such errors can involve painstakingly picking yarn out of needles and re-starting the entire knit piece. To make matters worse, the newly-formed stitches hang between the two opaque metal beds of a v-bed machine and are thus not even visible to the knitter until many rows later. In the case of "racked rib" patterning, the resulting fabric can be quite complex and difficult to visualize; additionally, this technique is rare in hand knitting (where there are no "beds"), so it is likely to be an unfamiliar type of patterning even to users with a hand-knitting background.

13.2. Implementing an Augmented Knitting Machine

My machine augmentations interpret the machine's settings for the knitter, and visualize possible results. As an on-machine interface, the system uses the machine itself as the input: the system tracks the physical cam and rack settings of the machine, and uses changes in the carriage position to determine when rows have been knit. These changes are reflected in a visual display which shows the current state of the machine (including recently-knit rows which may not be visible yet on the actual machine) as well as optional additional modules such as patterning guidance. I diagram the technical implementation of the system in Figure 13.4.

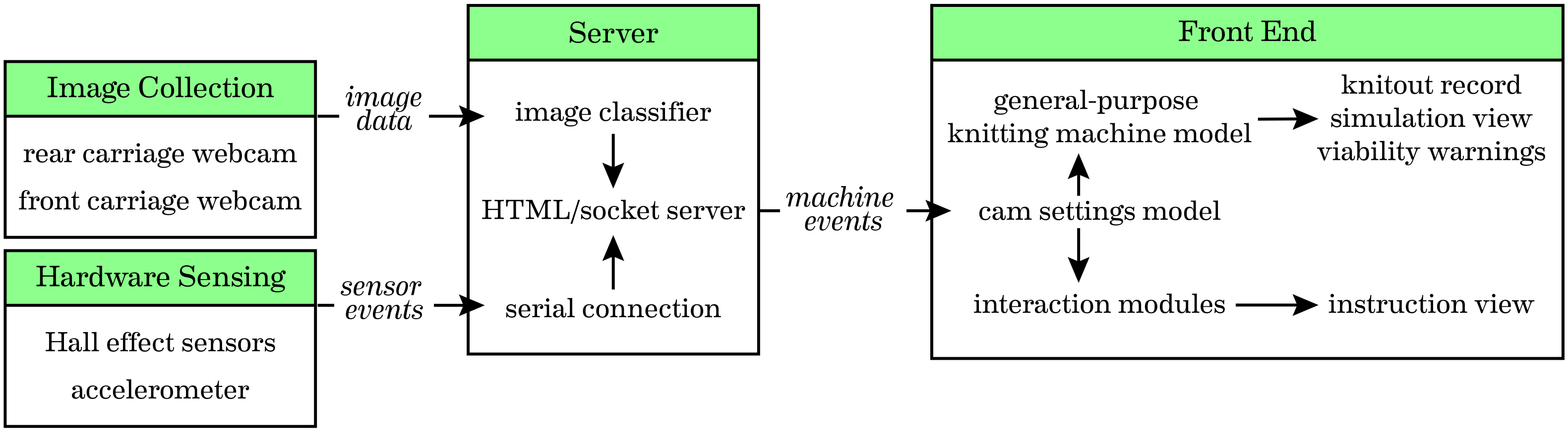


Figure 13.4 *The system combines hardware and computer vision as input to drive a machine simulation and other interaction modules.*

13.2.1. Sensing

I chose lightweight methods to capture the machine's settings at a given time. To capture the racking position, I mounted a simple 3-axis accelerometer (GY-61 ADXL335) to the racking lever at the side of the machine Figure 13.5, left). To sense carriage position, I mounted Hall effect sensors at the left and right sides of the machine to be triggered by magnets attached to the carriage. These sensors are mounted on rails and are positioned to be just outside the knitting area for a given task (e.g. for a narrower fabric, they can be brought closer to the center of the machine). I sense the left and right positions separately to support "leaving one position but not yet arriving at the other" as an input gesture. I use an Arduino to debounce these hardware sensor inputs and send change event notifications over USB serial.

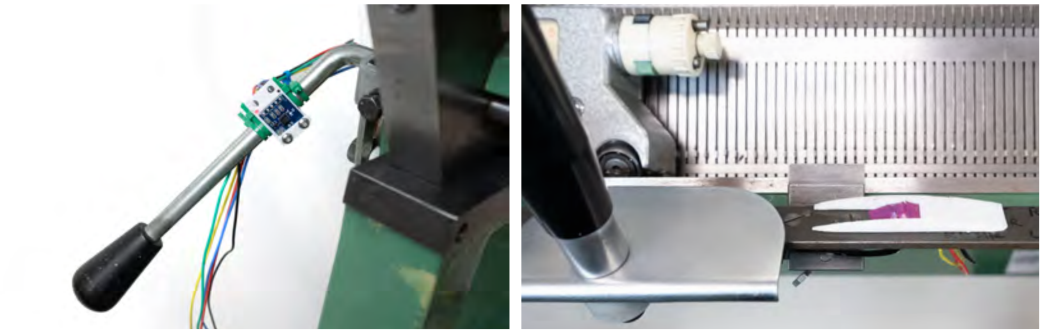


Figure 13.5 *The system's hardware modifications to the machine are all removeable with no damage to the underlying machine. Left: rack lever position is sensed with a three-axis accelerometer. Right: Hall effect sensors are mounted to the rails that are intended for use with a mechanical row-counter. Magnets attached near the handles of the carriage pass over the sensors at the end of each row.*

Because the cam switches are mechanically complex and somewhat numerous, I decided against hardware sensing for their positions. Instead, I used computer vision: I mounted two webcams to the bow of the carriage Figure 13.5, right), with one each pointed to the front and back carriages Figure 13.7). During system use, a Processing sketch captures data within calibrated crop areas of the webcams.

Figure 13.6 *A pair of webcams is mounted to the "bow" that connects the front bed and back bed carriages. Each camera is positioned to capture the switch positions for its carriage*

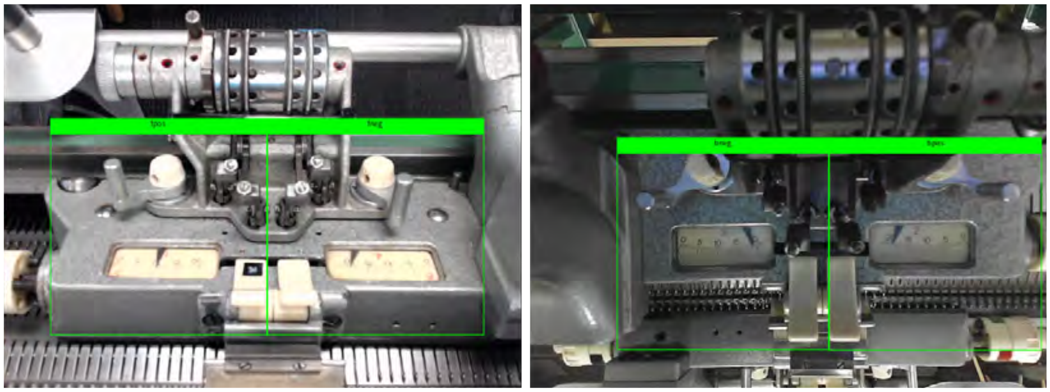


Figure 13.7 *The views from the webcams. Each camera captures the two dial sets belonging to one side of the carriage.*

For image classification, I used TensorFlow.js [1] with a separate model for each of the four switch sets (front leftward and rightward, and back leftward and rightward). Each model has ten classes: one for each of nine setting combinations (as listed in Table 13.1), plus one for "hands visible in the image," to minimize updating the switch position display while the knitter is in the middle of adjusting a switch. I captured approximately 320 images of each switch set position group (e.g. "tuck on all needles, for back bed rightward rows") using a second Processing sketch to manage the webcams and organize the data for each class. During image capture, I stored webcam input slightly outside the calibrated crop areas so that I could later augment the image data with randomly-chosen subcrops at the final image size. This process took approximately a half hour. I manually sorted out images with hands visible into a separate "hand" class for each switch set, then augmented the approximately 250 images remaining in each other class with randomized crops, blurring, and image contrast to a total of approximately 1600 images per knob set. Using a basic Keras model on a personal computer with an RTX 3070 GPU, I trained a three layer convolutional neural network with 1.6 million parameters. Training took thirty seconds per model, and reported 99.32% accuracy when reserving 20% of input images as validation data. I did not fine-tune my approach beyond what was suggested in an online tutorial [374], suggesting that comparable results do not require particular machine learning expertise.

I coordinate these sensors with a server written in Node which accepts the carriage and rack change events from the Arduino as well as image data from the Processing webcam sketch, uses Tensorflow.js to classify the image data, and passes machine state events to the frontend user interface over a websocket.

13.2.2. Machine model

On the front end, I have modeled the knitting machine state including carriage position, yarn carriers, bed rack position, and a graph representation of the knit fabric being formed.

Figure 13.8 *Because I track the machine operations, I can "replay" them on any Knitout-compatible machine. In this case, I have knit a duplicate scarf on a Shima Seiki SWG091N2, which has a much smaller stitch size than the Dubied.*

This underlying machine model is compatible with the Knitout knitting machine operation language spec [232], and I maintain an operation history that can be "replayed" on any Knitout-compatible computer-controlled knitting machine (Figure 13.8). This could allow a knitter to design interactively, then use an automated knitting machine to create multiple duplicates, or to knit at a different stitch size. On top of the basic needle-by-needle abstraction of Knitout, I model the carriage cam settings and needle types ("high" or "low"). Lastly, I maintain both 1) committed machine states, representing operations the knitter has already taken, and 2) potential machine states, representing possible futures given changes in the machine settings.

13.2.3. Visualization

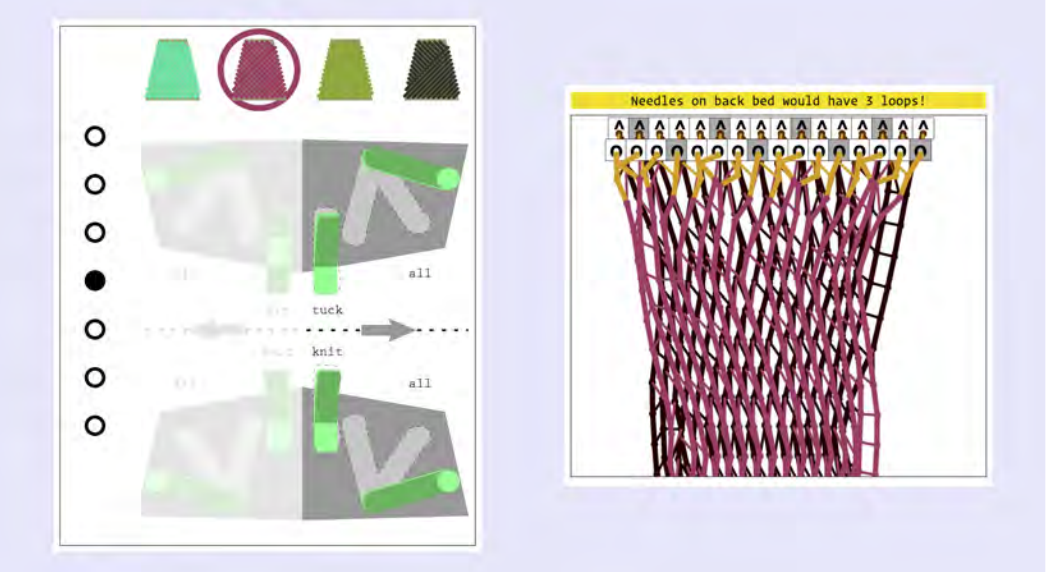
My front end system comprises several visualization modules which are written as interoperable JavaScript classes. (These can also can be used as input devices themselves—while not part of the main on-machine interface scope of this work, this capability does allow a user to practice knitting virtually.)

I render carriage, rack, and yarn settings diagrammatically, with textual labels for the switch settings, Figure 13.9. When the user changes a cam or rack setting on the physical machine, this view is automatically updated to indicated the current settings. I render the machine as a simplified needle bed, with the needles aligned according to the current rack position and a symbol on each needle showing which operation would be applied at that needle if a row were made with the current settings.

The in-progress knitting is visualized using a mass-spring simulation, with the back bed yarn connections shaded slightly darker than the front. I abstract the stitch connections in the fabric into simple nodes and edges, instead of showing a literal yarn path, for readability. (I chose the mass-spring simulation for its particular suitability in showing how columns of stitches deflect in the "racked" patterning I highlight in Chapter 13.) In this view, the rows that have already been knit are displayed in a yarn color, and future row predictions/suggestions are tinted yellow.

I also created a sequential panel representation of my "pattern rows" notation. Each panel shows the cam and rack settings, carriage direction, and yarn carrier needed to reproduce a particular row. When displayed as part of live instruction set, each panel highlights the changes the knitter would need to make to follow that instruction.

These machine state and instruction panel views form the visual basis of the patterning interface modules I describe in Section 13.3.



Needles on back bed would have 3 loops!

Figure 13.9 *A screenshot of my basic machine visualization. On the left, a diagrammatic rendering of the carriage shows the machine's current switch and rack settings. On the right, a mass-spring simulation shows the fabric that is being formed. The rows of knitting that are tinted yellow are a projection based on the current machine settings.*

13.2.4. Error checking

Because I model the fabric being formed and its relation to the machine, I can add error checking for problematic operations. For example, in Figure 13.9, the interface shows an error that "Needles on back bed would have 3 loops!" Needles can only hold so many loops, so when additional loops are added by successive tucking operations without intervening knit operations, the knitter runs a risk of overloading the needle, leading to dropped or torn loops. The machine model tracks the status of each needle, and can provide warning for certain conditions in either the committed or projected machine states.

13.3. Interface Modules

Using the machine model and visualizations as component parts, I created three modules to show opportunities for learning, carrying out specific tasks to create functional patterns, and working improvisationally on the knitting machine.

13.3.1. Basic Operational Assistance

First, I created a view that provides an interpretation for the knitter of the interconnected machine settings and their effects on the next rows to be knit. In this view, the diagram of the cam and rack settings is shown live alongside a simulation of the existing fabric and a preview of what the next two rows of knitting would look like with the current settings. The cam settings are labeled with the name of their position ("all"/"some"/"none" and "knit"/"kt"/"tuck"), and the diagrammatic view of the needles displays the operations as the would occur in the next pass at the current settings. When the knitter changes a cam or rack setting, these views update accordingly. The fabric display shows the recent rows that are still hidden from physical view behind the machine beds. Lastly, this module displays error checking messages to warn the knitter about potentially risky operations they have performed or would perform. This module therefore collects and displays information about the recent past, present, and potential near future states of the machine and fabric, giving the knitter information but not imposing any particular guidance.

13.3.2. Production Assistance for Function Integration: Pockets

The second module is intended to help a knitter produce a specific outcome. I focused on producing fabrics with two-layer "tubular knit" areas, which could be used as open pockets or as closed regions to contain other materials, Figure 13.10. This knitting style requires the user to plan the locations of High and Low needles, and to change cam settings at the beginning and end of the pocket section. If the user wants to knit a pocket which is open on one side of the knit, they will additionally need to switch cam settings every other row, even within the pocket section.

Figure 13.10 *To produce open pockets such as these, the knitter must switch the carriage cam settings every other row, and the rack setting every row. The system helps the knitter keep track of these.*

The "Pockets" module provides a simple sketch-like interface to plan pocket locations, Figure 13.11. During knitting, it shows the knitter's progress through the plan and provides row-by-row cam setting guidance.

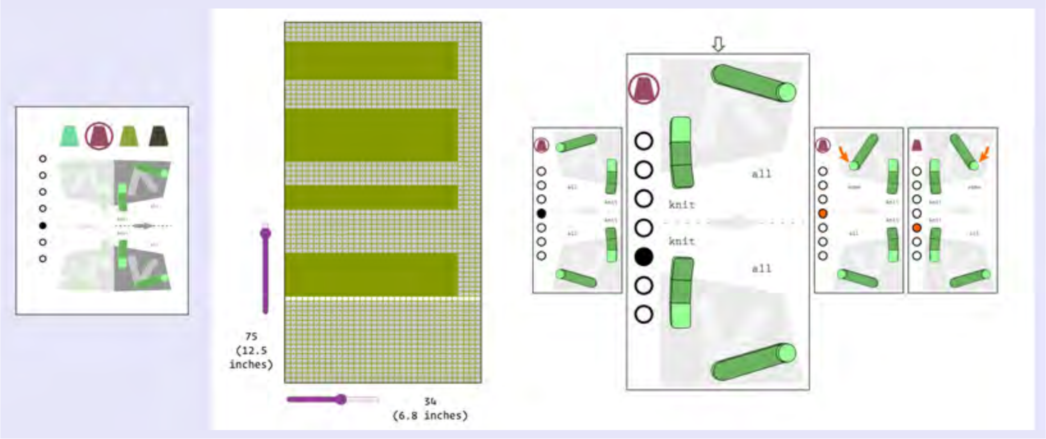


Figure 13.11 *The pocket-knitting interface. The left side panel shows the current machine settings. The center panel is an editable area in which the knitter lays out pockets. The right panel contains a scrolling sequence of instructions, with the next instruction magnified. If the knitter needs to change a machine setting, the instruction panel will highlight the needed changes with orange arrows.*

In Figure 13.12, I showcase an advantage of manual knitting machines over industrial knitting: a greater range of possibilities for integrating additional materials into the knit. (This would be dangerous and difficult with a high-gauge, fast-moving, delicate industrial machine.) In particular, items slimmer than the gap between the beds (6mm at knit time, which can be temporarily increased to 12mm while knitting is paused) can be embedded in the fabric by designing a *closed* pocket and inserting the object just before the end of pocket knitting.

Figure 13.12 *The two-layer area of the knit can be fully closed, and items can be embedded inside by inserting them just before the top row of the pocket area. Unlike in fully automated knitting, embedded items can be relatively large and fragile. Here, an LED backlight panel is embedded in a hat.*

13.3.3. Creativity Assistance: Paths of Improvisation

The third module targets open-ended exploration with a greater depth of complexity than the first module. Because of the necessary presence of the knitter, manually-operated machine knitting presents a great opportunity for real-time creativity. However, the effects of particular cam setting choices can take a few rows to become clear, and a beginner may not have much basis for understanding their range of options. With the additional complication that recently-knit rows aren't even visible to the knitter yet, the knitter might not have enough information to make improvisational choices.

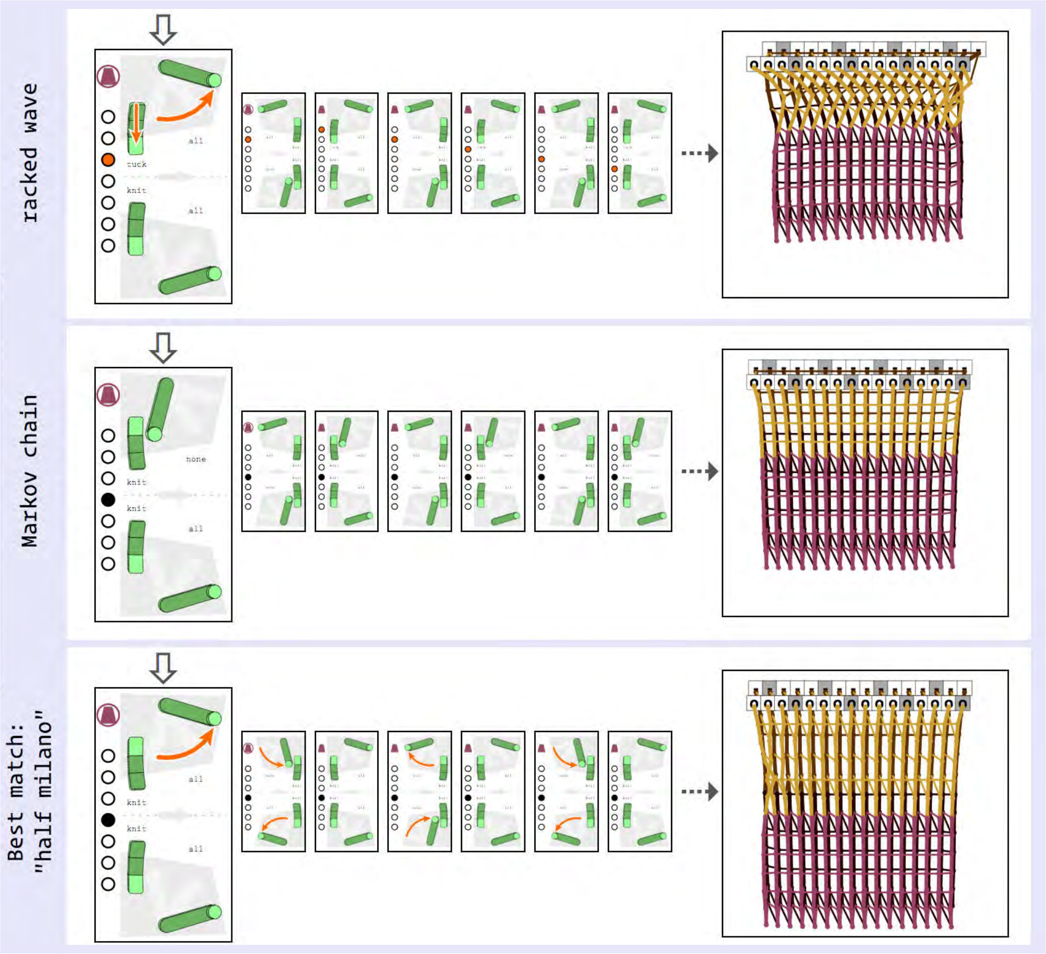


Figure 13.13 *The "Path Options" module shows three possible future outcomes based on different algorithmic tactics. For each path, a sliding window of instructions is shown alongside a preview of what the fabric would look like if that path were followed.*

To show how the knitter's understanding of complex possible outcomes could be supported, I produced an interface module which generates and simulates a set of "path options" for the knitter to consider pursuing. Each path generates its instructions using its own sequence generation algorithm, and it is displayed as a sliding sequence of instruction panels alongside a fabric simulation with the hypothetical stitches that would be generated by that path highlighted in yellow. As with the Pockets interface, instruction panels show a live view of which settings the knitter needs to change to pursue that instruction. As knitting progresses, the set of path options is updated accordingly. Paths whose "next step" corresponded to the action just taken by the knitter are advanced to show the following step; paths which did not include that action are recomputed starting from the new step.

The path options module is written to be flexible and extensible with respect to which generative algorithms are used. I implemented three:

1. A "racked rib" path generator, which proposes either "rib" or "cardigan" cam settings (based on similarity to the current settings) and then modulates the per-row suggested rack position according to a wave function, stepping up and down by one rack position per row to hit the full range of positions.
2. A Markov chain path generator, which derives suggestions based on past rows the knitter has made (with some initial seeding of basic row types).
3. A "best match" path generator, which attempts to match recent knitting sequence to one of a list of named fabric types. This list was derived from a swatchbook assembled by Stoll (a manufacturer of knitting machines), and it includes stitch patterns like full and half cardigan, full and half milano, and tubular knitting.

13.4. Improvisation by Novice Users

To gain insight about how my system could support learning and ultimately a creative practice, I introduced seven new users to the system.

13.4.1. Research questions

I aimed to study 1) basic usability: whether participants could understand the annotations and use them to reason about machine operation; 2) improvisational usability: whether the system sufficiently scaffolded real-time decision-making; and 3) overall participant attitudes toward hand fabrication, computational mediation, and improvisational practice, both in their own work and as they experienced these aspects of the system. The first two questions are assessments of my specific technical system, while the last question relates to the broader possibilities for augmented manual machines and exploratory use of interactive fabrication.

13.4.2. Participants

To avoid biasing the results on basic usability, I recruited participants with no machine knitting background, and no or minimal hand-knitting experience. For safety reasons, and to mitigate novelty effects from interacting with computational creation overall, I required experience in other computational production systems: six had 3D printed and/or laser-cut, and the remaining one has used computational systems for creative image generation. In order to meet these qualifications, and in accordance with covid-related limitations on visitors, I recruited participants within my department, or family members of department members, who were not textiles researchers. The participants ranged in age from 20 to approximately 40.

13.4.3. Procedure

For each session: after asking the user to practice moving the carriage, I introduced the basic "interpretation" view (Figure 13.9) and gave a verbal explanation of the carriage settings. The user was encouraged to interact with the settings and knit as many new rows as they liked until they were “ready to learn another capability,” at which point I introduced the racking lever. The user was given the option to view a swatch of several “named” patterns (rib, tube, cardigan, half-milano, and a mock interlock structure) along with a paper printout of instructions for how to knit each. Finally, I introduced the “suggested paths” view (Figure 13.13) and again encouraged each user to interact for as long as they liked with the system. In all, users spent approximately an hour each interacting with the system. After this, I conducted a semi-structured interview with each user, focusing on their experience of the system, how it compared to past fabrication experiences (both computational and manual), and their creative decision-making throughout their process. While the interviews were semi-structured, I asked each participant at least the following questions:

- Please tell me about past creative fabrication experiences you’ve had, especially either involving textiles or digitally-mediated fabrication?
- How did this experience compare to those?
- Please tell me about what you made.
- Please tell me about creative decisions you made during the fabrication process. (If there was a specific instance that emerged during the “think aloud” portion of the workshop, I reminded the participant of that.)
- Were you able to explore the possibilities you wanted to explore?
- Given more time, what additional things would you like to try?

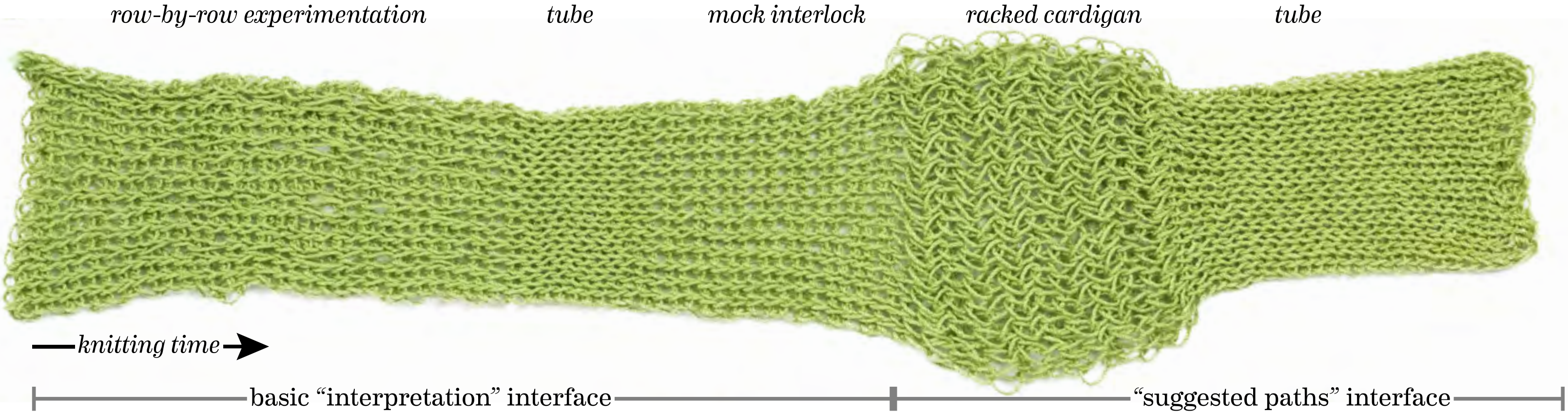


Figure 13.14 *P7’s swatch, showing a progression from row-by-row experiments through named fabric types, including the racked cardigan which requires per-row rack changes.*

13.4.4. Analysis

I recorded the audio from the interviews, photographs of knit artifacts, and time-stamped system logs. The system log data includes all user actions perceived by the system, such as changing a cam or rack setting and moving the carriage. (Note that this data is messy, because it is not debounced e.g. to remove moments when the classification system mis-categorized a cam setting—because categorization is done many times per second and is generally accurate, these only appear as brief flickers to a user, but would be recorded as “changes” in the system.)

To assess basic usability, I viewed the artifacts (Figure 13.14) and system logs. In the artifacts, I starting by looking for egregious knitting errors of the type which my error-checking Subsection 13.2.4 was designed to help avoid. I found that all participants did successfully avoid these errors. While I did not deliberately include a comparison case with error-checking turned off, several of the participants discovered another, comparably predictable common error which I had not built checks for. These participants each encountered the same problem multiple times, suggesting that it was a difficult problem to avoid without tool assistance.

I found that users made many more one-off setting changes within the first part of their knitting, including much more changing of cam/rack settings *without* moving the carriage, to see the effect of these without committing it to the knit. This implies a process of initially gaining literacy with the system.

To assess improvisational usability and participants attitudes about computationally-mediated hand fabrication, I analyzed the interviews. I performed a reflexive thematic analysis [42] by segmenting the interview transcripts and producing first highlights, then initial codes in a spreadsheet, then performing an iterative bottom-up coding. Because my questions largely centered on the experience of fabrication, my analysis is constructionist. I organize my observations of participant experiences and attitudes into themes in the following subsections.

13.5. Scaffolded Learning

The participants were novices to both knitting in general and machine knitting in particular. Most participants described their learning as initially undirected, and they expressed that the system made it possible to manipulate the machine without needing to first form a complete understanding of its operation. Indeed, participants described being able to operate the machine before understanding much at all: “This isn’t something I’d typically do and it’s nice to have something like this where I can just kind of jump in and I am very confused about a lot of things but eventually I will pick it up. With the help of the computer [...] I get a more intuitive sense as to what is happening under the surface as opposed to needing to be explained every little part of what’s happening.” (P6)

Similarly, P3 mentioned an initial period of knitting to get accustomed to the machine, before branching out: “It took me a little bit to get comfortable just going back and forth, but once I started being able to see what was happening, it was like, ‘Oh, I can change stuff up.’”

Depending on their goals, a knitter might find these modules to have too low of a learning ceiling. P2, who was mostly interested in gaining and refining a mental model of how the machine worked, expressed concern that they might not truly be learning and summarized their interaction with the Paths module as “Well, I’m kind of just following instructions.” (I discuss this possible negative outcome in Subsection 13.8.3.)

However, other participants balanced their priorities between gaining a deeper understanding and generating an interesting artifact (in P1’s words: “I don’t really like to feel like I’m making garbage’”). P6 enjoyed the system because “it was nice to see that I could put something together relatively easily and have some sort of guidance [...] and actually make something that looks like it was designed with purpose and intention," implicitly regardless of whether it was their *own* purpose and intention.

13.6. Interaction with Hybrid Processes

Participants touched on feelings of "stress" (P1), "confidence" (P1, P7), "trust" (P1, P2, P6), and "self-reliance" (P6) to describe how they viewed their relationship to the system over the course of their session. P1 described the interpretation assistance as a kind of "re-assurance" and an "encouragement."

In relation to how they thought of fully-automatic systems, they remarked on the relative power and also responsibility of hybrid interactive systems. Despite the usual premise that fully-automated systems aim to be reliable and predictable, every participant with computational fabrication experience mentioned that, when a problem does occasionally arise, the user typically doesn't know until after it occurs. P2 compared using a fully automatic machine to "the handoff that happens [when you] give a plan or geometry to a secondary fabricator and trust that happens." P6 gave a longer explanation that was also suggested by P1, P2, P3, and P7: "Since I'm physically at the system the whole time, working with it in this hybrid approach, it's much easier to avoid any issues that might come up. With a 3d printer, with a lot of automated fabrication, there's a kind of expectation that, well it's automated for a reason; I don't need to necessarily watch too much, within reason. [...]That is not always the case. Even printers that are industry standard sometimes can just have wild things happen to them. Things can go wrong and that is definitely something that is not likely going to happen with this hybrid approach. One, because it's telling me where things might go wrong, and two because I am constantly there at the machine [...] For example if I'm pulling the machine across and I feel all the resistance building up that's a pretty good indication that something is going wrong and I should be careful."

The benefits and drawbacks of interactivity were summarized by P4: "If I just give something to a printer, the output is predictable all the time. But the thing is, if I play with something like this, I have the control. [...] So I have the rights to make a mistake as well [...] If I play something with my hands, putting more effort on it, I feel like I did something really by myself."

13.7. Embodied Knowledge in Manual Machine Processes

In addition to the complex interpretive expertise of understanding the machine's settings and operations, a manual machine knitter must learn the haptic and auditory cues of successful operation. Each participant remarked on gaining this knowledge over the course of the session. For example, from P6: "Knowing how hard to push—I would say it definitely faded back into my subconscious by the end." And from P2: "even if you're following [the guided improvisation module], at the start there is a lot of experiencing the difficulty in the the haptics and understanding what feels right, and not, and the sort of rhythm you get into with switching the gear. Even if you're not thinking about all those switches, you're building that physical memory of the interaction with the machine how everything should feel and sound."

This embodied experience of using the manual machine was generally seen as a positive. In comparison to the fully automatic process of using a 3D printer, P6 said "Assuming in a perfect world that your [3D printer] is going to work well, you can just walk away from it and come back later once it's done. But [having to physically operate the machine] isn't always necessarily a bad thing in my mind. [...] I think there is an aspect to it, sometimes you just really want to zone in on one thing and make sure you're doing that one thing really well."

All participants at some point in their conversation made a full-body "moving the carriage with both arms" motion, and P5 did so with an onomatopoetic "shunk" sound as well. P3 made the gesture while saying "I was having fun with the process once I got more of a handle on it," and later summarized the experience with "there's a lot of satisfaction to it."

The hands-on aspect of the process also prompted feelings of pride, or ownership. P6 was very enthusiastic about the aspect of handcraft in the system: "I think that there is something really really special about being able to make something... I say 'by hand'—I'm putting some giant air quotes around that because it's using the machine—but, you know, something that you crafted yourself."

13.8. Discussion

13.8.1. On-Machine Interaction for Experiential Fabrication

I proposed that on-machine interaction is especially suitable for contexts in which a "hands on" experience is desirable. In the non-automated context of this work, the hands-on labor is not optional; however, this does not necessarily make it less desirable. Participants connected hands-on production to ideas of labor as a locus of value, for example suggesting that they might make nice gifts for loved ones (P2, P6). Additionally, participants, as well as the authors of this work and anecdotally numerous lab visitors, have found the physical sensation of manual machine-knitting delightful. The auditory and haptic cues, along with the smoothly repetitive motion and feelings of control over a complex mechanism, add up to a uniquely satisfying experience.

In building this system, I made several deliberate choices focused on maintaining an on-machine experience. I used the machine itself as the only input—the front-end interfaces could be used with mouse clicks, for debugging or for explaining machine operation to someone without their own machine, but I typically deployed the system without either keyboard or mouse visible. I arranged the computer screen physically very close to the bed, to allow quick glances between the two. The distance could be closed entirely with either projected imagery, or with an Augmented Reality headset.

13.8.2. Augmentation as a Way to Leverage Existing Machines

Participants also mentioned that the system allowed them to find value in a machine that they may not have otherwise interacted with, either because it was intimidating or because, as practitioners of computational fabrication, they found the idea of purely mechanical machines boring. While I do not share this latter opinion and am not of the belief that my system inherently “elevates” the knitting machine, I do see this as evidence that the system broadens access, bringing new attention to a mature and fascinating fabrication machine. Augmentation does not need to destroy or subjugate the underlying machine. I chose entirely reversible hardware interventions, and designed my modular software systems to offer flexible amounts of support.

13.8.3. Overreliance on Computational Guidance

A drawback of computational tools is that they can “water down” or de-skill production processes: if a user is simply enacting system instructions, they lose creative agency. This concern has become particularly topical as increasing use of machine learning techniques in creativity support has spurred a new wave of discourse on the relative roles of creators and computational systems.

Because I view machine augmentation as a possible way to scaffold learning, the idea that a creator could over-rely on a computational system to the detriment of developing their own intuition is concerning. Indeed, one participant mentioned exactly this concern. (See Section 13.5.) While each participant’s engagement with the system was too brief to produce deep expertise, I did observe that participants did not rely uniformly on each computational aspect of the system. The *did* lean heavily on basic usability assistance like error checking, which was explained by their fear of breaking the system (P2, P4, P6, P7). However, they followed higher-level suggestions (in the “Paths” interface) much less strictly. This implies that they were able to view these appropriately as suggestions, which they had more agency to reject.

13.9. Future Work

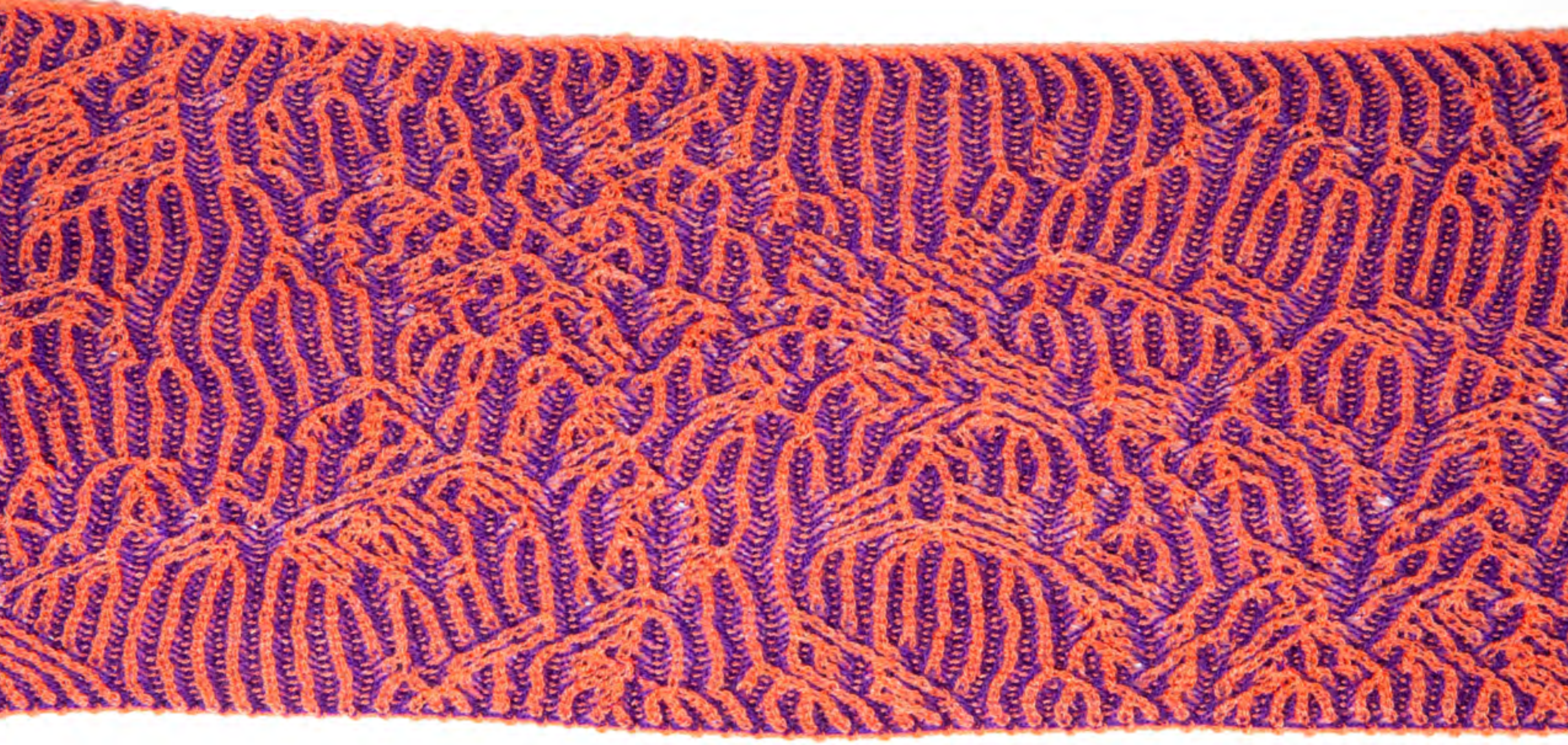
I discussed two research areas this work contributes to: on-machine interfaces and augmenting existing machines. The challenge of doing these simultaneously is that the system must be adaptable to a specific, possibly vintage or otherwise non-normative machine. In the case of the system documented in this work, as I stated in Subsection 13.1.1, the underlying Dubied knitting machine I used is very typical of industrial-style v-bed knitting machines; while some have a different number of needle types and/or a subset of these cam settings, my machine model (Subsection 13.2.2) can be easily configured to these differences. Consumer single-bed machines typically have a different style of needle selection, but my model could be extended to cover this as well. A trickier proposition is adapting the hardware, such as the camera mount which attaches to the mounting hole intended for an auto yarn-changing mechanism—while this is likely to be standard for Dubied machines of a similar era and onward, it is much less likely to be immediately portable to another brand. Similar situations exist in many other manual fabrication machines, such as machine shop tools, kitchen appliances, and sewing machines: while the basic mechanism of a given type of machine are well-established, the specific form of the tool may vary widely. To solve this problem, future work in this area could draw on research in “upcycling” [408] and adaptability [419] to generalize how disparate machines can best be outfitted with various categories of sensing.

I see this concrete technical system, and my discussion of my domain-aware implementation decisions, as a critical step toward broadly accessible real-time fabrication for creativity and education. I additionally hope this work can inspire the digital fabrication community to revisit the vast breadth of not-currently-computational fabrication equipment to support fabrication—whether automated, manual, or novel hybrids—in a wide variety of domains.

13.10. Summary

This chapter described a lightweight approach to equipping an existing mechanical fabrication machine with sensing and visualization to increase the operator's access to understanding the recent past, current, and potential future states of the machine.

My exploration of the domain space was especially driven by the particular **soft structure** patterning capabilities of this kind of machine knitting, including hybrid inclusions such as e-textile systems and complexly textured fabrics, as well as the **malleable context** of personal and reflective fabrication. "Lightweight augmentation" is a ***soft technology***: "augmentation" must be flexible enough to interface with an existing condition, and "lightweight" invites future modification.



14. Finding a Grain in Brioche Knitting

The computationally-augmented manual machine knitting I described in the previous chapter was quite constrained by the affordances of the underlying machine; in particular, the machine I used could only differentiate amongst two types of needles which had to be arranged in advance, and, without heavy manual intervention, could not produce the loop mergings and re-arrangements described in Chapter 4. That machine also had advantages, such as its relatively large loop size, which made intricate patterning more visible even with a single yarn color, and its support for temporarily moving the beds apart to insert other materials. The result of these constraints and advantages is an emergent set of recognizable *styles* of knitting that helped guide my development of both the system infrastructure and the demonstration modules.

A fully-automated industrial knitting machine, like the Jacquard looms I discussed in Chapter 10, has a creative possibility space that is ambifortunately much less constrained. In this chapter, I use the metaphor of a material's *grain* to underpin a specific approach to building tools for such domains.

14.1. Grain

The ability to create expressively in a given medium often involves gaining intuition about that medium's *grain*. By analogy to woodworking, in which the term refers to the anisotropic arrangements of fibers in wood, "grain" describes the cascading tendencies and opportunities emerging from intrinsic material properties throughout a fabrication process. Cutting or carving a piece of lumber "with the grain" requires different techniques than working "against" it, and the visual characteristics of woodgrain often influence the design of an overall project.

While grain arises from the low-level physical characteristics of a material, creators often manipulate it as an abstraction. For example, watercolor painting is rooted in a complex blend of rheology, pigment dispersal, and absorption dynamics, but a skilled painter may tacitly understand these in terms of effects like wet-on-wet color mingling [27]. Similarly, textile designers may refer to the "hand" of a fabric in determining its suitability for an application–a "crisp" fabric might pleat well, or a "clingy" one may conform to curves–as a subjective assessment incorporating flexural rigidity, friction, stiffness and softness [144, 414]. To summarize: **a medium's grain comprises the tendencies and opportunities, which emerge from its aggregate low-level properties, but which are conceptualized abstractly by skilled creators.**

In other words, an expert can pursue high-level goals by using mid-level composite abstractions to assess and manipulate low-level material effects. Working with such abstractions might be thought of as "artisanal intuition." (Indeed, subtle grain effects may be perceived as synonymous with "hand craft," as they are often discarded for simplicity in industrial production [310].) Hobbyist creators may look to kits and tutorials to explore unfamiliar media; professional creators may be guided by existing experts in a formal or informal apprenticeship [226]. However, it is less clear how to support finding a grain in a less-established medium. In HCI, the rise of digital fabrication has supported a wave of material inventiveness by managing complexities that would be unworkable in fully analog processes [54, 194, 381]. Low-level details such as cutting or extrusion speeds can have aggregate effects at the scale of an entire object. For example, deliberately over-extruding material in a 3D printing process can result in filligree-like curls of filament [204] or extended petal-like loops [296]; under-extrusion might produce a flexible, porous surface [95] or tunable micro-fibers [282]. These effects can be faithfully orchestrated by digital systems, which enable both the precision needed for thousands of repetitions and the flexibility for one-offs.

14.2. Grain Spaces

Unfortunately, high-level tools such as slicers for 3D printing typically optimize for conformity–aiming to replicate an implicitly grainless digital representation as accurately as possible–or fabrication-time efficiency, and thus often diminish or obfuscate the range of unique material possibilities. Creators who wish to interrogate this range for technical or expressive purposes must often work directly in a low level, such as raw or lightly-parameterized G-code. When these systems solely parameterize aspects of the machine process (e.g. temperature or feed rate), the relationship between these parameters and the eventual material output can be difficult to understand. Users of these tools have few opportunities to explore and build their intuitions, and the range of possibilities within even a simple digital fabrication process can be under-constrained and difficult to make sense of.

I propose an approach to building tools for expressive material intuition via a *grain space*. I define a grain space as a **specified set of material affordances, encapsulated as a high-level manipulable notation, alongside a way to compile from this notation into low-level fabrication steps**: a "way to think about" possible outcomes within the medium, coupled with a "way to do it." As a kind of a *style* of production—a set of associated aesthetic guidelines and boundaries—a grain space does not enable every possible material outcome from the broader fabrication method; rather, it delineates an area for exploration. Once curated and defined, a grain space forms a basis for reasoning about the design and implementation of tools for manipulating material effects.

14.2.1. Chapter Overview

This chapter demonstrates the grain space approach through the design and implementation of technical system for machine-knitting in a style known as "brioche." In machine knitting, the "low level" is a precise three-dimensional arrangement of yarn loops resulting from loop-by-loop instructions for a computational knitting machine; within this broad domain of possibility, the higher-level style of brioche knitting produces a a two-color fabric with a springy feel and an all-over visual texture of branching and merging. I chose brioche as an exemplar grain space because it can support complexly emergent outcomes with a simple yet evocative structural grammar (described in Subsection 14.3.1), and because it is an material which is not well-represented in simple mesh or pixel grid notations (see Section 12.1 and Figure 14.1).

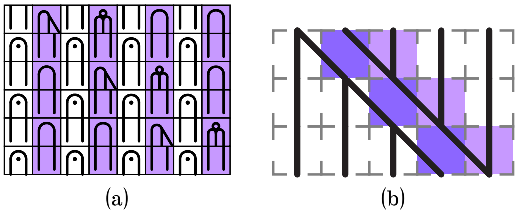

(a) (b)

__Figure 14.1__ a) Typical hand-knitting chart for two-color brioche. b) My notation for the same swatch as in (a).

I used my defined grain space as a design impetus to generate varying conceptualizations of brioche knitting—as a field of directional switches, as vector gradients, and as flow lines—and encapsulated these in a suite of exploratory creativity tools which are situated in the physical world to encourage immediate engagement and the potential for unique or messy inputs: doodling, curating, or composing nearby real-world objects as a way of interacting with the design space.

This chapter describes how I defined and encoded a brioche grain space into a modular processing framework with a knitting-specific computational backend, visualization and manipulation capabilities for my brioche data interchange format, and example input modalities. Finally, I reflect on the role of such tools in creative practice through observation and conversation with users of two of my tools.

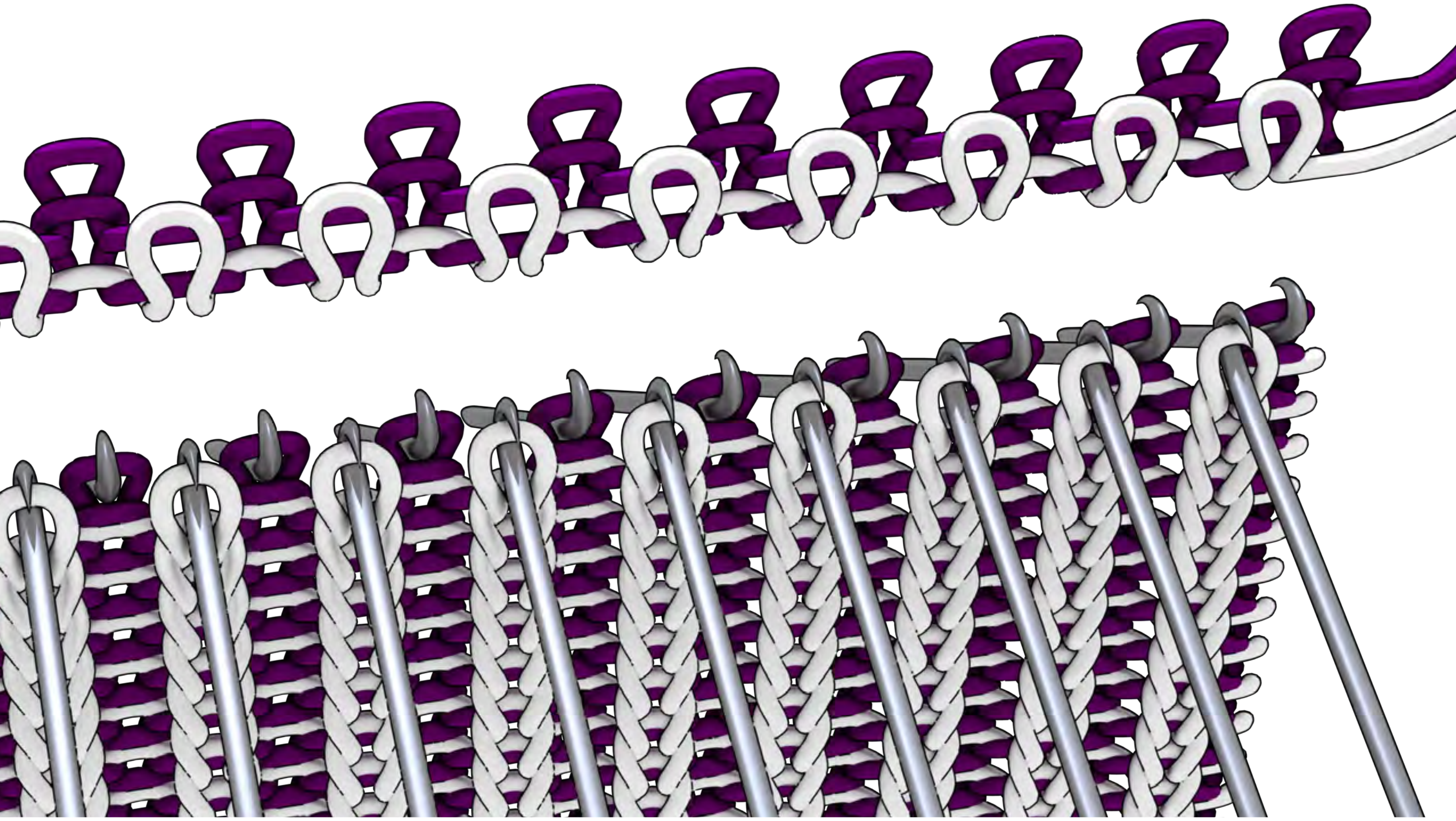


Figure 14.2 *The structure of two-color brioche knitting as formed on a v-bed knitting machine. Each of the two yarns forms the knit loops of one face of the fabric and joins with tucks to the other face; in this case, the back face is shown in purple yarn.*

14.3. Brioche Knitting

In addition to referring to a delicious egg-enriched bread bun, "brioche" is a hand-knitting term for what machine-knitters would call a "full cardigan" loop structure [227]. (I use the "brioche" term in this document to avoid confusion with the garment called a "cardigan," and because the name is charmingly evocative of the fluffy softness of the structure.)

The basic brioche structure consists of two conjoined faces of fabric. As shown in Figure 14.2, machine-knit brioche can be formed on a a two-bed ("v-bed") weft knitting machine with each face on its own bed. The yarn passes alternate between these two faces, knitting on this pass's primary face and tucking on the other. Because each yarn zig-zags between the beds, the distance between stitches is further and each face is somewhat loose and fluffy, giving an overall lofty hand to the fabric. Many knitters choose to knit the two faces in contrasting yarn colors ("two-color brioche") [228], resulting in faces with a clear "foreground" and "background" color each.

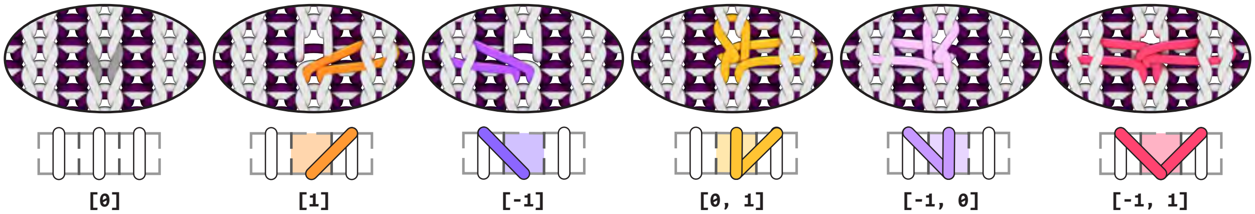


Figure 14.3 *Six elementary "brioche operations."*

14.3.1. A Grammar of Brioche Knitting

With the basic brioche structure as a basis, stitches in the foreground can be shifted, merged, split, and transposed. In this work, I consider six atomic "brioche operations," shown in Figure 14.3 above:

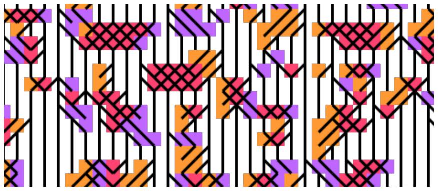
1. a default stitch (`[0]`), which goes straight up (is consumed in the same column as it was knit)
2. a "rightward" stitch (`[1]`), which merges to the right
3. a "leftward" stitch (`[-1]`), which merges to the left
4. a "split rightward" stitch (`[0,1]`), which splits in two; one stitch goes straight up and the other goes to the right
5. a "split leftward" stitch (`[-1,0]`), which splits in two; one stitch goes to the left and the other goes straight up
6. a "split both ways" stitch (`[-1,1]`), which splits in two; one stitch goes to the left and the other goes to the right

In aggregate, this simplified stitch vocabulary can give rise to many complex visual outcomes: the "grain" of brioche knitting. In two-color brioche, these manipulations are visually emphasized as a distinct figure and ground, such as in the "leafy" patterns that are popular amongst hand-knitters [229]: at positions where gaps are produced in the front face, the back face is exposed, creating both a change in visible color and in the physical feel of the material.

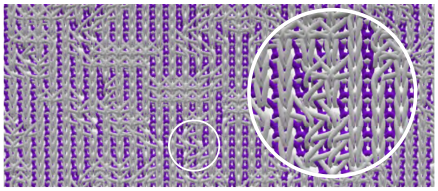
14.3.2. Representation

My representation encodes "brioche knitting" in two dimensions, which I support with automated compilation to sequential machine operations. Within my system, brioche format is represented in code as a 2D array of my stitch types, and it is visually represented in one of two ways:

1. Diagram, with simple lines standing in for loop directions. Stitches other than the default (`[0]`) may optionally be highlighted with colors indicating their direction and split, as I have done throughout this chapter.



1. Loop view, in which the fabric is represented as a 3D model which can be rotated and zoomed. This model does not have any physics-based simulation applied; however, it shows the color contrast effects of displacing front-bed loops.



14.4. System and Physical Inputs

As an exemplar of a tool for exploring a grain space, my system transforms easy-to-use input media extracted from the designer's physical context, including found snapshots and tactile manipulation, into instructions for fabrication on a knitting machine. The grain space of brioche bridges between the user's manipulations and the low-level machine instruction outputs, and provides inspiration for specific input modalities.

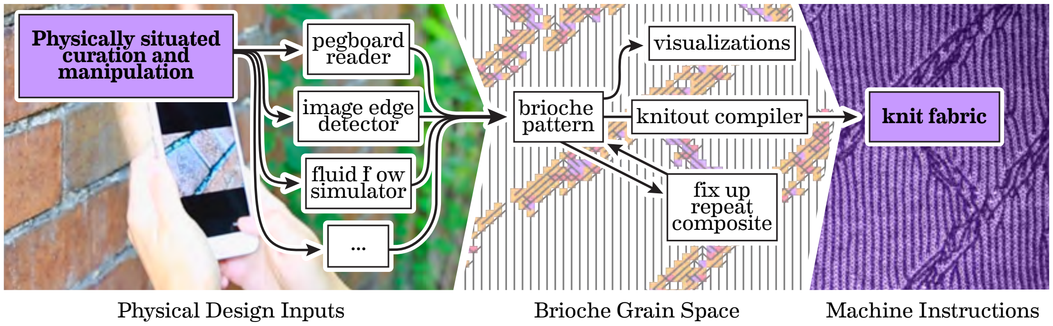


Figure 14.4 *Methods of viewing and manipulating brioche patterns (visualizing, compositing, and compiling) form a grain space that mediates between broad physical inputs and specific fabric output.*

I implemented this system as a set of interoperable modules, Figure 14.4:

1. input modalities which translate physically-situated inputs into my brioche format. Out of a vast space of possibilities, I created three input modules (described in detail in the following sections) to show a range of possibilities for immediate, impromptu, or experimental texture manipulation.
2. tools for viewing and manipulating a brioche structure. These include a simple visualizer which displays the resulting loop structure, either as a 3D model or as an abstracted diagram, as well as utilities for repeating a pattern, joining it it with other patterns, compositing it with short-row shaping, and applying image filter-like effects
3. a compiler from brioche format to Knitout code [232], which directly represents the low-level instructions for driving an industrial knitting machine

To best support physically situated creativity, each of the above is written in client-side JavaScript, enabling them to run straightforwardly in the browser on mobile devices. For input, I use the JavaScript (Emscripten) version of OpenCV [40] with either the device camera or stored images. The 3D visualizer uses ThreeJS [250]. A backend server, also written in JavaScript (Node.js) links these modules by collecting, storing, and transmitting brioche-format data over websockets.

All swatches in this chapter were knit on our Shima Seiki SWG091N2 15g knitting machine at half gauge [233] using Tamm Petit, a 2/30 Nm acrylic yarn.

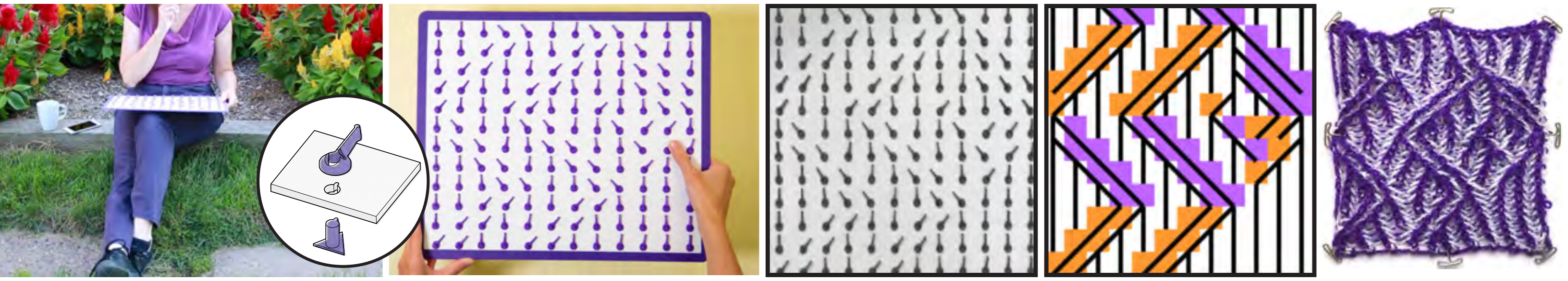


Figure 14.5 *A physical brioche "instrument" allows hands-on pattern exploration at the stitch level. a) The system's image processing runs in-browser on a mobile phone, allowing it to be quite portable. b) Each "peg" is 3D printed in two pieces, which snap in to laser-cut holes in the board. The dials have three detents—center, left, and right—enforced by printed-in compliant leaf springs. c) The overall board. d) Computer vision perspective rectification. e) The derived brioche pattern. f) The result.*

14.4.1. Tangible Instrument

To support relatively fine-grained manipulation within the brioche grain space, I constructed a physical "brioche instrument" representing twelve rows and twelve columns of brioche knitting, Figure 14.5.

I borrow the term "instrument" in this context from Kreminksy [184], building on Wardrip-Fruin [401], who use this term to refer to systems which offer a "noodling around" experience within a computational design space. "Noodling" is a form of early-stage material exploration [52] in either physical or digital worlds [401]; an instrument supports this experience by being less score-oriented than a game, more directed than a toy, and by contributing its own "voice," or, in the language of fabrication, its own grain.

In my brioche instrument, each grid operation is represented by a directional pointer knob with a haptic detent for each of its three valid positions, indicating the three single-loop (no split) operations. The pegboard therefore allows direct manipulation of the smallest "atom" of my design space, while abstracting the sub-atomic details such as the bilayer structure of the knit and the necessary machine-level instructions required to produce the represented knitting.

The pegboard input device is inexpensive and portable. The pointers and shafts were printed on a low-end filament deposition printer, and the base board was laser-cut to accept them. I use computer vision to read the board's state by: detecting the corners of the board; applying perspective rectification; and, for each knob, comparing the average pixel brightness in each of the three locations the knob could be in, Figure 14.5. Because this method uses relative brightness and the detents in the knobs provided a low number of possible positions, I found this simple method robust.

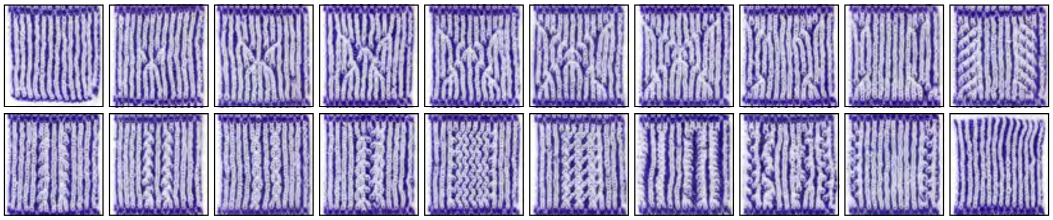


Figure 14.6 *Twenty panels of a "knit animation" designed with the instrument. These were knit as a continuous filmstrip scarf.*

I used this system to design a short "filmstrip" intended to be photographed and displayed as an animation, Figure 14.6.

14.4.2. Photographic Snapshots

To support an impromptu bricolage-like approach, I built an image processing pipeline to automatically generate a texture "suggested" by the input image, Figure 14.8. One goal of this work was to push beyond "pixel art" representations of knitting, which do not fully capture the characteristics of many styles of knitting beyond "colorwork." I observe that the distinctive visual element of my brioche grain space is the diagonal edges formed from stitches leaning into neighboring columns, so I focused on image processing options which highlight these.

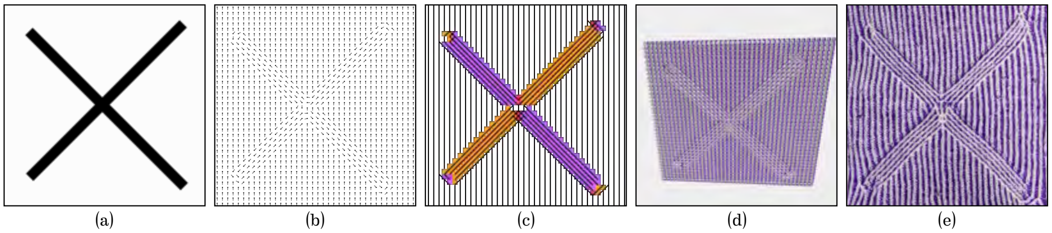


Figure 14.7 *A pipeline for generating brioche patterns with images as input. a) An input image. b) Gradients derived via Sobel operator. c) Brioche pattern. d) "Loop view" visualization. e) The resulting knit fabric.*

I created a pipeline which offers the following processing steps (Figure 14.7), including several which may be toggled or modified to modify the output:

1. (Optional) Apply a Gaussian blur
2. (Optional) Apply Canny edge detection [48]
3. (Optional) Isolate straighter edges with a probabilistic Hough line transform [181]
4. Detect image gradients using Sobel operator [358]
5. Downsample the matrix of gradients to the desired swatch dimensions (in loops)
6. Bucket the gradient directions into leftward-leaning, rightward-leaning, vertical, or horizontal (represented by a both-ways split)
7. (Optional) Apply replacement rules for modifying knittability or aesthetics, as described in Section 14.5

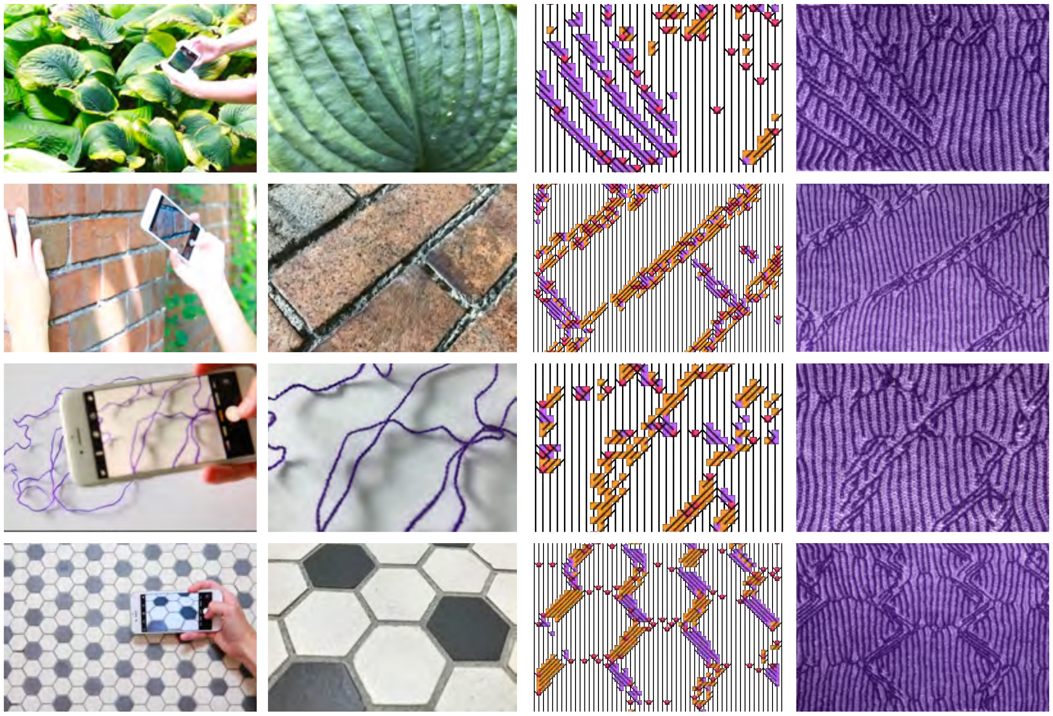


Figure 14.8 *I used a mobile phone to collect images in my homes and outdoors for photo-inspired texture swatches.*

I found that, by supporting simple, mobile image collection, possible texture elements could initially be captured without specific regard to the eventual knitted output, as high-level exploratory inputs. By immediately converting these to a diagrammatic or simplified loop view, the system allows its users to develop their own taste of what "works" as an interactive process of curation, or bricolage.

Choose File lamp.png

Blur? ☑

blur amount

Find Edges? ☑

lower threshold

upper threshold

Swatch Size

Include split stitches at boundaries? ☐

Figure 14.9 *A screenshot of the "Snapshots" interface on a mobile tablet.*

14.4.3. Fluid Simulation

The line-dominant character of brioche suggested a third input module which begins with the same high-level form of found or curated image input, but which performs a further computational manipulation on it–in this case an interactive 2D lattice-Boltzmann fluid simulation [342] using the contours of the image (extracted with OpenCV) as solid barriers, Figure 14.10.

The designer can stir the simulated fluid and choose when to pause the simulation. A tablet provides ample screen space to see and interact with the simulation, while retaining the mobility needed for a physically situated interface.

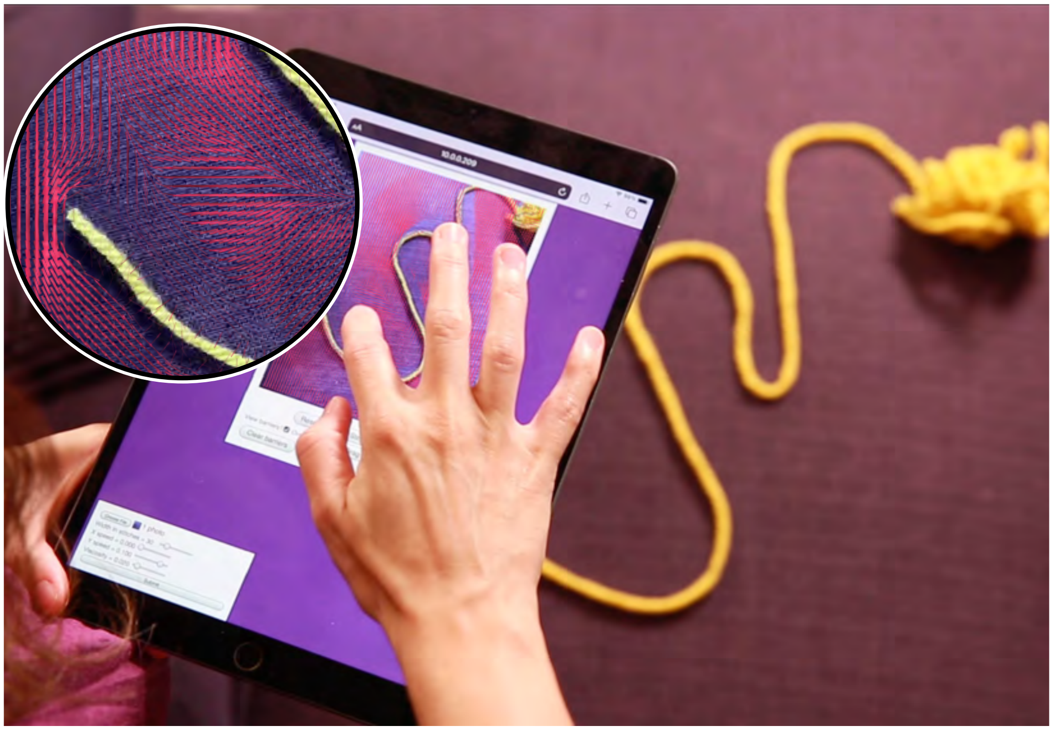


Figure 14.10 *The simulated fluid can be "stirred" interactively as it interacts with the edges in the image.*

While this module is superficially similar to the previous one in accepting found images as high-level input and ultimately deriving the brioche pattern from a vector field, I found several key differences in the design spaces afforded by each:

Compared to the straightforward image gradient pipeline, which highlighted allover texture and amplified within-figure tonal variation, the fluid simulation primarily operates on visually distinct outlines, encouraging figure/ground "massing." I found that simpler or more abstract inputs, such as the yarn and paper cut-out above, made the fluid simulation overlay clearer to understand and correspondingly more enjoyable to manipulate.

This difference in input has a corresponding effect on the interaction experience: where the "Snapshots" interface encouraged a collection and curation approach, the fluid simulation interface rewarded intervention and creating specific compositions.

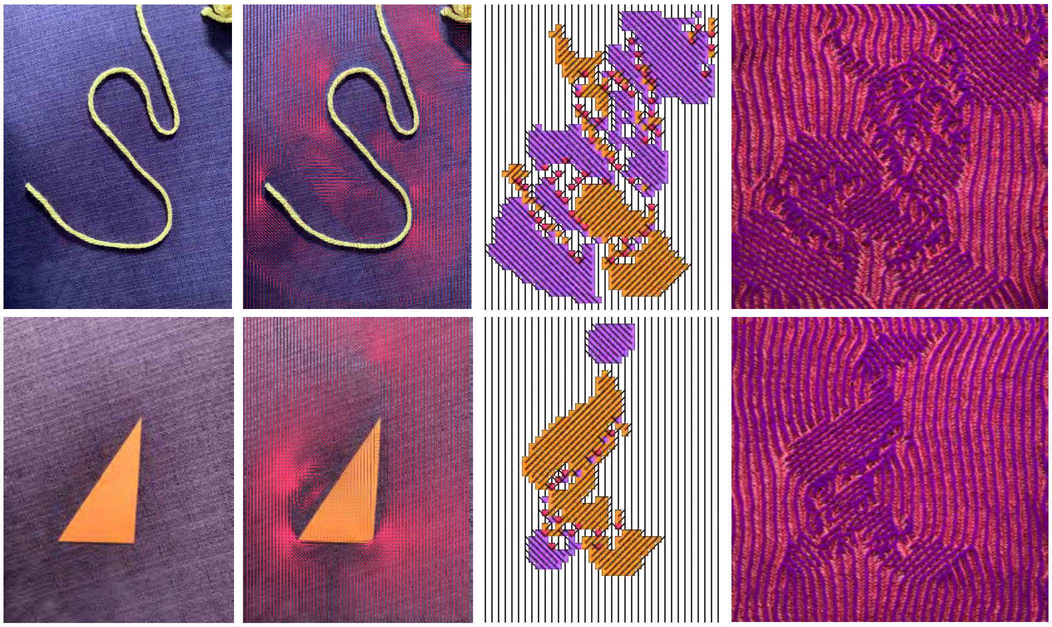


Figure 14.11 *Two knit swatches generated with my fluid simulation interface, shown with the input image, fluid flow lines, and resulting brioche pattern.*

As an input device, this module is literally chaotic–because the simulation is interactive, it's difficult to get the same swatch twice even with the same image and parameters. I consider this derivation of a knitting pattern from an abstract physical simulation a provocative demonstration of computation as a flexibly creative medium.

14.5. Manipulation in Brioche Space

The medium-level brioche format supports simple manipulations such as joining, compositing, and performing procedural transformations for aesthetics or knittability.

14.5.1. Joining and Compositing

I found my brioche format highly suitable for array-level manipulations such as joining patterns (as in the filmstrip shown in Subsection 14.4.1), repeating a pattern length- or width-wise, and overlaying a pattern onto a simple shaping template. For the last, I expanded the brioche vocabulary to include an `["x"]` operator, which represents a grid cell which is skipped in this row. This allows the use of "short-row shaping," which can produce non-flat knit sheets [15]. As shown in Figure 14.12, I produced a simple hat requiring just one seam by compositing a short-row "template" with the output from my fluid simulator, then applying a vertical repeat to the output. In this case, "composition" could be defined quite simply: each grid cell in the output is a copy of the corresponding cell in the design, except where an `["x"]` in the template overrides it. In practice, I found that knittability was improved if patterns avoided merging a stitch "into" the skipped area, so any stitches whose lean direction collided with a skipped cell were modified as well.

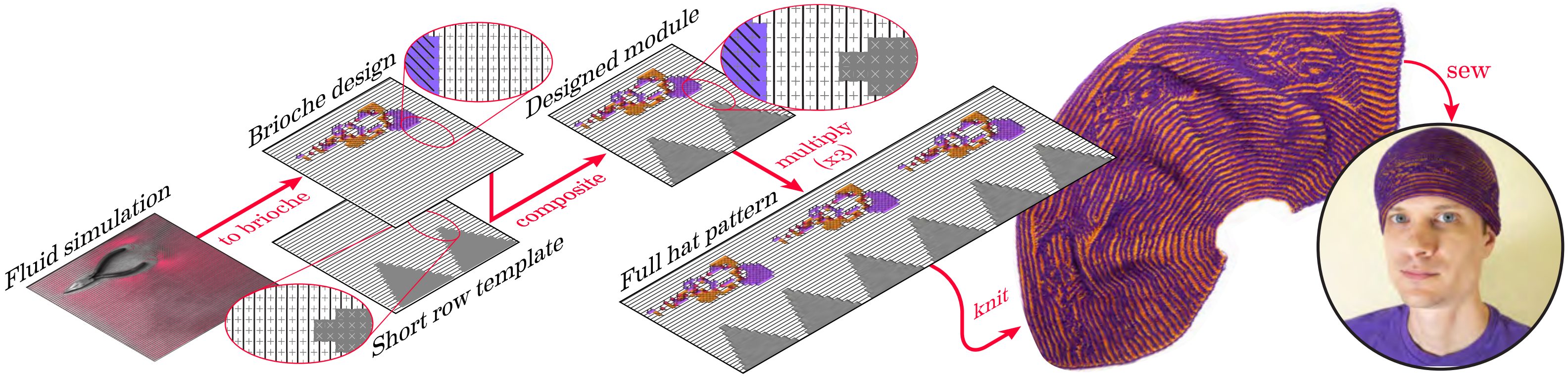


Figure 14.12 *A brioche pattern can be composited with another. Here, a pattern derived from the fluid simulation interface is overlaid onto a template which provides overall shaping. Because the final knit is a curved surface, it can easily be sewn into a hat.*

14.5.2. Perturbations and Filters

Just as the fluid simulation perturbs the input data in uniquely computational ways, the brioche format itself can be altered in ways that are reminiscent of image filters, while respecting the affordances of the brioche medium.

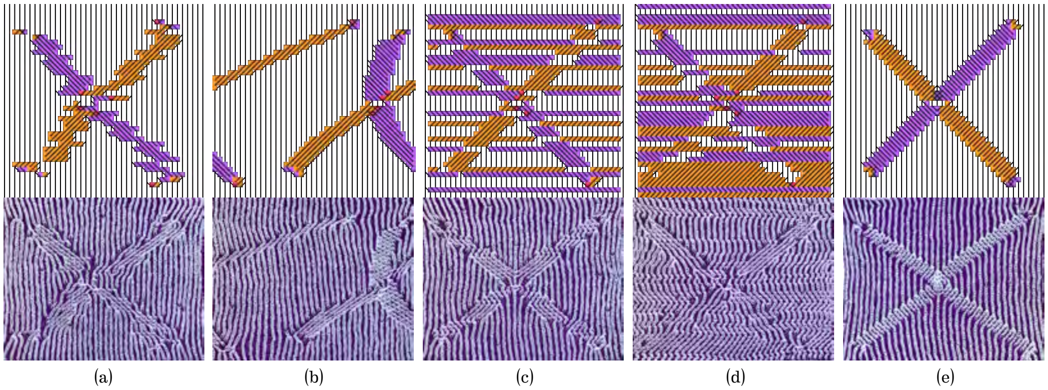


Figure 14.13 *"Glitch"-like manipulations of brioche patterns.*

For example, as shown in Figure 14.13(a), rows of the brioche pattern might be slid left or right with a parametric frequency and amplitude, similar to how a "scan line" filter might distort the pixels in an image; in (b), the lines are offset by a parameterized sine wave.

To go beyond pixel-like manipulations and include the unique nature of brioche, this sliding might additionally "skew" each stitch, (c): to "skew rightward," `[-1]` might become `[0]`, and `[0]` might become `[1]`. When increasingly large regions are skewed in this way, (d), the result verges on a shift between figure and ground, unique to brioche textures. Another naturally "brioche" filter to apply is inversion: rightward- and leftward-lean are swapped, (e).

All of these style-respecting filters are inspired by the data structure of the brioche format itself, and the simple logic operations that can be performed on it.

As in Subsection 14.5.1, “Joining and Compositing,” such logics might also include custom types of composition such as adding or subtracting different stitch types.

14.5.3. Replacement Rules

An optional component in my system can apply authored operation replacement rules to act directly on the intermediate brioche representation, similar to regular expressions. Such rules can support various improvements in the final knit results; I implemented one each for aesthetic and knittability robustness purposes.

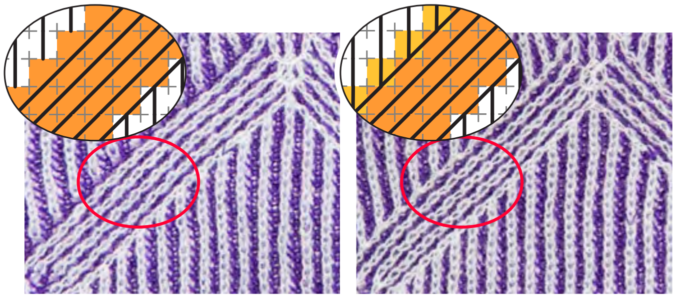


Figure 14.14 *Replacing* `[1]` *stitches with* `[0,1]` *at the boundary between a leaning area and a non-leaning area, for aesthetic effect.*

The first, Figure 14.14, allows the designer to choose to add the split versions of leaning loops at the boundaries between leaning and non-leaning loops. This reduces the directional asymmetry between merging vs. splitting columns.

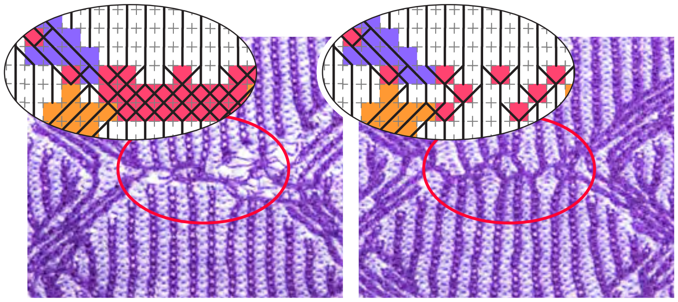


Figure 14.15 *Removing some* `[1,-1]` *stitches in an area where there are many of them, to improve robustness of the knit result.*

The second, Figure 14.15, replaces patches of stitches which are known to be fragile to knit. For example, two-way split stitches (type **`[-1,1]`** in my grammar) can put extra strain on the yarn of the loops in the split, which, depending on the tear strength of the yarn, can potentially cause a yarn break. This effect is compounded with several contiguous splits. My replacement rules break up these contiguous patches for more reliable knitting across a range of yarn types.

14.6. Grain Spaces for Novice Creators: User Study

To study how my system can support open-ended and early-stage creativity, I conducted a study with six individual participants.

In each session, the participant was shown swatches of brioche knitting and introduced to the Snapshots and Fluid Simulation interfaces, then instructed to use either interface for as long as they liked with the goal of ultimately choosing a single swatch design to knit and keep. Participants were told to "submit" (upload to the server) any interesting results as they generated them. When the participant was satisfied with their result (which took between twenty minutes and an hour), I held a semi-structured discussion with them. In each discussion, I opened by asking the participant to describe their submitted results, any memorable moments from their interactions with the tools, and which pattern they would like to knit. Then, while their pattern was being knit, I transitioned to a broader discussion of their relationship to design and creativity tools in their own analog and digital practice.

As a prerequisite to the study, all of the participants had some experience with designing patterns in a textile handcraft, including embroidery and weaving, Table 14.1. Four participants specifically had some experience with knitting, with one (P4) being a fairly advanced hand-knitter with experience hand-knitting brioche patterns (but not with designing their own brioche patterns). I included this selection criterion to study brioche pattern design as *proximally* unfamiliar (as opposed to wildly so), and to allow closer analogies to each participant's own creative practice in the open discussion portion of each session.

p	domain experience
P1	professional sewist, hobbyist cross-stitch embroiderer
P2	professional designer, hobbyist mixed media
P3	previously professional designer, hobbyist crocheter
P4	intermediate-advanced knitter, weaver, embroiderer, quilter
P5	professional photographer, hobbyist freeform embroiderer/quilter
P6	expert weaver/spinner, hobbyist knitter

Table 14.1 *Participants*

Participants mostly used their own mobile devices, with one exception preferring to borrow a tablet to use the Fluid Simulation interface, and another borrowing a phone because of low battery on their own. Several participants were therefore able to use images they had taken prior to their session. One participant additionally chose to use some images downloaded from the Internet during their session.

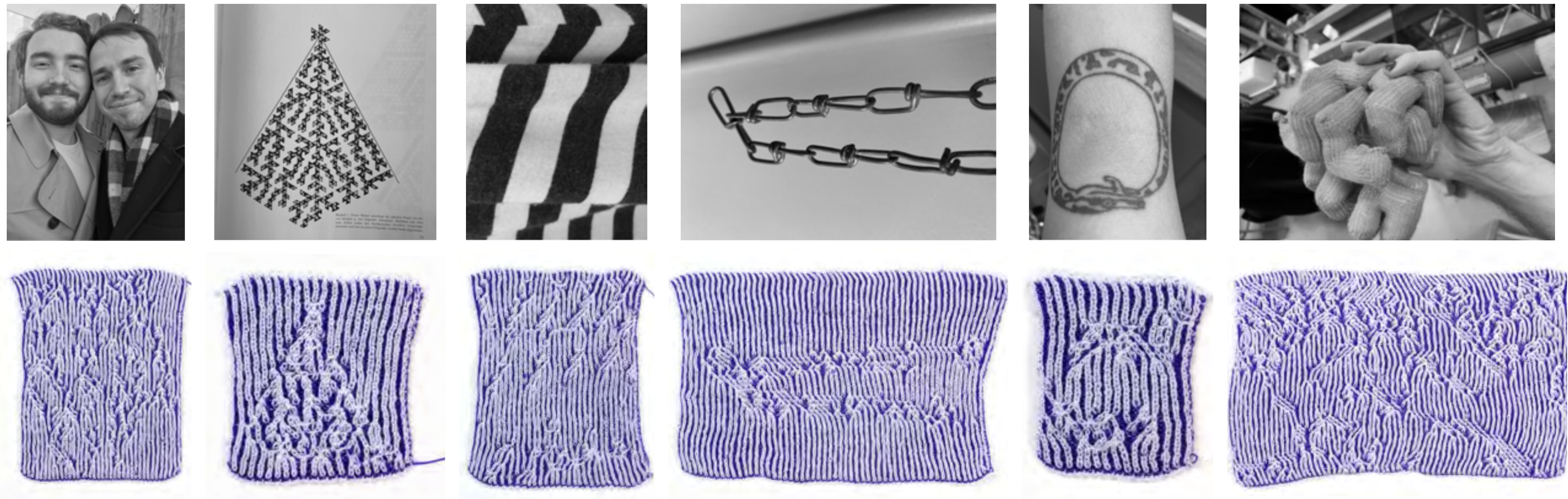


Figure 14.16 *Resulting knits from each participant.*

14.6.1. Control, Tool Collaboration, and Pushing Bounds

I position a grain space as something to be explored, not something which will necessarily directly enact a pre-decided outcome. Indeed, a creator will likely have a difficult time with either the Snapshots or Fluid Simulation tool if they have a very specific outcome in mind; because these both provide a limited set of specific mappings from image gradients to pattern output, they might be described as "opinionated" tools, or ones which do not offer the user a high degree of control.

I noticed a range of participant reactions to this exploratory rather than controlled mode of creativity. One participant, P3, began their session with a highly specific vision of a desired result, which may not have been possible within the bounds of brioche knitting; they tried the widest range of tactics to steer the system, including submitting images downloaded from the internet and taking pictures of whiteboard doodles. In discussing their work, they contrasted this experience with the "one to one matching" that they had come to expect, in their hobby crochet practice, between the photo provided with a pre-designed crochet pattern and that pattern's output. P3 described their session with the brioche tools as an arc from frustration, through compromise, to eventually "coming to meld with the material." P3 positioned creativity support tools in general as something to "fight"; however, when asked about digital tools that they enjoyed using, they mentioned highly-constrained tools such as social media image filters and Canva[49], an in-browser editor that emphasizes pre-designed templates. Because of their background doing communication design work, P3 felt overfamiliar with low-level graphic design (choosing fonts and color schemes "was a lot of work and I just don't want to go through that again"), so they appreciated that, in Canva, "the harder decisions have already been made. [...] Thank you, Canva!" In comparison, P3's underfamiliarity with the brioche pattern space meant that they didn't have a basis for what to expect, or whether the tools were "working."

Conversely, several participants cited experiential connections to how they would deliberately cede control in their own typical creative practice. This ranged from an overarching discussion of the roles of agency and collaboration in tool-use to P2's specific principled rejection of fully-controlled processes in their professional creative practice: "If I plan something—if I have something in my head and I just execute it—it's usually not that good. [...] I don't find it that interesting because the process is quite linear and there's no surprise [... I]t's often even boring." For P4, "working with this definitely feels like I'm collaborating with the software. Like I'm picking things but it's also making decisions for me." P5 used a similar metaphor of collaboration in describing much of their own practice, saying that "for most things I do I fall on the end of 'I kind of know vaguely which way I'm going but I let the tool have a big say in where I end up.''[...] I feel like I'm still the one making the decision but I want I want to know the boundaries of where the tool ends up putting me." To begin to understand these boundaries within the brioche system, P5 performed several bound-testing experiments: first, "when you give me a bunch of sliders [...] I just push everything to one side and I push everything to the other side"; then, "how closely can I make the thing look like the thing?" and "how far away can I get when like the final product is going to be stitching and the original thing was also stitching?" These exploratory tests were a common pattern across participants: "How organic can I make this?" (P1), "seeing both how obscured I can make it but also how almost-true-to-form I can make it as well" (P1), "I was just interested [to] see how granular of a structure, or what's the visual details you can translate" (P2), "Can I translate even something like typography into that system?" (P2), "It makes me want to draw a bunch of knot-work and then try to photograph it and translate it" (P4), "[Maybe] if you take a picture of a knitted object and put it in computer vision, something cool will happen." (P6).

Several participants specifically tried to make "bad" or uncanny results (e.g. "That is the simple thought: I want to see how terrible this will be," P1) and compared the process to a glitch practice [238] with desirable instabilities ("From translating from digital to analog there's always some loss in this process, or some translation error or whatever, which I find really inspiring," P2) These comments show the participants seeking the edges of the brioche patterning space, as an active part of understanding the overall possibilities in conversation with the system.

14.6.2. Personal Involvement and Ownership

Because I define grain as something which is ultimately understood tacitly, each artisan's understanding of grain becomes becomes personal. I was interested in participants' perception of their personal involvement in the creation process, their feelings of ownership over the results, and how these relate to the control and collaboration themes in the previous subsection.

Several participants chose imagery with personal meaning. For example, P6 used the tools as an excuse to re-examine a familiar location: "I've known this building since 2009, like very intimately, and I haven't looked at the brick the way I look at it now." Indeed, in several cases, this imagery was chosen with a desire for it to *stay* personal—that the transformation into brioche space could encode meaningful secrets. When P1 tried an image of themself with their partner, they verbally acknowledged that they expected it to be almost entirely illegible;

this ended up being their selected knit (Figure 14.16). P5 chose an image of one of their tattoos (Figure 14.16) and processed at a very small swatch size to be especially abstract: “There’s a connection but only I really know it.”

Others mentioned that the process itself imbued some personal meaning. P5 contrasted the curational process of choosing and selectively processing photographic inputs with a less hands-on process (“if you had generated eight thousand completely random brioches and said ‘go through these and pick your favorite’”) but quickly clarified that they saw this distinction as private to the practitioner: “there’s a difference for the person making it, but there’s very little difference for anybody else.” Similarly, when P2 discussed their preference for surprise and undercontrolled processes (mentioned in Subsection 14.6.1), they acknowledged that the difference might be entirely their own internal perception: when things are too predictable, “maybe others find it good, but I don’t.”

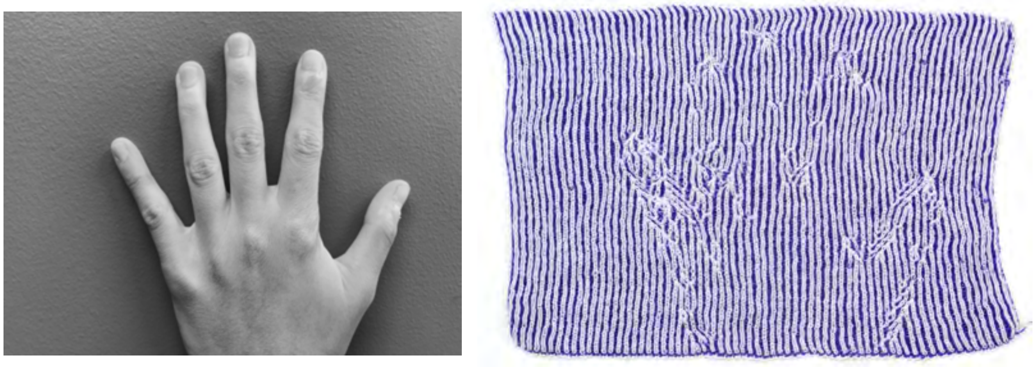


Figure 14.17 *P5’s “handprint” pattern was inspired by the feel of the fabric swatches they handled.*

14.6.3. Blank Pages and Curation

Because I wanted the participants to focus on experimenting with a gestalt mapping from photo to pattern, I did not provide them with my tools for editing in the brioche space. Several participants mentioned this, touching on desires for “the ability to just sort of remove parts of it” (P5), to “just come in here and put these things around and and manually fix these little details” (P2), and to “edit these patterns now, like refine them.”(P2) “But,” as P2 immediately followed up, “at least it brings you to to a state where you don’t start from scratch on a blank.”

This highlights a strength of the system as an *early* stage in a creative process. P2 highly valued avoiding “the blank page,” and described various tactics from their own design practice, including “found footage” and manipulating sketches from previous projects: “I just create options over options over options—use something I did as a starting point for something else, and iterate over and over, and then at the end you have a large selection” from which to curate the best outcomes. P2 found a similar opportunity in the brioche tools: “And then it’s about selecting those moments that you like which is something I also like.” P5 contrasted the curational process with a blank canvas, saying that the Snapshots interface was “different than if you have a blank canvas [...] I’m very much going around looking at objects in the world, or patterns around me.”

P5 mentioned what they see as a negative aspect of many digital processes, that often there is “no effective cost in twiddling with things forever,” which “changes how you make decisions, because it’s like you don’t have to think about resource consumption, except for your time and energy, which is a resource that people don’t think about when they’re doing things digitally.” As such, they saw the lack of fine-grained editing within the Snapshots tool as a positive constraint. P4 mentioned a similar tendency toward perfectionism in users of digital tools, and said that in their own practice, they prefer to use the digital tools as a jumping-off point for hand work (e.g. using generative design tools that are intended for machine embroidery, but doing the embroidery by hand instead).

14.6.4. Textile Materiality

Because a grain space encapsulates both a style and a physical material, I was interested in my participants’ perceptions of computational brioche knitting as both a computer-mediated process and as a tactile material—in particular, whether the material specifics of knit fabric supported, or even affected, the creative practice.

While the participants mostly had no experience with brioche knitting (excepting P4), they incorporated their associations to adjacent material domains. P5, who has a longstanding photographic practice, referred to the grain of photography as an important part of how they considered their input: “there’s an object or there’s a thing or there’s some sort of whatever that I’ve put into a photograph, which is already using a tool; [...] the camera is the tool that takes [it] in reality and translates it into a thing that then I submit into the tool that you gave me, and then with that there’s several more sliders.”

The mobile phone/tablet screen is the most immediate surface of my tools, but participants remained aware of the material properties of the eventual knit output, and some incorporated it into their conceptual exploration. Inspired by the springiness of brioche knitting and in reference to pinscreen toys, P5 made one swatch based on their handprint, Figure 14.17: “the samples are so pleasing to touch, so having one where it’s it is literally just a handprint and you can put your hand on it, touch the handprint, so that was sort of playing with the the touching sensation [...] And it’s the opposite of the pin toy because it’s so soft.” P3, P4, P5, and P6 each submitted imagery which itself referred to textiles; P6 described this choice as a kind of “magical thinking.” P5 explained a composition using an image of the heavy hand-embroidery on their jeans as their “meta submission,” explaining that it was “a pure exercise in just deconstructing a thing and then making a thing. [...] How far away can I get when like the final product is going to be stitching and the original thing was also stitching?” Additionally, P4’s prior experience with brioche knitting led them to try some foliage motifs, and to discuss the possibilities of repeating or tiling patterns.

In these examples, inspiration from the material itself becomes a kind of helpful conceptual constraint. This illustrates how a grain space can encompass associations and inspirations that influence not just what is *possible,* but what might be desirable, or meaningful.

14.7. Future Work

From a technical knitting perspective, brioche is a specific grain space of knit texture manipulation. It is deliberately constrained–even within traditional brioche knitting, I might have supported different loop stacking orders, splitting a stitch more than once, or manipulating both the front and back faces of the fabric. However, without a clear underlying logic, such a system can quickly become unwieldy. A broader space would necessarily include a principled consideration of how to expressively compose various families of texturing techniques, including re-integrating texture with overall shaping techniques beyond short-row templates. Further effort in automating compositional concerns might allow freer exploration in the grain space. For example, my knittability operations use a simple find-and-replace mechanism. As the underlying knit structure becomes more complex, knittability-aware optimization techniques could lessen the burden of defining new texture interactions.

More broadly, I see applications of this work beyond knitting to expressive fabrication anywhere a unique digital/physical grain might be found: for example, in the effects of varying pressure and lead hardness on a plotter-drawn pencil drawing [388], in varying extrusion rates for alternative material characteristics in filament deposition 3D printing [95], or in pushing the bounds of printed overhangs in clay [351]. Computational fabrication tools are often built on re-used abstractions that turn out to be an inelegant fit for a particular fabrication domain. By identifying an approach grounded in both material practice and computational abstraction, I expect this work to inspire HCI tool-builders to craft more deliberate abstractions and interactions for creative computational fabrication across material domains.

As digital fabrication research continues to invent and refine a broad range of material practices, it becomes increasingly important to support not just predefined goals, but to greet new creators who may not even have such goals yet. By offering curated inroads to digital material exploration, we can cultivate a flourishing landscape of creativity in computational fabrication.

14.8. Summary

This chapter described how a grain space—a manipulable notation paired with a fabrication compiler—can support exploratory expressive creativity by bridging from high-level tangible tools to complex fabricated output. The grain space provides both a technical scaffold that ensures knittability and enables quick manipulation through composability, as well as design implications such as notational systems.

Grain is a **tendency**, not a rigid constraint; a computational space is inherently **malleable**. A grain space is a hybrid of these: a ***soft technology*** suited to underpinning modular manipulation systems which are experimental and experiential, drawing on physicality as a source of productive imprecision, as well as computation as source of emergent transformation.

IV.

Structuring Softness

15. Structuring Softness

In this chapter, I systematize *aspects* of soft computational fabrication systems and the *values* and *technical underpinnings* that enable them. This analysis is based on recurring themes in the systems discussed as well as related work.

In drawing together qualities like portability, composability, and parametricity under the heading of softness, my intention isn't to reinvent any individual quality but to show that they have a kinship; that where one of them is applicable, others might be too.

Just as how, as I described in Part I, the complex but pliable structure of a textile is what gives rise to its strengths, I intend that this structured analysis of softness to be a tool for designing and building future systems. "Aspects" can be used to understand where softness is currently found, or could be added, in a system; "Approaches" are high-level properties of systems which span different aspects; at the level of actually creating a new soft system, "Values" and "Techniques" are tactics for designing and implementing, respectively.

This part is summarized in Figure 15.1, which diagrams relationships between aspects of softness, soft technical approaches, underlying design and engineering tactics, and other concepts from throughout this dissertation.

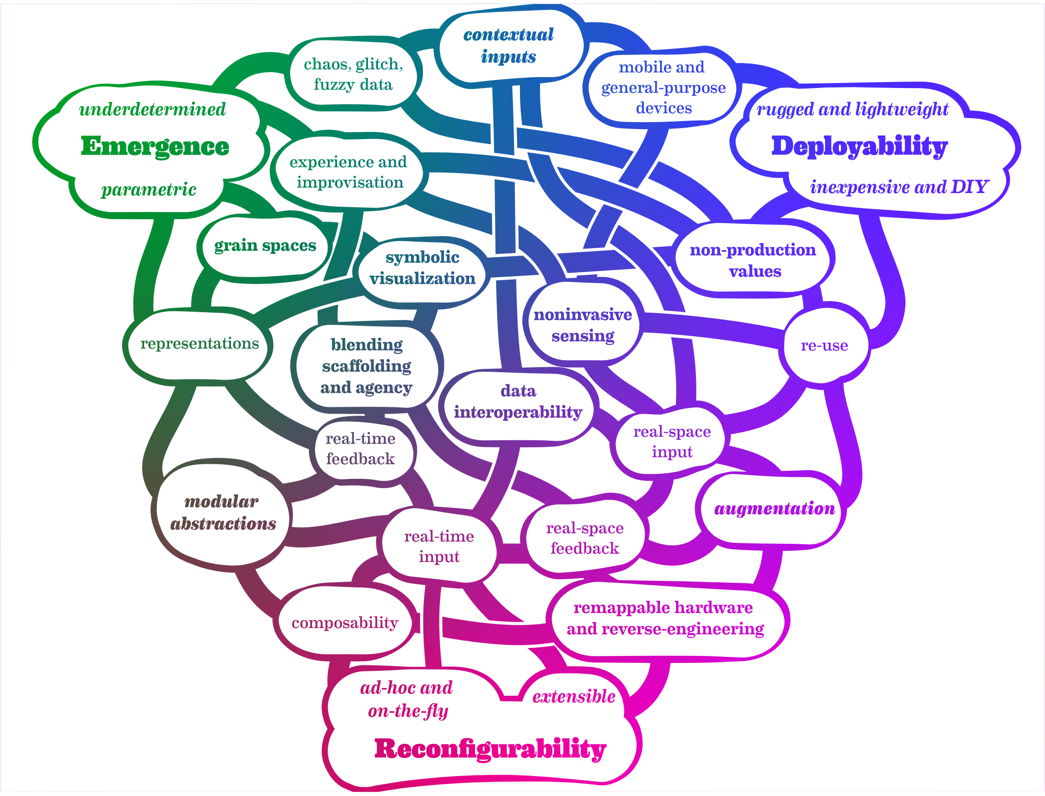


Figure 15.1 *Aspects of, and approaches to, softness.*

15.1. Aspects of Softness

In this section, I discuss three overlapping *aspects* of softness—what it means to say a technology is “soft.” These aspects have particular conceptual loci within and around the system and its output: in other words, they are generalized answers to the question “what part of this system can be considered soft?” Of course, these aspects are rarely found in perfect isolation; I describe their overlaps as productive *approaches* in the next subsection.

15.1.1. Emergent

The softness of an *emergent* system is located in its output. This might involve the use of deliberate underdetermination to channel a chaotic or fuzzy input, as in the Twitch and self-portrait sketches in Chapter 10, or it may be a more structured emergence, such as the metamaterial characteristics in Part I or the “improvisation paths” in Subsection 13.3.3. It’s worth noting that emergence is a perceptual phenomenon: a complex system can be technically fully deterministic but still unpredictable. I discuss ways of reasoning about emergent phenomena particularly from a materials perspective in Chapter 6 and from the dual perspectives of procedural generation and weaving design in Chapter 10.

15.1.2. Configurable

The softness of a *configurable* system is in the system itself. As opposed to mechnically simpler loom types, a Jacquard loom is distinguished by its on-the-fly configurability—a new pattern can be woven simply by changing a data input to the loom, without having to re-thread any heddles. The personal Jacquard loom in Chapter 9 was further configurable with optional camera-based sensing and networking capabilities. The TC2 Jacquard loom was not necessarily designed to be extended in this way, but reverse-engineering its control protocol enabled it too to be modified. Configurability particularly has implications for real-time and real-space interactions, where the system might be adjusted on the fly or as-needed for an unusual input. Reconfigurable systems might involve modular and composable components, or affordances for aspects of the system to be swapped or overwritten on the fly, such as how the various editing and compositing tools in the Brioche-generating system (Chapter 14) could be chained together flexibly.

15.1.3. Deployable

The softness of a *deployable* system is in the context surrounding it. This includes physical portability, as in the Personal Jacquard Loom (Chapter 9) and the “in the field” mobile interfaces for brioche pattern design (Subsection 14.4.1); more metaphorically, it includes bringing a system into a different social or cultural context, such as in the “Twitch Plays Loom” sketch (Section 10.6). Deployable technologies could be physically lightweight and/or difficult to break, or it could be relatively inexpensive to support new, casual, and independent users. It might incorporate existing infrastructure to accomplish these goals, for example by running on mobile phones or by repurposing vintage hardware.

15.2. Approaches to Softness

The intersections of the above three loci imply a complementary set of fulcra to develop softness within a system.

15.2.1. Augmentation

Combining configurability and deployability, augmentation involves improving existing capabilities or adding new ones. Hobbyists often improve their tools, for example by upgrading the filament reel on a 3D printer [257]. *Computational* augmentation can include adding capabilities that computers are particularly good at, like tracking and modifying data and displaying extra information. The possibility of augmentation unlocks resources that might otherwise be overlooked. Especially in fabrication, there is a wealth of historical technologies that are highly sophisticated (and are often more robust than contemporary technologies) but may be dismissed as obsolete. In this work I particularly prioritized augmentation that was reversible without damaging the underlying system, to promote reusability, reduce waste, and encourage creative risk without necessarily incurring *financial* risk.

15.2.2. Contextual inputs

Combining deployability and emergence, contextual inputs are those that don't come from the system or the user, but from the surrounding context. In this dissertation, I particularly discussed *physical* inputs, which are both a high-bandwidth input medium and can introduce noise, mess, and imperfections to act as creative seeds. The potential for unpredictability of physical inputs has a long history of supporting creativity [356]. In the brioche work, locating the input on a mobile phone encourages users to use the camera of the phone and therefore their own physical surroundings as inputs. In Chapter 10, two of the "underdetermined" weaving systems included physical inputs: the actual-sized paper "Blobs" that could be nudged, scattered, or torn, and the weaver's own body. To different degrees, these demonstrate the use of imprecise processes as a source of emergent patterning. Physical input is also a natural paradigm for real-space interaction, as shown in the double-cloth example of the personal Jacquard loom (Subsection 9.3.1).

Other contextual inputs are possible as well. The social context is an important input into the the "Twitch Plays Loom" sketch in Section 10.6, where the site-specific social norms of Twitch encouraged an emergent semi-structured chaos from the fluctuating audience of guests making joint creative choices amongst themselves.

15.2.3. Modular abstractions

Combining emergence and configurability, modular abstractions are computational constructs which are designed to be interoperable across a range of representations from the material object to the fabrication operations. In the knit materials work in Part I, the knitting operations were defined on row-by-row bases, parameterized by factors such as whether a drawstring or a spacer filler row should be present.

15.3. Soft Values

The "soft values" in this section are provided as prompts to inspire the design of soft systems, particular in early-stage ideation.

15.3.1. Non-production priorities

Many computational fabrication systems prioritize "production"-oriented concerns like efficiency, automation, and precision. These are certainly worthwhile goals in many contexts, but not all. Indeed, full automation may be actually detrimental to a system, if collaboration or learning are goals. Precision can be de-emphasized if speed is a priority [131, 255], and artifact-centred metrics of efficiency may not be appropriate for experiential systems.

In Chapter 9, I showed how de-prioritizing weaving speed and precision made it possible to greatly improve portability, price, and hands-on interactivity. In Section 14.6, creative practitioners discussed how imprecision and unpredictability could be inspirational, and in Chapter 13, participants discussed productive frustration and labor as a source of value, especially in the contexts of learning and of fabricating gifts for loved ones.

Other possible priorities include:

- accessibility: something that is bigger, slower, and/or quieter provides more cognitive tractability and better access to interpretive technologies such as audio explanation for blind and low-vision practitioners [37], as well as more pleasant for practitioners with sensory sensitivities
- safety: slower speeds, lower mechanical complexity, and manual operation can all make fabrication machines safer for an interactive user, e.g. the manual Dubied knitting machine is designed for hands-on operation whereas the industrial Shima Seiki one prohibits it
- ease of repair: ruggedness, lower mechanical complexity and interoperability with common tools and components give a system longer-term longevity and could encourage creative risk-taking
- fun, education, and other experiential aspects: systems which incorporative narrative and/or gameplay [13, 368], or which are simply surprising and joyous can provoke further exploration
- beauty: systems which produce beautiful output, or which are themselves beautiful, for myriad contextual definitions of "beauty," can increase joy in operation and pride and ownership over the results.

Additionally, even the basic "production" priorities may be slightly modified in other contexts. For example, instead of overall "time to finish" speed, a personal practitioner may care more about the ability to fit a creative process into their own schedule, e.g. by pausing and re-starting a project as time allows. "Efficiency" may be dynamically defined in a lab context, in which some materials, such as those that are being hand-synthesized, are appreciably more resource-constrained than others.

15.3.2. Blending Scaffolding and Agency

Soft systems are often experiential, hybrid, or interactive, and they therefore must handle the design of creator agency during the creative process.

Throughout this work, and particular in the practitioner discussions in Part III, I discuss aspects of systems which either:

- *scaffold* an experience: make it accessible to a creator by minimizing complexity, abstracting details, and handling the difficult parts; in other words, they introduce some kind of automation, or
- seek to provide extra *agency* by allowing a creator to intervene at critical moments, personalize the system's configuration, and choose personally meaningful outcomes; in other words, they can be overriden.

While at first glance these aspects may seem fundamentally at odds—there's not exactly any agency over an automated process—they are not zero-sum. Selective automation can amplify creative choices. Additionally, automation of sensemaking can help a creator understand what their choices even are. Some participants who tried the augmented manual knitting machine felt powerless, especially at the beginning, when they didn't feel that they had enough information to make any kind of actual decision; with more scaffolding, they were able to move more confidently and felt more ownership. At the other extreme, though, simply automating everything might be condescending or bland.

Any new soft or hybrid system should carefully consider *which* parts of the creative process are the locus of automation. For example, a system like the augmented knitting machine can automate the production of *possibilities,* while leaving the specific outcomes up to the creator. A system like the Jacquard self-portrait provided the very transparent automation affordance of doing video processing and row-tracking, without any input into the subject matter. The brioche knitting system automates all of the low-level machine control needed for viable knitting, and allows the creator to concentrate on mid-level effects.

Other fruitful loci of automation might include:

- duplicating a creator's work. For example when they want to "undo" a step that is physically permanent, a system can recreate the work up to the just before that step; alternatively, a creator may wish to re-create the artifact in batch or at a different scale.
- setup, finishing, and re-setting. Many processes have setup or cleanup overhead that many creators would be perfectly happy to automate; more generally, it is helpful to understand which parts of the process a creator considers integral to the act of creation versus incidental.
- dexterity-centric tasks. The gap between what is straightforward to describe and what is easy to physically do can be a welcome place for automation. For example, computationally-defined jigs or endstops [379] can be chosen by the creator to augment their own abilities.

15.3.3. Grain Spaces

While I specifically defined the term “grain space” in Chapter 14, the general approach of defining a style notation and encapsulating it in a parametric space can be seen throughout this dissertation.

In Chapter 6, I described a functional style of parametric weft-knit spacer fabrics and correlated its material characteristics to fabrication parameters. In Chapter 10, my method of compiling greyscale imagery to Jacquard weaving is based on “shaded” weave structures using the “principle of inclusion” [338]; this approach not only maintains viable weavability but also, depending on the family of weave structure used, can emphasize the “weavingness” of the output—the shaded satins used in the self-portrait (Section 10.5) are quite similar to various dithering algorithms, but the pointed twill structures in the blob composer (Section 10.4) evoke blankets woven on handlooms. In Chapter 9, the double-cloth example uses a variant of this approach, tuned specifically to double-cloth weaving. The collaborative weaving editor (Subsection 9.3.2) uses this approach to underpin a correct-by-construction “brush” editor which can only paint in idiomatically compiled ways. In Chapter 13, I used an intermediate format which could be thought of as “racked rib Knitout”—that is, a format which encompasses the needle layout and cam settings of the machine and which can be losslessly translated to Knitout, but which more easily enables reasoning about the unique pattern space of the manual knitting machine.

Unlike the more general “design language” or “pattern language” [66], a grain space is specifically about material fabrication, it’s ideally a relatively continuous parametric space allowing clear paths to adjacent outcomes, and it’s *small*: a grain space can, and should, be “opinionated” relative to the overall fabrication technology it is situated within. Defining a grain space is a creative act which balances low-level material technique with context-specific goals. Similar to doing an exploratory thematic analysis or contextual inquiry, developing a grain space can be an inspirational and educational introduction to working with a new computational fabrication medium, and it can help map out the necessary technical underpinnings for a new system.

New grain spaces could be inspired by historical and contemporary craft styles, as in the brioche work, or as in work by other researchers on paper piecing for quilts [357], complex wood joinery [194], or slab-based ceramics [140]. A trip to a museum or library, or perusal of contemporary practice on social media [388], can turn up nearly endless grain-space-sized creative material practices. They could equally be developed from functional material goals, as in the spacer fabric work, or as in work by other researchers on cellular metamaterials [155] and underextrusion effects in 3D printing [95]; potential inspiration might be found in materials science and other material-centric disciplines like electrical and compliant mechanical engineering.

15.4. Soft Techniques

The “soft techniques” in this section are provided to help guide the actual implementation of soft systems. These are concrete hardware and software components that support the higher-level goals of softness, and they’re the ones that I found myself returning to repeatedly.

15.4.1. Lightweight sensing

Sensing that is “lightweight” makes use of existing ubiquitous or inexpensive hardware, and aims to be minimally invasive and/or reversible. These are keys aspect of deployability as well as configurability. Sensors that can be removed or upgraded without damaging the underlying system was a critical part of my approach to augmentation, which aims to strictly add to the underlying machine without overriding or interrupting its core functionality.

In practice, lightweight sensing can include some hardware-based sensing, such as the use of the hall effect sensors and accelerometer in Subsection 13.2.1; these are low-cost and straightforward to work with. However, I found that the most flexible and generalizable approach to lightweight sensing was camera-based sensing using either classic computer vision algorithms or image classification models.

At the simplest end, I used perspective correction and color threshholding from OpenCV [40] to generate weaving patterns from real-space on-warp inputs in the two Jacquard projects (Part II). I used classic computer vision, particularly perspective rectification, segmentation, and brightness clustering to sense slider positions in the personal Jacquard loom (Subsection 9.2.4) and knob positions for brioche instrument (Subsection 14.4.1). These were well-suited to the task because the input was structured by the physical mechanisms, which I was able to design to have high color contrast and regular rectilinear positions. For the messier “in the wild” task of determining the carriage settings on the manual knitting machine (Subsection 13.2.1), I used image classification, which was able to pick up on nuances of shadow and light to determine even the position of the knit/tuck rocker switch.

These techniques do not necessarily require cutting-edge computing equipment for interactive results. They can be reasonably implemented with typical, low- to mid-tier streaming cameras—indeed, several of these were carried out during the early part of the Covid pandemic, when it would have been difficult to obtain a high-end streaming camera—and, particularly in the image classification, much work has been put into the maintainers of OpenCV and Tensorflow to enable operability under computing resource constraints. The desktop computer I used for the augmented manual knitting machine was approximately a decade old at the time, and the mobile phone interfaces in Chapter 14 were usable on various participant phones, including my own which was approximately six years old at the time.

While camera-based sensing may be less accurate than task-specific hardware, I found that the results were quite usable, especially for the real-tine interactive tasks like the augmented knitting machine, where it was possible to re-sample and average the results over a rolling time window.

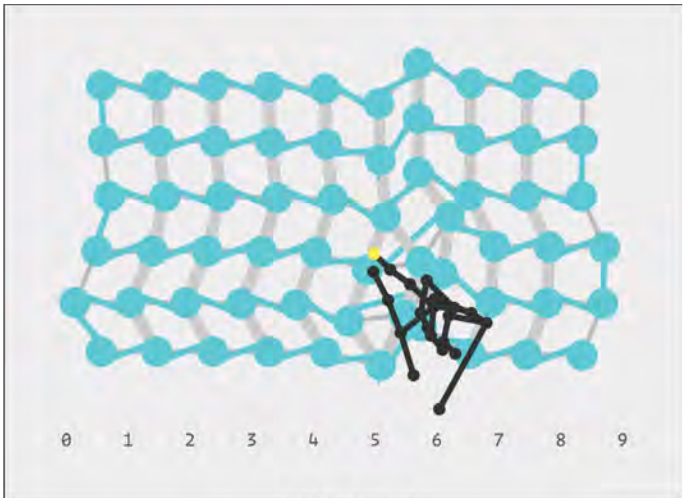


Figure 15.2 *A "pinch" pose, tracked in real time, can manipulate an on-screen representation of knit stitches.*

For even more general classification-based sensing, transfer learning is very promising. Figure 15.2 shows how a hand pose classifier built with Google's MediaPipe framework [218] can be trained to support manipulation of on-the-fly configurable interfaces based on tracking the hand itself instead of the interface. The "tapping" pose in this example was trained with 700 images, collected in <5 minutes with a Processing sketch similar to the one in Subsection 13.2.1, with no effort to investigate a minimum viable number of input images. (The documentation claims that approximately 100 samples per class is sufficient.) In addition to using transfer learning for the specific poses, the pose classifier does not act directly on the raw image data; it acts on hand "skeleton" landmarks identified by a network trained on a Google-large dataset, and is therefore robust to differences in lighting, background, and specific hand geometry and theoretically color (though this last one is a known weakness of [245]).

I am certainly not the first to propose cameras as a kind of ultimately flexible sensor for HCI-related tasks [51, 191]. That said, cameras can have serious implications for user privacy [39], especially in the "personal" and "educational" contexts that I believe soft technologies are particularly suited for. In the work in this dissertation, cameras are primarily pointed at the machines themselves, typically with narrow fields of view cropped to specific areas of interest, and image data is not logged. (The exceptions are the self-portrait and Twitch examples in Chapter 10; in these, I am both the subject captured by the camera and the custodian of the system's data, though I do also cede the data stream to Twitch as a necessary consequence of situating the work in that context.) Every project in this dissertation handled the vision tasks as separable input modules. For the augmented knitting machine, the output of the image classifier was a stream of "current settings" data comparable to the serial data stream from the Arduino that was handling the carriage and rack positions. After training the classifier, the entire camera-sensing process could have been handled by a fully separate computer such as a Raspberry Pi. For the mobile web interfaces in Chapter 14, the computer vision was done entirely client-side, without requiring image data to be sent to central server. (In Section 14.6, I retained some image data as part of user study analysis, but was separate from the image processing and I only received image data that participants specifically chose to “submit,” in accordance with our instructions to them as regulated by our IRB.) Structuring the input in separate, modular, and local-first processes makes it more privacy-preserving, as well as more scalable (because it avoids load on a central server), and easier to adapt or adjust later if more efficient or accurate processing techniques become available.

15.4.2. Symbolic visualization

Visualization that is “symbolic” does not pretend to be a photorealistic preview of an output. In addition to being computationally expensive and often highly task-specific (and therefore not particularly deployable or configurable), highly simulationist visualization can be a worse user experience. A gorgeous, fiber-level rendering of a knit surface is a delight and a marvel, but just like a real knit object, it is likely to be visually confusing. (Even as a practitioner with approximately a decade of experience in machine knitting, I need to concentrate to “read” many knit textures.) Additionally, it implies a level of accuracy to the output that may not be true, especially if the input materials are variable, such as with a color-varying yarn.

Designers have written of the power of a partial and under-defined “sketch,” as opposed to a hardlined “drawing,” to retain semantic access to adjacent possibilities, and to focus attention on the salient aspects of the sketch while minimizing distractions [108]. A sketch can itself be an explanatory and exploratory tool.

For all of these reasons, the visual notations I developed in this work are predominantly symbolic. While I did develop an additional more-literal “loop view” visualization for the brioche notation (Section 14.4) which was intended to show the shifting two-color effects of the brioche style, in practice this visualization was much less useful than the greatly simplified symbol view with its optional directional color highlighting.

Some symbol notations can be easily rendered by composing simple raster tile sets or SVG symbols, as I do in the brioche “diagram” view. Where physics-based material effects are more influential in the output, it may be helpful to indicate these effects in the notation, beyond composing tiles in a grid. The hand-knitting chart tool Stitch Maps [43] indicates the non-rectilinear outcomes of lace knitting by distorting the grid of the chart itself. In the augmented manual knitting machine’s preview, I implemented what I describe as a “physics-inspired” symbolic notation: stitches are represented as abstract nodes, but I wrote a simple mass-spring simulation that shows the emergent ripples and layers that are the major patterning moves in that system. I implemented the physics in vanilla JavaScript, though similar results could be had with actual physics engines, especially games-focused ones like Box2D [53], which prioritize interactive speeds over physical accuracy. Throughout this dissertation, I have used SVG.JS [89] for rendering, as the SVG format makes it simple to add event handling (e.g. by adding a click/drag handler directly to each node of a stitch mesh representation; while mouse interaction wasn’t part of the main workflow for the augmented manual knitting machine, it was certainly helpful for debugging), multiple layers (e.g. front- vs back-bed stitches), and dynamic graphical styles (e.g. re-coloring stitches to reflect a different yarn choice).

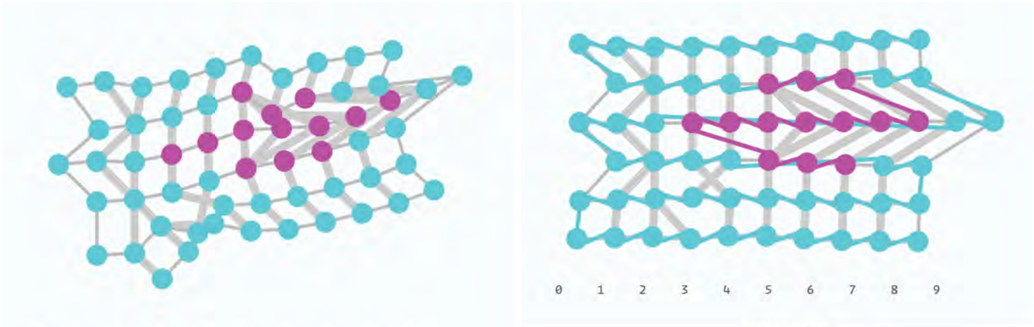


Figure 15.3 *Left: a "simulated" view of a knit stitch graph, showing distortions to the grid from loop connections. Right: a "scheduled" view of the same knit graph, with needle positions, loop formation order, and continuous yarn paths assigned.*

Symbolic representations are particularly able to adapt to various parts of a processing pipeline. In Figure 15.3, I show how parts of a visualization can be emphasized or de-emphasized for different parts of a creative process.

15.4.3. Data interoperability

In discussing contextual inputs and configurability, data interoperability is a primary technical concern. The systems in Part II and Part III all included multiple input/output modalities, including webcams, microcontroller data, web connections, and of course fabrication machines. One of the primary technical tasks of these systems is to manipulate, compose, and translate across these modalities; one of the central aspects of a grain space is that it specifies an interchange format and defines manipulation affordances.

Wherever possible, components of a soft system should be modular and flexibly composable. On a practical level, this has largely meant ensuring that data is serializable, particularly into a JSON format (though sometimes into simpler plain-ASCII representations for initial experiments). I typically use Node.js [279] and its websocket, serial, and local file APIs for servers that coordinate amongst modules. In some cases, I have developed modules as libraries for Processing [102], especially when that enabled other users to build with the modules; for example, I developed the work in Chapter 10 as a guest of the Unstable Design Lab of the ATLAS Institute at University of Colorado Boulder, where the TC2 loom is frequently used for interdisciplinary research and art practice. I particularly try to target browser-based interfaces because of the ubiquitous availability of browsers.

Underpinning the multiple possible visualizations shown in Figure 15.3 is a necessary dual representation: the knitting is modeled as a *partially scheduled stitch graph*, which can be operated on as a traversible and editable stitch graph as well as it can be represented as a sequence of Knitout operations. This allows interoperability between CAD- and CAM-based workflows (as defined in Subsection 11.4.2).

15.5. What softness is not: a note about "AI"

Softness is not amorphousness. The structures of softness, from complex weaving and knitting patterns to computational modularity and composability, are what gives it meaning and power.

As I am writing this in summer 2024, "generative AI"—in this work, I mean this in the specific sense of Large Language Models [30] and other high-data probabilistic/statistical methods, as opposed to the earlier, more general sense of "an algorithmic model that produces something"—are being promoted and studied as a solution to many creativity support tasks.

I believe that "AI" can be a useful *component* of a soft system (concretely, by being a way to implement the camera-based sensing I discuss in Subsection 15.4.1), and some aspects of statistical approaches share a kinship with softness—particularly, generative AI is an example of emergence at a dizzying scale, any given model does have its own telltale quirks (i.e. its grain), and the wide array of interface systems that are built on top of ChatGPT show that it can be thought of as configurable. These aspects have been studied in HCI, particularly in the highly relevant thread of research on the materiality of "generative AI" and how people interact with it [425, 428]. However, I don't consider generative AI to be particularly well-aligned to the definition of softness that I propose in this dissertation, for two main reasons.

First, it's not just malleable, it's downright amorphous: it lacks the structure of sub-component modularity and composability that I propose as a basis for constructing usefully soft workflows.

Second: "generative AI," as currently exists, is premised on acontextuality. Of course, any actual such system has a context: it is shaped by the goals of its creators, and it is trained on a specific dataset, even if it is a very large one. However, it is presented as a universal technology, and indeed the sheer scale of the underlying data has a kind of flattening effect, in which any particular result might as well be any other. Individual creators can, and do, create unique and situated work from these kinds of systems, but I don't think it's a coincidence that some of the most poetic work from generative AI involves creators intervening at the training data level, such as in Anna Ridler's "Myriad (Tulips)" [317].

16. Softer Technologies: Future Work

The aspects and approaches to softness in this dissertation range from literal (e.g. the softness of a textile) to metaphorical (e.g. the softness of a hybrid context). Within the work documented, these senses flow from each other—the literal material softness of fabrics have made them suitable for an incredible variety of contexts, from the personal to the industrial, while resisting turn-key fabrication pipelines. Systems like the brioche knitting generators include softness at all three levels of material, workflow, and context.

Looking forward, it's possible to think about softness as an approach in a more targeted manner, without necessarily being about fabric or computational fiber arts specifically.

16.1. Sustainable and Local Material Futures

Sustainability is a critical aspect of any kind of fabrication. Mending or re-using existing objects and incorporating materials that are found locally are both promising tactics to reduce resource consumption; however, these are challenging in fully-automated computational fabrication, which is more typically engineered to build fully new things, from standardized materials. Interactive fabrication, that can incorporate real-time and real-space inputs, has been a promising lead toward augmenting existing objects [418] and working with non-uniform materials [195]. I believe this can be pushed even further by recognizing that local materials often entail local material technologies— for example, materials like clay, bamboo, textiles, and wire armatures [270, 271] are often the basis of vernacular crafts, with both deep expertise amongst artisans as well as some population-wide familiarity. Jacobs showed the possibility of using computational fabrication as the basis of a collaboration with artisans in Ju/'Hoan artisans in Namibia who had an established practice of carving ostrich eggshell for jewelry [159]. Soft approaches that can handle underdefined and emergent physical characteristics, composable abstractions that embrace experiential inputs, and overall modular systems designed to be extensible by users can help make hybrid fabrication systems which could be tuned to specific local material contexts for both cultural connections and to leverage local and re-used materials.

16.2. Material Learning

Workflows exist on various time scales, from a single session producing a single output to the meta-workflow of a decades-long material practice [247].

I observe that learning is very often a soft process, full of unpredictable results and shifting goals. Many pedagogical approaches, certainly including my own, emphasize experiential and reflective workflows. And HCI research very often prioritizes "novice" users. But learning is not just one process, especially as it unfolds over time. Learning can happen over wildly different time scales: in brief weekend workshops, over the course of a year-long research, and through years of engagement with a hobby [12]. There are many models of skill acquisition, but they all include the idea that learning includes pivots between different modes, for example from undirected exploration to initial goal-setting, or from curating results to combining the best aspects of each. Different learners, in different modes, need different workflows. Weekend workshop versus three years of baskets are wildly different time scales. Additionally, a "novice user" is not a blank slate [134]. Learners also bring their own skills, preferences, and contexts, especially cultural and aesthetic ones.

Soft approaches could be brought to bear not just workflows per se, but longer-term arcs of material practice, which are often both cultural and also intensely personal. Such systems must be configurable, to remain contextually appropriate at all of the parts of that journey; underdetermination and emergence could supply experiential "spark" and can be varied to the appropriate blend of scaffolding and agency for a particular practitioner.

16.3. Fuzzy Computation

We are in an era when the stuff of "computation" itself is increasingly probabilistic and fuzzily-defined. As I discussed in Section 15.5, I see contemporary "generative AI," which is increasingly included in creativity support systems, as being unfortunately amorphous and acontextual. I believe that soft structure might be added to purely stochastic systems to scope and channel underdetermination. For example, Yang notes that designers working with AI are engaged in a process of growing and refining a "working set" of "designerly abstractions" about the capabilities of AI [425]. Looking to the diagrammatic notation systems that I describe as a hallmark of soft systems, inspiration could be drawn from the way artisans develop their own notational systems to bridge between low-level operations and their creative intent. Another tactic, discussed in Subsection 10.7.3, is the careful hybridization of more- and less-deterministic methods, to allow the more regular one to temper the results. For example, the "weaving shader" constraint imposed weavability on the image in the self-portrait sketch, no matter how chaotic the video feed got; in the Twitch sketch, repeating the pattern across the width of the fabric gave it an overall gestalt compositional frame to tame the chaos.

Soft approaches to reflective practice have implications not just for fabrication with physical materials, but also for the broader landscape of computational creativity.

17. Conclusion

In this work, I have shown how ***soft*** approaches like adaptability, underdetermination, and hybridity can be applied throughout the stack of material manipulation— from underlying physical properties through medium-specific workflows to contexts of use—to support everything from learning how to work with cutting-edge materials to designing sustainable systems for repair.

By identifying the lens of softness, I provide a conceptual frame and a set of system-building tactics that can be used to create flexible and adaptable computational fabrication systems. Such systems can be built to work with unusual materials, to support varied contexts, and to adapt to current and futures needs. To confront this complex topic, my work unites methodologies from technical systems research and design inquiry.

This work includes a number of supporting contributions:

- operational fabrication techniques not previously documented in HCI literature, including an explanation of manual v-bed knitting machines (Chapter 13) and the "spacer" and "brioche" knit fabric types (Chapter 6 and Chapter 14)
- specific low-level techniques for producing complex mechanisms with machine knitting (Chapter 5 and Chapter 6)
- demonstrations of effective use of camera-based sensing in fabrication tasks (Chapter 10, Chapter 9, Chapter 13, Chapter 14)
- the design of a uniquely interactive Jacquard handloom with on-loom input and networked capabilities (Chapter 9)
- a design exploration of procedural generation in the context of fabrication
- the term "grain space" to describe an approach to creating fabrication tools, and an exemplar grain space of brioche knitting
- insights from creative practitioners on the roles of machine and material agency, social aspects of craft, and procedural design in their work (Chapter 13, Chapter 14)

Computational fabrication can help people make things for their own specific contexts, discover new material technologies, and work with local and re-used materials and machines—but only if we approach fabrication systems critically, as a site of adaptable and contextual possibilities, not simply scaled-down versions of industrial factories. Approaches like augmention, contextual input, and modular abstractions are how we can find complexities and opportunities in the shifting grains and at the blurry margins between material and computational processes.

* * *

⁂ References

1. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/
2. Lilý Erla Adamsdóttir and Trausti Dagsson. 2011. *KnitBird*. Iceland. https://knitbird.com/ (retrieved December 30, 2022)
3. Adobe Inc. 2015. *Adobe Capture*. https://www.adobe.com/products/capture.html (retrieved August 21, 2021)
4. Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. 2007. *Compilers: Principles, Techniques, & Tools*. Pearson/Addison Wesley.
5. Ahrens and Violette Looms (AVL Looms Inc). 2013. **Little Weaver Manual**. https://www.avlusa.com/downloads/LW_Manual_Current.pdf (retrieved December 29, 2020)
6. Ahrens and Violette Looms (AVL Looms Inc). 2017. **Price List September 2017**. http://www.avlusa.com/downloads/AVLPriceList.pdf (retrieved October 10, 2017)
7. Ahrens and Violette Looms (AVL Looms Inc). 2020. **Jacq3G Jacquard Loom**. https://avllooms.com/products/jacq3g (retrieved September 3, 2020)
8. Ahrens and Violette Looms (AVL Looms Inc). 2024. **Little Weaver 2 - AVL Looms Inc**. https://avllooms.com/products/little-weaver?variant=45391934849327 (retrieved July 18, 2024)
9. Roland Aigner, Mira Alida Haberfellner, and Michael Haller. 2022. **spaceR: Knitting Ready-Made, Tactile, and Highly Responsive Spacer-Fabric Force Sensors for Continuous Input**. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 1–15. https://doi.org/10.1145/3526113.3545694
10. Lea Albaugh. 2014. *Clothing For Moderns*. http://lea.zone/clothing-for-moderns.html
11. Lea Albaugh. 2018. **Little Looms**. http://lea.zone/little-looms.html (retrieved December 28, 2020)
12. Lea Albaugh, Jesse T Gonzalez, and Scott E Hudson. 2024. **Tensions and Resolutions in Hybrid Basketry: Joining 3D Printing and Handweaving**. In *Proceedings of the Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction*, 1–13. https://doi.org/10.1145/3623509.3633400
13. Lea Albaugh, April Grow, Chenxi Liu, James McCann, Gillian Smith, and Jennifer Mankoff. 2016. **Threadsteading: Playful Interaction for Textile Fabrication Devices**. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 285–288. https://doi.org/10.1145/2851581.2889466

14. Lea Albaugh, Scott E. Hudson, Lining Yao, and Laura Devendorf. 2020. **Investigating Underdetermination Through Interactive Computational Handweaving**. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, 1033–1046. https://doi.org/10.1145/3357236.3395538
15. Lea Albaugh, Scott Hudson, and Lining Yao. 2019. **Digital Fabrication of Soft Actuated Objects by Machine Knitting**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3290605.3300414
16. Lea Albaugh, James McCann, Lining Yao, and Scott E. Hudson. 2021. **Enabling Personal Computational Handweaving with a Low-Cost Jacquard Loom**. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–10. https://doi.org/10.1145/3411764.3445750
17. Anni Albers. 1965. *On Weaving*. Wesleyan University Press.
18. Byoungkwon An, Ye Tao, Jianzhe Gu, Tingyu Cheng, Xiang 'Anthony'Chen, Xiaoxiao Zhang, Wei Zhao, Youngwook Do, Shigeo Takahashi, Hsiang-Yun Wu, Teng Zhang, and Lining Yao. 2018. **Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3173574.3173834
19. Kristina Andersen, Ron Wakkary, Laura Devendorf, and Alex McLean. 2019. **Digital Crafts-Machine-Ship: Creative Collaborations with Machines**. *Interactions* 27, 1: 30–35. https://doi.org/10.1145/3373644
20. Andreas Müller. 2015. **AYAB - All Yarns Are Beautiful**. https://ayab-knitting.com/ (retrieved September 7, 2022)
21. Anonymous. 2022. **Is It True That the First Computer Was a Loom?** *Quora*. https://www.quora.com/Is-it-true-that-the-first-computer-was-a-loom-How-is-a-loom-a-computer (retrieved August 2, 2024)
22. Mario de Araújo, Raul Fangueiro, André Catarino, and Hong Hu. 2001. **Recent Developments in Weft-Knitting Science and Technology: The Way Ahead in the New Millennium**. *Revista Romana de Textile–Pielarie* 1: 61–66.
23. Marcia Ascher and Robert Ascher. 1981. *Code of the Quipu: A Study in Media, Mathematics, and Culture*. University of Michigan Press, Ann Arbor.
24. Ozgur Atalay, Asli Tuncay, Muhammad D Husain, and William R Kennon. 2017. **Comparative Study of the Weft-Knitted Strain Sensors**. *Journal of Industrial Textiles* 46, 5: 1212–1240. https://doi.org/10.1177/1528083715619948
25. Andy Baker and Mark Koors. 2017. **AndyMark String Potentiometer Am-2674**. http://files.andymark.com/am-2674_StringPot_Assembly_Instructions.pdf (retrieved September 17, 2018)
26. Yuliya Baranovskaya, Marshall Prado, Moritz Dörstelmann, and Achim Menges. 2016. **Knitflatable Architecture - Pneumatically Activated Preprogrammed Knitted Textiles**. 571–580. https://doi.org/10.52842/conf.ecaade.2016.1.571
27. David Edward Baron. 2017. **Fluid Dynamics of Watercolor Painting : Experiments and Modelling**. Master's thesis: Montclair State University.

28. Patrick Baudisch and Stefanie Mueller. 2017. **Personal Fabrication**. *Foundations and Trends® in Human-Computer Interaction* 10, 3–4: 165–293. https://doi.org/10.1561/1100000055
29. Mark Beecroft. 2016. **3D Printing of Weft Knitted Textile Based Structures by Selective Laser Sintering of Nylon Powder**. *IOP Conference Series: Materials Science and Engineering* 137: 012017. https://doi.org/10.1088/1757-899X/137/1/012017
30. Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. **On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜**. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 610–623. https://doi.org/10.1145/3442188.3445922
31. Martin P. Bendsøe. 1989. **Optimal Shape Design as a Material Distribution Problem**. *Structural Optimization* 1, 4: 193–202. https://doi.org/10.1007/BF01650949
32. Margarita Benitez, Nathan Clark, Chris Yanc, Markus Vogl, Ed Bennett, and Gary Shimomura. 2013. **Open Source Jacquard Loom Project**. http://www.osloom.org/ (retrieved December 29, 2020)
33. Walter Benjamin. 1935. **The Work of Art in the Age of Mechanical Reproduction**. In *Illuminations*, Hannah Arendt (ed.). Schocken Books, New York, NY, USA, 26.
34. Joanna Berzowska and Maksim Skorobogatiy. 2010. **Karma Chameleon: Bragg Fiber Jacquard-Woven Photonic Textiles**. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, 297. https://doi.org/10.1145/1709886.1709950
35. Glenn Blauvelt, Thomas Wrensch, and Michael Eisenberg. 2000. **Integrating Craft Materials and Computation**. *Knowledge-Based Systems* 13, 7-8: 471–478.
36. Leonardo Bonanni, Sam Sarcia, Subodh Paudel, and Ted Selker. 2005. **Dishmaker: Personal Fabrication Interface**. In *Extended Abstracts on Human Factors in Computing Systems*.
37. Katya Borgos-Rodriguez, Maitraye Das, and Anne Marie Piper. 2021. **Melodie: A Design Inquiry into Accessible Crafing Through Audio-Enhanced Weaving**. *ACM Transactions on Accessible Computing* 14, 1: 30. https://doi.org/10.1145/3444699
38. Samuelle Bourgault, Pilar Wiley, Avi Farber, and Jennifer Jacobs. 2023. **CoilCAM: Enabling Parametric Design for Clay 3D Printing Through an Action-Oriented Toolpath Programming System**. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–16. https://doi.org/10.1145/3544548.3580745
39. Michael Boyle, Christopher Edwards, and Saul Greenberg. 2000. **The Effects of Filtered Video on Awareness and Privacy**. In *Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work*, 1–10. https://doi.org/10.1145/358916.358935
40. Gary Bradski. 2000. *The OpenCV Library*. Dr. Dobb's Journal of Software Tools. https://opencv.org/ (retrieved September 18, 2024)
41. Matthew Bragg. *DesignaKnit 9*. SoftByte Ltd. https://www.softbyte.co.uk/designaknit9.htm (retrieved December 30, 2022)

42. Virginia Braun and Victoria Clarke. 2019. **Reflecting on Reflexive Thematic Analysis**. *Qualitative Research in Sport, Exercise and Health* 11, 4: 589–597. https://doi.org/10.1080/2159676X.2019.1628806

43. JC Briar. 2013. **Stitch Maps**. https://stitch-maps.com/ (retrieved September 18, 2024)

44. Sander de Bruijne. 2015. *Stitch Fiddle*. Stitch Fiddle, Amsterdam, The Netherlands. https://www.stitchfiddle.com/en/chart/create/knitting (retrieved December 30, 2022)

45. Toni Buckby. 2018. **DIY Laser Cut Rigid Heddle Loom**. *Instructables*. https://www.instructables.com/DIY-Laser-Cut-Rigid-Heddle-Loom-Part-1-Building-th/ (retrieved December 28, 2020)

46. Leah Buechley, Daniela K. Rosner, Eric Paulos, and Amanda Williams. 2009. **DIY for CHI: Methods, Communities, and Values of Reuse and Customization**. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, 4823. https://doi.org/10.1145/1520340.1520750

47. Jane Camann. 2012. *EnvisioKnit Design Studio*. https://www.envisioknit.com/ (retrieved December 30, 2022)

48. John Canny. 1986. **A Computational Approach to Edge Detection**. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 8, 06: 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

49. Canva. 2013. *Free Design Tool: Presentations, Video, Social Media*. Canva. https://www.canva.com (retrieved September 15, 2022)

50. Daniel Cardoso Llach. 2020. **Between Accident and Control: Contrasting Traditions of Computational Design**. In *WWW Drawing: Architectural Drawing from Pencil to Pixel*, Mehrdad Hadighi and Janet Abrams (eds.). ACTAR Publishers, Barcelona.

51. Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. 2020. **Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification**. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–8. https://doi.org/10.1145/3334480.3382839

52. Erin A. Carroll and Celine Latulipe. 2009. **The Creativity Support Index**. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, 4009–4014. https://doi.org/10.1145/1520340.1520609

53. Erin Catto. 2007–2020. *Box2D*. https://github.com/erincatto/Box2D

54. Amy Cheatle and Steven J. Jackson. 2015. **Digital Entanglements: Craft, Computation and Collaboration in Fine Art Furniture Production**. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 958–968. https://doi.org/10.1145/2675133.2675291

55. Erin Cherry and Celine Latulipe. 2014. **Quantifying the Creativity Support of Digital Tools Through the Creativity Support Index**. *ACM Transactions on Computer-Human Interaction* 21, 4: 1–25. https://doi.org/10.1145/2617588

56. Inrak Choi, Nick Corson, Lizzie Peiros, Elliot W. Hawkes, Sean Keller, and Sean Follmer. 2018. **A Soft, Controllable, High Force Density Linear Brake Utilizing Layer Jamming**. *IEEE Robotics and Automation Letters* 3, 1: 450–457. https://doi.org/10.1109/LRA.2017.2761938

57. Emma Cocker. 2017. **Weaving Codes/Coding Weaves: Penelopean Mêtis and the Weaver-Coder's Kairos**. *TEXTILE* 15, 2: 124–141. https://doi.org/10.1080/14759756.2017.1298233

58. Kate Compton (@galaxykate). 2016. *Artistic Intentions (@Artideabot)*. https://twitter.com/artideabot (retrieved January 6, 2022)

59. Kate Compton (@galaxykate). 2018. *A Baffling Murder (@Bafflingbot)*. https://twitter.com/bafflingbot

60. Kate Compton and Michael Mateas. 2015. **Casual Creators**. In *Proceeedings of the Sixth International Conference on Computational Creativity*, 228–235.

61. L. J. Comrie. 1932. **The Application of the Hollerith Tabulating Machine to Brown's Tables of the Moon.** *Monthly Notices of the Royal Astronomical Society* 92, 7: 694–701. https://doi.org/10.1093/mnras/92.7.694

62. Alessandrina Costa. 2008. **Alessandrina.com**. *Machine Knitting*. https://alessandrina.com/category/machine-knitting/ (retrieved February 2, 2022)

63. Alessandrina Costa. 2023. **Traveling Between Brands, Duo and E6, Settings and Techniques**. *alessandrina.com*. https://alessandrina.com/2023/08/28/traveling-between-brands-duo-and-e6-settings-and-techniques/ (retrieved September 18, 2024)

64. Alessandrina Costa. 2023. **Seasonal Knits Inspired by Published Repeats 1**. *alessandrina.com*. https://alessandrina.com/2023/12/06/seasonal-knits-inspired-by-published-repeats/ (retrieved September 18, 2024)

65. Alessandrina Costa. 2024. **Mac Experiments on Printing Needle Tapes and Punchcard Templates to Scale and Other Tips**. *alessandrina.com*. https://alessandrina.com/2024/03/21/mac-experiments-on-printing-needle-tapes-and-punchcard-templates-to-scale-and-other-tips/ (retrieved September 18, 2024)

66. Andy Dearden and Janet Finlay. 2006. **Pattern Languages in HCI: A Critical Review**. *Human–Computer Interaction* 21, 1: 49–102. https://doi.org/10.1207/s15327051hci2101_3

67. Ashley Del Valle, Mert Toka, and Jennifer Jacobs. 2024. **Engaging Young People in the Expressive Opportunities of Digital Fabrication Through Craft-Oriented CAM-Based Design**. In *Designing Interactive Systems Conference*, 1162–1176. https://doi.org/10.1145/3643834.3660693

68. Dena Molnar and Chelsea Molnar. **WOVNS**. https://www.wovns.com/ (retrieved December 29, 2022)

69. Audrey Desjardins, Cayla Key, Heidi R. Biggs, and Kelsey Aschenbeck. 2019. **Bespoke Booklets: A Method for Situated Co-Speculation**. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, 697–709. https://doi.org/10.1145/3322276.3322311

70. Laura Devendorf, Kristina Andersen, Daniela K. Rosner, Ron Wakkary, and James Pierce. 2019. **From HCI to HCI-Amusement: Strategies for Engaging What New Technology Makes Old**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3290605.3300265

71. Laura Devendorf, Katya Arquilla, Sandra Wirtanen, Allison Anderson, and Steven Frost. 2020. **Craftspeople as Technical Collaborators: Lessons Learned Through an Experimental Weaving Residency**. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3313831.3376820

72. Laura Devendorf, Abigail De Kosnik, Kate Mattingly, and Kimiko Ryokai. 2016. **Probing the Potential of Post-Anthropocentric 3D Printing**. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16*, 170–181. https://doi.org/10.1145/2901790.2901879

73. Laura Devendorf, Sasha de Koninck, and Etta Sandry. 2022. **An Introduction to Weave Structure for HCI: A How-to and Reflection on Modes of Exchange**. In *Designing Interactive Systems Conference*, 629–642. https://doi.org/10.1145/3532106.3534567

74. Laura Devendorf, Joanne Lo, Noura Howell, Jung Lin Lee, Nan-Wei Gong, M. Emre Karagozler, Shiho Fukuhara, Ivan Poupyrev, Eric Paulos, and Kimiko Ryokai. 2016. **"I Don't Want to Wear a Screen": Probing Perceptions of and Possibilities for Dynamic Displays on Clothing**. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 6028–6039. https://doi.org/10.1145/2858036.2858192

75. Laura Devendorf and Kimiko Ryokai. 2013. **AnyType: Provoking Reflection and Exploration with Aesthetic Interaction**. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1041–1050. https://doi.org/10.1145/2470654.2466133

76. Laura Devendorf and Kimiko Ryokai. 2015. **Being the Machine: Reconfiguring Agency and Control in Hybrid Fabrication**. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2477–2486. https://doi.org/10.1145/2702123.2702547

77. Laura Devendorf, Kathryn Walters, Marianne Fairbanks, Etta Sandry, and Emma R Goodwill. 2023. **AdaCAD: Parametric Design as a New Form of Notation for Complex Weaving**. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–18. https://doi.org/10.1145/3544548.3581571

78. Digital Weaving Norway. 2020. **TC2 Loom | Digital Weaving Machine**. https://www.digitalweaving.no/tc2-loom/ (retrieved September 3, 2020)

79. Mustafa Doga Dogan, Patrick Baudisch, Hrvoje Benko, Michael Nebeling, Huaishu Peng, Valkyrie Savage, and Stefanie Mueller. 2022. **Fabricate It or Render It? Digital Fabrication Vs. Virtual Reality for Creating Objects Instantly**. atExtended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA. https://doi.org/10.1145/3491101.3516510

80. Tamara Anna Efrat, Moran Mizrahi, and Amit Zoran. 2016. **The Hybrid Bricolage: Bridging Parametric Design with Craft Through Algorithmic Modularity**. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 5984–5995. https://doi.org/10.1145/2858036.2858441

81. David Eickhoff, Stefanie Mueller, and Patrick Baudisch. 2016. **Destructive Games: Creating Value by Destroying Valuable Physical Objects**. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 3970–3974. https://doi.org/10.1145/2858036.2858113

82. Hassan M. El-Dessouky and Mohamed N. Saleh. 2018. **3D Woven Composites: From Weaving to Manufacturing**. In *Recent Developments in the Field of Carbon Fibers*, Rita Khanna and Romina Cayumil (eds.). IntechOpen, Rijeka, 51–66. https://doi.org/10.5772/intechopen.74311

83. Elkágyé. 21 January 2018, 07:23:16. *Manually Operated Flat Knitting Machine*. https://commons.wikimedia.org/wiki/File:V-bed_flat_knitting_machine.jpg (retrieved September 12, 2022)

84. Shreyosi Endow, Mohammad Abu Nasir Rakib, Anvay Srivastava, Sara Rastegarpouyani, and Cesar Torres. 2022. **Embr: A Creative Framework for Hand Embroidered Liquid Crystal Textile Displays**. In *CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3491102.3502117

85. Osuanyi Quaicoo Essel. 2017. **Deconstructing the Concept of "African Print" in the Ghanaian Experience**. *Africology: The Journal of Pan African Studies* 11, 1. https://www.jpanafrican.org/docs/vol11no1/11.1-5-Essel-final.pdf

86. James Essinger. 2004. *Jacquard's Web: How a Hand-Loom Led to the Birth of the Information Age*. Oxford University Press, Oxford, UK.

87. Marianne Fairbanks, Erica Hess, Kat Bunke, and Sofia Hagström Møller. 2020. **HELLO! LOOMS**. *Weaving Lab*. https://www.weavinglab.com/tools (retrieved December 29, 2020)

88. Ylva Fernaeus, Martin Jonsson, and Jakob Tholander. 2012. **Revisiting the Jacquard Loom: Threads of History and Current Patterns in HCI**. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, 1593. https://doi.org/10.1145/2207676.2208280

89. Wout Fierens. 2012–2018. *SVG.JS*. https://svgdotjs.github.io/ (retrieved September 18, 2024)

90. Sean Follmer, David Carr, Emily Lovell, and Hiroshi Ishii. 2010. **CopyCAD: Remixing Physical Objects with Copy and Paste from the Real World**. In *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, 381–382. https://doi.org/10.1145/1866218.1866230

91. Sean Follmer and Hiroshi Ishii. 2012. **KidCAD: Digitally Remixing Toys Through Tangible Tools**. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, 2401. https://doi.org/10.1145/2207676.2208403

92. Sean Follmer, Daniel Leithinger, Alex Olwal, Nadia Cheng, and Hiroshi Ishii. 2012. **Jamming User Interfaces: Programmable Particle Stiffness and Sensing for Malleable and Shape-Changing Devices**. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 519. https://doi.org/10.1145/2380116.2380181

93. Jessica Forbes and Cassidy Forbes. 2015. **Ravelry: AYAB Members**. https://www.ravelry.com/groups/ayab/members (retrieved September 13, 2022)

94. Jessica Forbes and Cassidy Forbes. 2022. **Ravelry: About Our Site**. *Ravelry*. https://www.ravelry.com/about (retrieved December 31, 2022)

95. Jack Forman, Mustafa Doga Dogan, Hamilton Forsythe, and Hiroshi Ishii. 2020. **DefeXtiles: 3D Printing Quasi-Woven Fabric via Under-Extrusion**. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 1222–1233. https://doi.org/10.1145/3379337.3415876

96. Jack Forman, Ozgun Kilic Afsar, Sarah Nicita, Rosalie Hsin-Ju Lin, Liu Yang, Megan Hofmann, Akshay Kothakonda, Zachary Gordon, Cedric Honnet, Kristen Dorsey, Neil Gershenfeld, and Hiroshi Ishii. 2023. **FibeRobo: Fabricating 4D Fiber Interfaces by Continuous Drawing of Temperature Tunable Liquid Crystal Elastomers**. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–17. https://doi.org/10.1145/3586183.3606732

97. Jack Forman, Taylor Tabb, Youngwook Do, Meng-Han Yeh, Adrian Galvin, and Lining Yao. 2019. **ModiFiber: Two-Way Morphing Soft Thread Actuators for Tangible Interaction**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–11. https://doi.org/10.1145/3290605.3300890

98. Blender Foundation. **Blender.org - Home of the Blender Project - Free and Open 3D Creation Software**. *blender.org*. https://www.blender.org/ (retrieved December 12, 2020)

99. Freddy Brault. *Pointcarre CAD*. http://www.pointcarre.com/ (retrieved December 29, 2022)

100. Mikhaila Friske, Shanel Wu, and Laura Devendorf. 2019. **AdaCAD: Crafting Software For Smart Textiles Design**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3290605.3300575

101. Devon Frost, Raina Lee, Eun-Ha Paek, and Jennifer Jacobs. 2024. **SketchPath: Using Digital Drawing to Integrate the Gestural Qualities of Craft in CAM-Based Clay 3D Printing**. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–16. https://doi.org/10.1145/3613904.3642684

102. Ben Fry and Casey Reas. 2001–2024. *Processing.org*. Processing Foundation. https://processing.org/ (retrieved December 12, 2020)

103. Philip Galanter. 2003. **What Is Generative Art? Complexity Theory as a Context for Art Theory**. In *6th Generative Art Conference*. https://generativeart.com/on/cic/papersGA2003/a22.pdf

104. Gary Garrels (ed.). 2000. *Sol LeWitt: A Retrospective*. Yale University Press, New Haven, CT, USA.

105. Martha Glazzard. 2014. **Re-Addressing the Role of Knitted Textile Design Knowledge: Auxetic Textiles from a Practice-Led, Designer-Maker Perspective**. PhD Thesis: Nottingham Trent University.

106. Seth Glickman, Nathan McKenzie, Joseph Seering, Rachel Moeller, and Jessica Hammer. 2018. **Design Challenges for Livestreamed Audience Participation Games**. In *The Annual Symposium on Computer-Human Interaction in Play Extended Abstracts*, 187–199. https://doi.org/10.1145/3242671.3242708

107. Glowforge. *Glowforge*. https://glowforge.com/

108. Gabriela Goldschmidt. 1994. **On Visual Design Thinking: The Vis Kids of Architecture**. *Design Studies* 15, 2: 158–174. https://doi.org/10.1016/0142-694X(94)90022-1

109. Antonio Gomes, Andrea Nesbitt, and Roel Vertegaal. 2013. **MorePhone: An Actuated Shape Changing Flexible Smartphone**. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, 2879–2880. https://doi.org/10.1145/2468356.2479558

110. Jun Gong, Yu Wu, Lei Yan, Teddy Seyed, and Xing-Dong Yang. 2019. **Tessutivo: Contextual Interactions on Interactive Fabrics with Inductive Sensing**. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 29–41. https://doi.org/10.1145/3332165.3347897

111. Michael A. Goodrich and Alan C. Schultz. 2007. **Human-Robot Interaction: A Survey**. *Foundations and Trends® in Human-Computer Interaction* 1, 3: 203–275. https://doi.org/10.1561/1100000005

112. Bruna Goveia da Rocha, Oscar Tomico, Panos Markopoulos, and Daniel Tetteroo. 2020. **Crafting Research Products Through Digital Machine Embroidery**. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, 341–350. https://doi.org/10.1145/3357236.3395443

113. Grand View Research. 2024. *Textile Market Size, Share & Trends Analysis Report By Raw Material (Wool, Chemical, Silk), By Product (Natural Fibers, Polyester), By Application, By Region, And Segment Forecasts, 2024 - 2030*. Grand View Research. https://www.grandviewresearch.com/industry-analysis/textile-market

114. Jordan Graves and Anne Sullivan. 2020. **eLoominate: Tools for Casual Creation in Hybrid Craft**. In *Proceedings of the International Conference on Computational Creativity 2020 Workshops*, 5. https://ceur-ws.org/Vol-2827/CAC-Paper_4.pdf

115. Theodore Gray and Nina Paley. 2015. **Quantitative Stretching**. http://home.theodoregray.com/stitchblog/2015/2/1/quantitative-stretching

116. Anthony Grbic and George V. Eleftheriades. 2002. **Experimental Verification of Backward-Wave Radiation from a Negative Refractive Index Metamaterial**. *Journal of Applied Physics* 92, 10: 5930–5935. https://doi.org/10.1063/1.1513194

117. Saul Greenberg and Chester Fitchett. 2001. **Phidgets: Easy Development of Physical Interfaces Through Physical Widgets**. In *Proceedings of the 14th annual ACM symposium on user interface software and technology*, 209–218. https://doi.org/10.1145/502348.502388

118. Tobias Grosse-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S. Reynolds, and Joshua R. Smith. 2017. **Finding Common Ground: A Survey of Capacitive Sensing in Human-Computer Interaction**. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 3293–3315. https://doi.org/10.1145/3025453.3025808

119. Jianzhe Gu, David E. Breen, Jenny Hu, Lifeng Zhu, Ye Tao, Tyson Van de Zande, Guanyun Wang, Yongjie Jessica Zhang, and Lining Yao. 2019. **Geodesy: Self-Rising 2.5D Tiles by Printing Along 2D Geodesic Closed Path**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–10. https://doi.org/10.1145/3290605.3300267

120. Jianzhe Gu, Vidya Narayanan, Guanyun Wang, Danli Luo, Harshika Jain, Kexin Lu, Fang Qin, Sijia Wang, James McCann, and Lining Yao. 2020. **Inverse Design Tool for Asymmetrical Self-Rising Surfaces with Color Texture**. In *Symposium on Computational Fabrication*, 1–12. https://doi.org/10.1145/3424630.3425420

121. Susan Guagliumi. 2008. *Hand-Manipulated Stitches for Machine Knitters*. Taunton Press, Newtown, CT.

122. Runbo Guo, Jenny Lin, Vidya Narayanan, and James McCann. 2020. **Representing Crochet with Stitch Meshes**. In *Symposium on Computational Fabrication*, 1–8. https://doi.org/10.1145/3424630.3425409

123. Lars Hallnäs and Johan Redström. 2001. **Slow Technology – Designing for Reflection**. *Personal and Ubiquitous Computing* 5, 3: 201–212. https://doi.org/10.1007/PL00000019

124. Nur Al-huda Hamdan, Simon Voelker, and Jan Borchers. 2018. **Sketch&Stitch: Interactive Embroidery for E-Textiles**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3173574.3173656

125. BL Hardy, M-H Moncel, Céline Kerfant, Matthieu Lebon, Ludovic Bellot-Gurlet, and Nicolas Mélard. 2020. **Direct Evidence of Neanderthal Fibre Technology and Its Cognitive and Behavioral Implications**. *Scientific Reports* 10, 1: 1–9.

126. Claire Harvey, Emily Holtzman, Joy Ko, Brooks Hagan, Rundong Wu, Steve Marschner, and David Kessler. 2019. **Weaving Objects: Spatial Design and Functionality of 3D-Woven Textiles**. *Leonardo* 52, 4: 381–388. https://doi.org/10.1162/leon_a_01780

127. Alice C Haynes and Jürgen Steimle. 2024. **Flextiles: Designing Customisable Shape-Change in Textiles with SMA-Actuated Smocking Patterns**. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–17. https://doi.org/10.1145/3613904.3642848

128. Maria Hellstrom. 2013. **Knitting Ourselves into Being: The Case of Labour and Hip Domesticity on the Social Network Ravelry.com**. Master's thesis: Victoria University of Wellington, Wellington, New Zealand.

129. Lejaren A. Hiller. 1959. **Computer Music**. *Scientific American* 201, 6: 109–121. http://www.jstor.org/stable/24941187 (retrieved August 3, 2024)

130. Mare E. Hirsch. 2022. **Making in Time: How Timescale Impacts the Experience, Outcomes, and Expressive Opportunities of Digital Fabrication**. PhD Thesis: https://www.proquest.com/dissertations-theses/making-time-how-timescale-impacts-experience/docview/2716926322/se-2

131. Mare Hirsch, Jennifer Jacobs, and Yon Visell. 2019. **SK3TCH: Fast, Interactive 3D Printing via Liquid Crystallization**. atACM Symposium on Computational Fabrication, Pittsburgh, PA, USA.

132. Megan Hofmann, Lea Albaugh, Ticha Sethapakadi, Jessica Hodgins, Scott E. Hudson, James McCann, and Jennifer Mankoff. 2019. **KnitPicking Textures: Programming and Modifying Complex Knitted Textures for Machine and Hand Knitting**. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 5–16. https://doi.org/10.1145/3332165.3347886

133. Megan Hofmann, Gabriella Hann, Scott E. Hudson, and Jennifer Mankoff. 2018. **Greater Than the Sum of Its PARTs: Expressing and Reusing Design Intent in 3D Models**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3173574.3173875

134. Megan Hofmann, Kristin Williams, Toni Kaplan, Stephanie Valencia, Gabriella Hann, Scott E. Hudson, Jennifer Mankoff, and Patrick Carrington. 2019. **"Occupational Therapy Is Making": Clinical Rapid Prototyping and Digital Fabrication**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3290605.3300544

135. Douglas Hofstadter. 1986. *Metamagical Themas*. Basic Books, New York.

136. Dónal P. Holland, Evelyn J. Park, Panagiotis Polygerinos, Gareth J. Bennett, and Conor J. Walsh. 2014. **Actuators | Soft Robotics Toolkit**. https://softroboticstoolkit.com/actuators

137. Dónal P. Holland, Evelyn J. Park, Panagiotis Polygerinos, Gareth J. Bennett, and Conor J. Walsh. 2014. **The Soft Robotics Toolkit: Shared Resources for Research and Design**. *Soft Robotics* 1, 3: 224–230. https://doi.org/10.1089/soro.2014.0010

138. Cedric Honnet, Hannah Perner-Wilson, Marc Teyssier, Bruno Fruchard, Jürgen Steimle, Ana C Baptista, and Paul Strohmeier. 2020. **PolySense: Augmenting Textiles with Electrical Functionality Using In-Situ Polymerization**. 13. https://doi.org/10.1145/3313831.3376841

139. Kristina Höök. 2018. *Designing with the Body: Somaesthetic Interaction Design*. MIT Press.

140. Melody Horn, Amy Traylor, and Leah Buechley. 2022. **Slabforge: Design Software for Slab-Based Ceramics**. In *CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3491102.3517663

141. Hot World Media GmbH. 2020. **Repetier-Firmware**. https://www.repetier.com/documentation/repetier-firmware/ (retrieved September 3, 2020)

142. Hot World Media GmbH. 2020. **Repetier-Firmware Configuration Tool V0.92.9**. https://www.repetier.com/firmware/v092/ (retrieved September 3, 2020)

143. Larry L. Howell. 2013. *Handbook of Compliant Mechanisms*. John Wiley & Sons Ltd, Oxford, UK. https://doi.org/10.1002/9781118516485.ch1

144. J.Y. Hu, Y.I. Li, and K.W. Yeung. 2006. **Mechanical Tactile Properties**. In *Clothing Biosensory Engineering*, Y. Li and A.S.W. Wong (eds.). Woodhead Publishing, Online, 261–284. https://www.sciencedirect.com/science/article/pii/B9781855739253500162

145. Kunpeng Huang, Ruojia Sun, Ximeng Zhang, Md. Tahmidul Islam Molla, Margaret Dunne, Francois Guimbretiere, and Cindy Hsin-Liu Kao. 2021. **WovenProbe: Probing Possibilities for Weaving Fully-Integrated On-Skin Systems Deployable in the Field**. In *Designing Interactive Systems Conference 2021*, 1143–1158. https://doi.org/10.1145/3461778.3462105

146. Scott E. Hudson. 2014. **Printing Teddy Bears: A Technique for 3D Printing of Soft Interactive Objects**. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, 459–468. https://doi.org/10.1145/2556288.2557338

147. Scott E. Hudson and Jennifer Mankoff. 2006. **Rapid Construction of Functioning Physical Interfaces from Cardboard, Thumbtacks, Tin Foil and Masking Tape**. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, 289–298. https://doi.org/10.1145/1166253.1166299

148. Yuki Igarashi and Takeo Igarashi. 2009. **Designing Plush Toys with a Computer**. *Communications of the ACM* 52, 12: 81–88. https://doi.org/10.1145/1610252.1610275

149. Yuki Igarashi, Takeo Igarashi, and Jun Mitani. 2012. **Beady: Interactive Beadwork Design and Construction**. *ACM Transactions on Graphics* 31, 4: 1–9. https://doi.org/10.1145/2185520.2185545

150. Yuki Igarashi, Takeo Igarashi, and Hiromasa Suzuki. 2008. **Knitting a 3D Model**. *Computer Graphics Forum* 27, 7: 1737–1743. https://doi.org/10.1111/j.1467-8659.2008.01318.x

151. Yuki Igarashi, Takeo Igarashi, and Hiromasa Suzuki. 2008. **Knitty: 3D Modeling of Knitted Animals with a Production Assistant Interface.** In *Eurographics (Short Papers)*, 17–20.

152. Miwa Ikemiya and Daniela K. Rosner. 2014. **Broken Probes: Toward the Design of Worn Media**. *Personal and Ubiquitous Computing* 18, 3: 671–683. https://doi.org/10.1007/s00779-013-0690-y

153. Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. 2016. **Metamaterial Mechanisms**. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 529–539. https://doi.org/10.1145/2984511.2984540

154. Alexandra Ion, Robert Kovacs, Oliver S. Schneider, Pedro Lopes, and Patrick Baudisch. 2018. **Metamaterial Textures**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3173574.3173910

155. Alexandra Ion, David Lindlbauer, Philipp Herholz, Marc Alexa, and Patrick Baudisch. 2019. **Understanding Metamaterial Mechanisms**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 1–14. https://doi.org/10.1145/3290605.3300877

156. Alexandra Ion, Ludwig Wall, Robert Kovacs, and Patrick Baudisch. 2017. **Digital Mechanical Metamaterials**. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 977–988. https://doi.org/10.1145/3025453.3025624

157. Kenneth E. Iverson. 1980. **Notation as a Tool of Thought**. *Communications of the ACM* 23, 8: 444–465. https://doi.org/10.1145/358896.358899

158. Jennifer Jacobs. 2013. **Algorithmic craft : the synthesis of computational design, digital fabrication, and hand craft**. Thesis: Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/91843 (retrieved October 30, 2020)

159. Jennifer Jacobs and Amit Zoran. 2015. **Hybrid Practice in the Kalahari: Design Collaboration Through Digital Tools and Hunter-Gatherer Craft**. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 619–628. https://doi.org/10.1145/2702123.2702362

160. Johnny Russell. 2020. **RAMPS 1.4**. *RepRap*. https://reprap.org/wiki/RAMPS_1.4 (retrieved September 3, 2020)

161. JolieTricotrice. 2014. **TIL the World's First Programmable Computer, the Jacquard Loom, Was Built in 1801**. *r/todayilearned*. https://www.reddit.com/r/todayilearned/comments/2jo308/til_the_worlds_first_programmable_computer_the/ (retrieved August 2, 2024)

162. Benjamin Jones, Yuxuan Mei, Haisen Zhao, Taylor Gotfrid, Jennifer Mankoff, and Adriana Schulz. 2022. **Computational Design of Knit Templates**. *ACM Transactions on Graphics* 41, 2: 1–16. https://doi.org/10.1145/3488006

163. Lee Jones and Audrey Girouard. 2022. **Learning with Stitch Samplers: Exploring Stitch Samplers as Contextual Instructions for E-Textile Tutorials**. In *Designing Interactive Systems Conference*, 949–965. https://doi.org/10.1145/3532106.3533488

164. Lee Jones and Sara Nabil. 2022. **Goldwork Embroidery: Interviews with Practitioners on Working with Metal Threads and Opportunities for E-Textile Hybrid Crafts**. In *Creativity and Cognition*, 364–379. https://doi.org/10.1145/3527927.3532809

165. Yasmin B. Kafai, Kristin Searle, Eliot Kaplan, Deborah Fields, Eunkyoung Lee, and Debora Lui. 2013. **Cupcake Cushions, Scooby Doo Shirts, and Soft Boomboxes: E-Textiles in High School to Promote Computational Concepts, Practices, and Perceptions**. In *Proceeding of the 44th ACM technical symposium on Computer science education*, 311–316. https://doi.org/10.1145/2445196.2445291

166. Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2008. **Simulating Knitted Cloth at the Yarn Level**. *ACM Transactions on Graphics* 27, 3: 1–9. https://doi.org/10.1145/1360612.1360664

167. Viirj Kan, Emma Vargo, Noa Machover, Hiroshi Ishii, Serena Pan, Weixuan Chen, and Yasuaki Kakehi. 2017. **Organic Primitives: Synthesis and Design of pH-Reactive Materials Using Molecular I/O for Sensing, Actuation, and Interaction**. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 989–1000. https://doi.org/10.1145/3025453.3025952

168. Laewoo (Leo) Kang, Steven J. Jackson, and Phoebe Sengers. 2018. **Intermodulation: Improvisation and Collaborative Art Practice for HCI**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3173574.3173734

169. Isaac Karth. 2019. **Preliminary Poetics of Procedural Generation in Games**. *Transactions of the Digital Games Research Association* 4, 3. https://doi.org/10.26503/todigra.v4i3.106

170. Alexandre Kaspar. 2022. **Garment Design Workflows for On-Demand Machine Knitting**. PhD Thesis: Massachusetts Institute of Technology, Boston, MA, USA.

171. Alexandre Kaspar, Liane Makatura, and Wojciech Matusik. 2019. **Knitting Skeletons: A Computer-Aided Design Tool for Shaping and Patterning of Knitted Garments**. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 53–65. https://doi.org/10.1145/3332165.3347879

172. Alexandre Kaspar, Tae-Hyun Oh, Liane Makatura, Petr Kellnhofer, and Wojciech Matusik. 2019. **Neural Inverse Knitting: From Images to Manufacturing Instructions**. In *Proceedings of the 36th International Conference on Machine Learning*, 3272–3281. https://proceedings.mlr.press/v97/kaspar19a.html (retrieved December 12, 2022)

173. Alexandre Kaspar, Kui Wu, Yiyue Luo, Liane Makatura, and Wojciech Matusik. 2021. **Knit Sketching: From Cut & Sew Patterns to Machine-Knit Garments**. *ACM Transactions on Graphics* 40, 4: 63:1–63:15. https://doi.org/10.1145/3450626.3459752

174. Ozgun Kilic Afsar, Ali Shtarbanov, Hila Mor, Ken Nakagaki, Jack Forman, Karen Modrei, Seung Hee Jeong, Klas Hjort, Kristina Höök, and Hiroshi Ishii. 2021. **OmniFiber: Integrated Fluidic Fiber Actuators for Weaving Movement Based Interactions into the "Fabric of Everyday Life"**. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 1010–1026. https://doi.org/10.1145/3472749.3474802

175. Jeeeun Kim, Anhong Guo, Tom Yeh, Scott E. Hudson, and Jennifer Mankoff. 2017. **Understanding Uncertainty in Measurement and Accommodating Its Impact in 3D Modeling and Printing**. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, 1067–1078. https://doi.org/10.1145/3064663.3064690

176. Jeeeun Kim, Haruki Takahashi, Homei Miyashita, Michelle Annett, and Tom Yeh. 2017. **Machines as Co-Designers: A Fiction on the Future of Human-Fabrication Machine Interaction**. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 790–805. https://doi.org/10.1145/3027063.3052763

177. Jeeeun Kim, Clement Zheng, Haruki Takahashi, Mark D Gross, Daniel Ashbrook, and Tom Yeh. 2018. **Compositional 3D Printing: Expanding & Supporting Workflows Towards Continuous Fabrication**. In *Proceedings of the 2nd ACM Symposium on Computational Fabrication*, 1–10. https://doi.org/10.1145/3213512.3213518

178. Jin Hee (Heather) Kim, Kunpeng Huang, Simone White, Melissa Conroy, and Cindy Hsin-Liu Kao. 2021. **KnitDermis: Fabricating Tactile On-Body Interfaces Through Machine Knitting**. In *Designing Interactive Systems Conference 2021*, 1183–1200. https://doi.org/10.1145/3461778.3462007

179. Jin Hee (Heather) Kim, Shreyas Dilip Patil, Sarina Matson, Melissa Conroy, and Cindy Hsin-Liu Kao. 2022. **KnitSkin: Machine-Knitted Scaled Skin for Locomotion**. In *CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3491102.3502142

180. David Kindy. 2019. **How Susan Kare Designed User-Friendly Icons for the First Macintosh**. *Smithsonian*. https://www.smithsonianmag.com/innovation/how-susan-kare-designed-user-friendly-icons-for-first-macintosh-180973286/

181. Nahum Kiryati, Yuval Eldar, and Alfred M Bruckstein. 1991. **A Probabilistic Hough Transform**. *Pattern Recognition* 24, 4: 303–316.

182. Kobakant. 2010. **Knit Stroke Sensor**. https://www.kobakant.at/DIY/?p=4317 (retrieved February 27, 2020)

183. Tomomi Kono and Keita Watanabe. 2017. **Filum: A Sewing Technique to Alter Textile Shapes**. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, 39–41. https://doi.org/10.1145/3131785.3131797

184. Max Kreminski and Michael Mateas. 2021. **Toward Narrative Instruments**. In *14th International Conference on Interactive Digital Storytelling*, 499–508. https://doi.org/10.1007/978-3-030-92300-6_50

185. Max Kreminski and Michael Mateas. 2021. **Reflective Creators**. In *Proceedings of the 12th International Conference on Computational Creativity*, 10.

186. Max Kreminski and Noah Wardrip-Fruin. 2018. **Gardening Games: An Alternative Philosophy of PCG in Games**. In *PCG Workshop*.

187. Kris Basta. 2019. **Needle Beetle Needle Selector for LK Machines**. *Kriskrafter*. https://www.kriskrafter.com/product-page/needle-beetle-needle-selector (retrieved September 13, 2022)

188. Pin-Sung Ku, Kunpeng Huang, and Cindy Hsin-Liu Kao. 2022. **Patch-O: Deformable Woven Patches for On-Body Actuation**. In *CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3491102.3517633

189. Stacey Kuznetsov and Eric Paulos. 2010. **Rise of the Expert Amateur: DIY Projects, Communities, and Cultures**. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries*, 295. https://doi.org/10.1145/1868914.1868950

190. Sophie Landwehr Sydow, Martin Jonsson, and Jakob Tholander. 2022. **Modding the Pliable Machine: Unpacking the Creative and Social Practice of Upkeep at the Makerspace**. In *Creativity and Cognition*, 220–233. https://doi.org/10.1145/3527927.3532804

191. Gierad Laput, Walter S. Lasecki, Jason Wiese, Robert Xiao, Jeffrey P. Bigham, and Chris Harrison. 2015. **Zensors: Adaptive, Rapidly Deployable, Human-Intelligent Sensor Feeds**. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 1935–1944. https://doi.org/10.1145/2702123.2702416

192. Gierad Laput, Xiang 'Anthony' Chen, and Chris Harrison. 2015. **3D Printed Hair: Fused Deposition Modeling of Soft Strands, Fibers, and Bristles**. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, 593–597. https://doi.org/10.1145/2807442.2807484

193. Jill H. Larkin and Herbert A. Simon. 1987. **Why a Diagram Is (Sometimes) Worth Ten Thousand Words**. *Cognitive Science* 11, 1: 65–100. https://doi.org/10.1111/j.1551-6708.1987.tb00863.x

194. Maria Larsson, Hironori Yoshida, Nobuyuki Umetani, and Takeo Igarashi. 2020. **Tsugite: Interactive Design and Fabrication of Wood Joints**. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 317–327. https://doi.org/10.1145/3379337.3415899

195. Maria Larsson, Hironori Yoshida, and Takeo Igarashi. 2019. **Human-in-the-Loop Fabrication of 3D Surfaces with Natural Tree Branches**. In *Proceedings of the ACM Symposium on Computational Fabrication*, 1–12. https://doi.org/10.1145/3328939.3329000

196. Yi-Chin Lee. 2023. **Digital Tufting Bee: Expanding Computational Design Boundaries Through Collective Material Practice and Social Play**. *Digital Creativity*: 1–19. https://doi.org/10.1080/14626268.2023.2205389

197. Yi-Chin Lee and Lea Albaugh. 2021. **Hybrid Embroidery Games: Playing with Materials, Machines, and People**. In *Designing Interactive Systems Conference 2021*, 749–762. https://doi.org/10.1145/3461778.3462019

198. Yi-Chin Lee and Daniel Cardoso Llach. 2020. **Hybrid Embroidery: Exploring Interactive Fabrication in Handcrafts**. In *ACM SIGGRAPH 2020 Art Gallery*, 429–433. https://doi.org/10.1145/3386567.3388575

199. Danny Leen, Tom Veuskens, Kris Luyten, and Raf Ramakers. 2019. **JigFab: Computational Fabrication of Constraints to Facilitate Woodworking with Power Tools**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3290605.3300386

200. Golan Levin. 2005–2015. *An Informal Catalogue of Slit-Scan Video Artworks and Research*. http://www.flong.com/texts/lists/slit_scan/

201. Jingyi Li, Eric Rawn, Jacob Ritchie, Jasper Tran O'Leary, and Sean Follmer. 2023. **Beyond the Artifact: Power as a Lens for Creativity Support Tools**. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–15. https://doi.org/10.1145/3586183.3606831

202. Min Li, Shaokai Wang, Zuoguang Zhang, and Boming Wu. 2009. **Effect of Structure on the Mechanical Behaviors of Three-Dimensional Spacer Fabric Composites**. *Applied Composite Materials* 16, 1: 1–14. https://doi.org/10.1007/s10443-008-9072-4

203. Zefang Li, Gina Olson, Dinesh K. Patel, Lining Yao, and Carmel Majidi. 2023. **Electrically Controlled Liquid Crystal Elastomer Surfaces for Dynamic Wrinkling**. *Advanced Intelligent Systems* 6, 2: 2200402. https://doi.org/10.1002/aisy.202200402

204. LIA. 2014. **Filament Sculptures**. http://www.liaworks.com/theprojects/filament-sculptures/

205. Jenny Lin and James McCann. 2021. **An Artin Braid Group Representation of Knitting Machine State with Applications to Validation and Optimization of Fabrication Plans**. In *2021 IEEE International Conference on Robotics and Automation*, 1147–1153. https://doi.org/10.1109/ICRA48506.2021.9562113

206. Jenny Lin, Vidya Narayanan, Yuka Ikarashi, Jonathan Ragan-Kelley, Gilbert Bernstein, and James Mccann. 2023. **Semantics and Scheduling for Machine Knitting Compilers**. *ACM Transactions on Graphics* 42, 4: 1–26. https://doi.org/10.1145/3592449

207. Jenny Lin, Vidya Narayanan, and James McCann. 2018. **Efficient Transfer Planning for Flat Knitting**. In *Proceedings of the 2nd Annual ACM Symposium on Computational Fabrication*, 1–7. https://doi.org/10.1145/3213512.3213515

208. Pamela Liou. 2015. **Doti the Desktop Loom**. https://pamelaliou.com/doti.html (retrieved December 29, 2020)

209. Ani Liu. 2017. *Mind in the Machine: Psyche in the Age of Mechanical Production*. https://ani-liu.com/mind-in-machine

210. Szu-Yu (Cyn) Liu, Jeffrey Bardzell, and Shaowen Bardzell. 2019. **Decomposition as Design: Co-Creating (with) Natureculture**. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, 605–614. https://doi.org/10.1145/3294109.3295653

211. Yanping Liu and Hong Hu. 2010. **Sound Absorption Behavior of Knitted Spacer Fabrics**. *Textile Research Journal* 80, 18: 1949–1957. https://doi.org/10.1177/0040517510373639

212. Yanping Liu and Hong Hu. 2011. **Compression Property and Air Permeability of Weft-knitted Spacer Fabrics**. *The Journal of The Textile Institute* 102, 4: 366–372. https://doi.org/10.1080/00405001003771200

213. Zishun Liu, Xingjian Han, Yuchen Zhang, Xiangjia Chen, Yu-Kun Lai, Eugeni L. Doubrovski, Emily Whiting, and Charlie C. L. Wang. 2021. **Knitting 4D Garments with Elasticity Controlled for Body Motion**. *ACM Transactions on Graphics* 40, 4: 62:1–62:16. https://doi.org/10.1145/3450626.3459868

214. Marco Livesu, Stefano Ellero, Jonàs Martínez, Sylvain Lefebvre, and Marco Attene. 2017. **From 3D Models to 3D Prints: An Overview of the Processing Pipeline**. *Computer Graphics Forum* 36, 2: 537–564. https://doi.org/10.1111/cgf.13147

215. Daniel Cardoso Llach. 2015. **Software Comes to Matter: Toward a Material History of Computational Design**. *Design Issues* 31, 3: 41–54. https://doi.org/10.1162/DESI_a_00337

216. Jesse Louis-Rosenberg, Alec Resnick, and Jessica Rosenkrantz. 2012. **Multiphase Numerical Modeling of Dendritic Solidification for Jigsaw Puzzle Generation**. https://n-e-r-v-o-u-s.com/education/papers/nervousSystem_2012-jigsaw-puzzle-paper.pdf (retrieved September 18, 2024)

217. Ada Augusta Lovelace. 1843. **Notes by AAL [Augusta Ada Lovelace]**. *Taylor's Scientific Memoirs* 3: 666–731.

218. Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, and others. 2019. **Mediapipe: A Framework for Building Perception Pipelines**. https://doi.org/10.48550/arXiv.1906.08172

219. Danli Luo, Aditi Maheshwari, Andreea Danielescu, Jiaji Li, Yue Yang, Ye Tao, Lingyun Sun, Dinesh K. Patel, Guanyun Wang, Shu Yang, Teng Zhang, and Lining Yao. 2023. **Autonomous Self-Burying Seed Carriers for Aerial Seeding**. *Nature* 614, 7948: 463–470. https://doi.org/10.1038/s41586-022-05656-3

220. Michal Luria. 2018. **Designing Robot Personality Based on Fictional Sidekick Characters**. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 307–308. https://doi.org/10.1145/3173386.3176912

221. Dor Ma'ayan, Wode Ni, Katherine Ye, Chinmay Kulkarni, and Joshua Sunshine. 2020. **How Domain Experts Create Conceptual Diagrams and Implications for Tool Design**. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3313831.3376253

222. Hua Ma and Junichi Yamaoka. 2022. **SenSequins: Smart Textile Using 3D Printed Conductive Sequins**. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 1–13. https://doi.org/10.1145/3526113.3545688

223. Shiran Magrisso, Moran Mizrahi, and Amit Zoran. 2018. **Digital Joinery For Hybrid Carpentry**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–11. https://doi.org/10.1145/3173574.3173741

224. Peter Manning. 2013. *Electronic and Computer Music*. Oxford University Press, USA, Oxford, UK.

225. Hugh S. Manon and Daniel Temkin. 2011. **Notes on Glitch**. *World Picture* 6: 118. http://worldpicturejournal.com/article/notes-on-glitch/

226. Trevor HJ Marchand. 2008. **Muscles, Morals and Mind: Craft Apprenticeship and the Formation of Person**. *British Journal of Educational Studies* 56, 3: 245–271. https://doi.org/10.1111/j.1467-8527.2008.00407.x

227. Nancy Marchant. 2010. *Knitting Brioche: The Essential Guide to the Brioche Stitch*. Penguin Publishing Group, New York, NY, USA.

228. Nancy Marchant. 2014. **2-Color Brioche Stitch**. *Brioche Stitch*. http://www.briochestitch.com/archives/twocolorbrioche/2-color-brioche-stitch (retrieved August 21, 2021)

229. Nancy Marchant and Alexandra Feo. 2016. *Leafy brioche: knitted cowls and scarfs*. Nancy Marchant / Schoolhouse Press, Pittsville, WI.

230. Margaret Wertheim, David Henderson, and Daina Taimina. 2004. **Crocheting the Hyperbolic Plane: An Interview with David Henderson and Daina Taimina**. *Cabinet Magazine*. https://cabinetmagazine.org/issues/16/wertheim_henderson_taimina.php (retrieved December 31, 2022)

231. David N.G. McCallum. 2018. **Glitching the Fabric: Strategies of New Media Art Applied to the Codes of Knitting and Weaving**. PhD Thesis: University of Gothenburg. http://rgdoi.net/10.13140/RG.2.2.13674.98242 (retrieved May 18, 2020)

232. James McCann. 2017. **The "Knitout" (.k) File Format V0.5.3**. https://textiles-lab.github.io/knitout/knitout.html (retrieved April 3, 2018)

233. James McCann, Lea Albaugh, Vidya Narayanan, April Grow, Wojciech Matusik, Jennifer Mankoff, and Jessica Hodgins. 2016. **A Compiler for 3D Machine Knitting**. *ACM Transactions on Graphics* 35, 4: 49:1–49:11. https://doi.org/10.1145/2897824.2925940

234. James McCann and Vidya Narayanan. 2017–2020. *Autoknit: A Public-Domain Reimplementation of "Automatic Machine Knitting of 3D Meshes"*. https://github.com/textiles-lab/autoknit (retrieved April 5, 2023)

235. Alex McLean, Ellen Harlizius-Klück, and Janis Jefferies. 2017. **Introduction: Weaving Codes, Coding Weaves**. *TEXTILE* 15, 2: 118–123. https://doi.org/10.1080/14759756.2017.1298232

236. Gianluca Memoli, Letizia Chisari, Jonathan P. Eccles, Mihai Caleap, Bruce W. Drinkwater, and Sriram Subramanian. 2019. **VARI-SOUND: A Varifocal Lens for Sound**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3290605.3300713

237. Rosa Menkman. 2010. *The Glitch Moment(um)*. Institute of Network Cultures, Amsterdam.

238. Rosa Menkman. 2011. **Glitch Studies Manifesto**. In *Video Vortex Reader II: Moving Images Beyond YouTube*, Geert Lovink and Rachel Somers Miles (eds.). Institute of Network Cultures, Amsterdam, 336–347. https://networkcultures.org/videovortex/vv-reader/

239. Michelle Lock. 2005. **Taking Crochet to a Higher Plane**. *Interweave Crochet*. https://theiff.org/press/interweave.html (retrieved December 31, 2022)

240. Acme Mills. 2020. **3d Spacer Fabrics ¦ Acme Mills**. https://acmemills.com/brands/3-d-spacer-fabrics

241. Abby Millward. 2014. *Chart Minder*. https://www.chart-minder.com/about (retrieved December 30, 2022)

242. Jarkko Moilanen, Angela Daly, Ramon Lobato, and Darcy Allen. 2014. **Cultures of Sharing in 3D Printing: What Can We Learn from the Licence Choices of Thingiverse Users?** *Journal of Peer Production*, 6. http://peerproduction.net/issues/issue-6-disruption-and-the-law/peer-reviewed-articles/cultures-of-sharing-in-thingiverse-what-can-we-learn-from-the-licence-choices-of-thingiverse-users/ (retrieved September 18, 2024)

243. Vera Molnar. 1975. **Toward Aesthetic Guidelines for Paintings with the Aid of a Computer**. *Leonardo* 8, 3: 185. https://doi.org/10.2307/1573236

244. Nicholas de Monchaux. 2011. *Spacesuit: Fashioning Apollo*. MIT press.

245. Alexander Monea. 2019. **Race and Computer Vision**. In *The Democratization of Artificial Intelligence: Net Politics in the Era of Learning Algorithms* (1st ed.), Andreas Sudmann (ed.). transcript Verlag, Bielefeld, Germany. https://doi.org/10.14361/9783839447192

246. Jennifer Moore. 2010. *The Weaver's Studio: Doubleweave*. Interweave, USA.

247. Hedieh Moradi, Long N Nguyen, Quyen-Anh Valentina Nguyen, and Cesar Torres. 2022. **Glaze Epochs: Understanding Lifelong Material Relationships Within Ceramics Studios**. In *Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction*, 1–13. https://doi.org/10.1145/3490149.3501310

248. Ilan E Moyer, Samuelle Bourgault, Devon Frost, and Jennifer Jacobs. 2024. **Throwing Out Conventions: Reimagining Craft-Centered CNC Tool Design Through the Digital Pottery Wheel**. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–22. https://doi.org/10.1145/3613904.3642361

249. Ilan Moyer. 2013. **24-Bit Friendship Loom**. *Ilan Moyer's Online Portfolio*. http://web.mit.edu/imoyer/www/portfolio/jacquard/index.html (retrieved December 29, 2020)

250. Mr.doob. 2021. *Three.js*. https://github.com/mrdoob/three.js (retrieved September 3, 2021)

251. Stefanie Mueller. 2016. **Interacting with Personal Fabrication Devices**. PhD Thesis: Universität Potsdam.

252. Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. 2014. **WirePrint: 3D Printed Previews for Fast Prototyping**. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 273–280. https://doi.org/10.1145/2642918.2647359

253. Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. 2012. **Interactive Construction: Interactive Fabrication of Functional Mechanical Devices**. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 599. https://doi.org/10.1145/2380116.2380191

254. Stefanie Mueller, Pedro Lopes, Konstantin Kaefer, Bastian Kruck, and Patrick Baudisch. 2013. **Constructable: Interactive Construction of Functional Mechanical Devices**. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, 3107–3110. https://doi.org/10.1145/2468356.2479622

255. Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. 2014. **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*, 3827–3834. https://doi.org/10.1145/2556288.2557005

256. Stefanie Mueller, Anna Seufert, Huaishu Peng, Robert Kovacs, Kevin Reuss, François Guimbretière, and Patrick Baudisch. 2019. **FormFab: Continuous Interactive Fabrication**. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, 315–323. https://doi.org/10.1145/3294109.3295620

257. Multiple. 2024. **Search Thingiverse**. *Thingiverse - Digital Designs for Physical Objects*. https://www.thingiverse.com/search?q=filament+reel

258. Georges Nader, Yu Han Quek, Pei Zhi Chia, Oliver Weeger, and Sai-Kit Yeung. 2021. **KnitKit: A Flexible System for Machine Knitting of Customizable Textiles**. *ACM Transactions on Graphics* 40, 4: 1–16. https://doi.org/10.1145/3450626.3459790

259. Satoshi Nakamaru, Ryosuke Nakayama, Ryuma Niiyama, and Yasuaki Kakehi. 2017. **FoamSense: Design of Three Dimensional Soft Sensors with Porous Materials**. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 437–447. https://doi.org/10.1145/3126594.3126666

260. Ryosuke Nakayama, Ryo Suzuki, Satoshi Nakamaru, Ryuma Niiyama, Yoshihiro Kawahara, and Yasuaki Kakehi. 2019. **MorphIO: Entirely Soft Sensing and Actuation Modules for Programming Shape Changes Through Tangible Interaction**. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, 975–986. https://doi.org/10.1145/3322276.3322337

261. Vidya Narayanan. 2022. **Foundations for 3D Machine Knitting**. thesis: Carnegie Mellon University. https://doi.org/10.1184/R1/19658928.v1

262. Vidya Narayanan, Lea Albaugh, Jessica Hodgins, Stelian Coros, and James Mccann. 2018. **Automatic Machine Knitting of 3D Meshes**. *ACM Transactions on Graphics* 37, 3: 1–15. https://doi.org/10.1145/3186265

263. Vidya Narayanan, Kui Wu, Cem Yuksel, and James McCann. 2019. **Visual Knitting Machine Programming**. *ACM Transactions on Graphics* 38, 4: 63:1–63:13. https://doi.org/10.1145/3306346.3322995

264. Harold G. Nelson and Erik Stolterman. 2012. *The Design Way: Intentional Change in an Unpredictable World*. The MIT Press. https://doi.org/10.7551/mitpress/9188.001.0001

265. Carman Neustaedter and Phoebe Sengers. 2012. **Autobiographical Design in HCI Research: Designing and Learning Through Use-It-Yourself**. In *Proceedings of the Designing Interactive Systems Conference*, 514–523. https://doi.org/10.1145/2317956.2318034

266. Jerry Nicholls. 2018. **Weav3r LEGO Loom Weaving a Scanned Pattern, Unloaded.** *YouTube*. https://www.youtube.com/watch?v=7pjrM-yqhVA (retrieved September 3, 2020)

267. Nicky Case. 2018. **The Wisdom and/or Madness of Crowds**. http://ncase.me/crowds/ (retrieved December 31, 2022)

268. Kristina Niedderer and Katherine Townsend. 2014. **Designing Craft Research: Joining Emotion and Knowledge**. *The Design Journal* 17, 4: 624–647. https://doi.org/10.2752/175630614X14056185480221

269. Michael Nitsche and Anna Weisling. 2019. **When Is It Not Craft?: Materiality and Mediation When Craft and Computing Meet**. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, 683–689. https://doi.org/10.1145/3294109.3295651

270. Vernelle A.A. Noel. 2015. **The Bailey-Derek Grammar: Recording the Craft of Wire-Bending in the Trinidad Carnival**. *Leonardo* 48, 4: 357–365. https://doi.org/10.1162/LEON_a_01089

271. Vernelle A.A. Noel. 2022. **Computational Regionalism: De-Familiarization of Tectonics in the Wire-Bending Craft**. *International Journal of Architectural Computing* 20, 2: 277–296. https://doi.org/10.1177/14780771221097685

272. Paul R. O'Connor. 2006. **Double Weave - A Retrospective**. In *Complex Weavers Seminars*, 34. https://www2.cs.arizona.edu/patterns/weaving/topic_doubleweave.html

273. Gerard O'Regan. 2018. **Hollerith's Tabulating Machines and the Birth of IBM**. In *The Innovation in Computing Companion: A Compendium of Select, Pivotal Inventions*. Springer Cham, 151–153. https://doi.org/10.1007/978-3-030-02619-6_31

274. Alex Olwal, Jon Moeller, Greg Priest-Dorman, Thad Starner, and Ben Carroll. 2018. **I/O Braid: Scalable Touch-Sensitive Lighted Cords Using Spiraling, Repeating Sensing Textiles and Fiber Optics**. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, 485–497. https://doi.org/10.1145/3242587.3242638

275. Emel Önder and Ömer Berk Berkalp. 2006. **Jet Weaving**. https://web.itu.edu.tr/~berkalpo/Weaving_Lecture/Weaving_Chapter8_Airjet.pdf (retrieved August 11, 2024)

276. Yoko Ono. 1964. *Grapefruit*. Wunternaum, Tokyo Japan.

277. Doenja Oogjes and Ron Wakkary. 2022. **Weaving Stories: Toward Repertoires for Designing Things**. https://summit.sfu.ca/item/22028 (retrieved April 15, 2022)

278. Daniëlle Ooms, Nick Voskuil, Kristina Andersen, and Hanna Ottilia Wallner. 2020. **Ruta, a Loom for Making Sense of Industrial Weaving**. In *Companion Publication of the 2020 ACM Designing Interactive Systems Conference*, 337–340. https://doi.org/10.1145/3393914.3395815

279. OpenJS Foundation, Node Technical Steering Committee, and Node Collaborators. 2009–2024. *Node.js*. https://nodejs.org/en

280. Kate Orton-Johnson. 2014. **Knit, Purl and Upload: New Technologies, Digital Mediations and the Experience of Leisure**. *Leisure Studies* 33, 3: 305–321. https://doi.org/10.1080/02614367.2012.723730

281. Windell H. Oskay and Lenore M. Edman. 2007. **Bristlebot: A Tiny Directional Vibrobot**. https://www.evilmadscientist.com/2007/bristlebot-a-tiny-directional-vibrobot/

282. Jifei Ou, Gershon Dublon, Chin-Yi Cheng, Felix Heibeck, Karl Willis, and Hiroshi Ishii. 2016. **Cilllia: 3D Printed Micro-Pillar Structures for Surface Texture, Actuation and Sensing**. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 5753–5764. https://doi.org/10.1145/2858036.2858257

283. Jifei Ou, Daniel Oran, Don Derek Haddad, Joseph Paradiso, and Hiroshi Ishii. 2019. **SensorKnit: Architecting Textile Sensors with Machine Knitting**. *3D Printing and Additive Manufacturing* 6, 1: 1–11. https://doi.org/10.1089/3dp.2018.0122

284. Jifei Ou, Mélina Skouras, Nikolaos Vlavianos, Felix Heibeck, Chin-Yi Cheng, Jannik Peters, and Hiroshi Ishii. 2016. **aeroMorph - Heat-Sealing Inflatable Shape-Change Materials for Interaction Design**. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 121–132. https://doi.org/10.1145/2984511.2984520

285. Robert Owen, Colin Sherborne, Richard Evans, Gwendolen C. Reilly, and Frederik Claeyssens. 2020. **Combined Porogen Leaching and Emulsion Templating to Produce Bone Tissue Engineering Scaffolds**. *International Journal of Bioprinting* 6, 2: 265. https://doi.org/10.18063/ijb.v6i2.265

286. Oxford University Press. 1891–2022. **Complex, Adj.** *OED Online*. http://www.oed.com/view/Entry/37672 (retrieved November 17, 2022)

287. Neri Oxman. 2007. **Digital Craft: Fabrication Based Design in the Age of Digital Production**. In *Workshop proceedings for UBICOMP 2007*, 534–538. https://web.archive.org/web/20090415180525/http://ambient.media.mit.edu/transitive/ubicomp07.htm

288. Neri Oxman. 2010. **Structuring Materiality: Design Fabrication of Heterogeneous Materials**. *Architectural Design* 80, 4: 78–85. https://doi.org/10.1002/ad.1110

289. Ryuta Ozawa, Hiroaki Kobayashi, and Kazunori Hashirii. 2014. **Analysis, Classification, and Design of Tendon-Driven Mechanisms**. *IEEE Transactions on Robotics* 30, 2: 396–410. https://doi.org/10.1109/TRO.2013.2287976

290. Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. **Elastic Textures for Additive Fabrication**. *ACM Transactions on Graphics* 34, 4: 135:1–135:12. https://doi.org/10.1145/2766937

291. Naomi Parkhurst. 2018. **About My Stitches Based on Words**. *String Geekery*. https://www.gannetdesigns.com/2018/09/10/about-my-stitches-based-on-words/ (retrieved December 31, 2022)

292. Dinesh K. Patel, Ke Zhong, Haiqing Xu, Mohammad F. Islam, and Lining Yao. 2023. **Sustainable Morphing Matter: Design and Engineering Practices**. *Advanced Materials Technologies* 8, 23: 2300678. https://doi.org/10.1002/admt.202300678

293. Celia Pearce, Gillian Smith, Jeanie Choi, and Isabella Carlsson. 2016. **eBee: Merging Quilting, Electronics & Board Game Design**. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 3877–3880. https://doi.org/10.1145/2851581.2891099

294. Nadya Peek, James Coleman, Ilan Moyer, and Neil Gershenfeld. 2017. **Cardboard Machine Kit: Modules for the Rapid Prototyping of Rapid Prototyping Machines**. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 3657–3668. https://doi.org/10.1145/3025453.3025491 (retrieved January 26, 2021)

295. Nadya Peek and Ilan Moyer. 2017. **Popfab: A Case for Portable Digital Fabrication**. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, 325–329. https://doi.org/10.1145/3024969.3025009

296. Mark G. Peeters. 2014. **Super Flowers (Drooloop Flowers)**. *Thingiverse*. https://www.thingiverse.com/thing:240158 (retrieved January 25, 2022)

297. Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. 2018. **RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3173574.3174153

298. Huaishu Peng, Jennifer Mankoff, Scott E. Hudson, and James McCann. 2015. **A Layered Fabric 3D Printer for Soft Interactive Objects**. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 1789–1798. https://doi.org/10.1145/2702123.2702327

299. Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. 2016. **On-The-Fly Print: Incremental Printing While Modelling**. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 887–896. https://doi.org/10.1145/2858036.2858106

300. Ken Perlin. 1985. **An Image Synthesizer**. In *SIGGRAPH '85*, 287–296. https://doi.org/0-89791-166-0/85/007/0287

301. Hannah Perner-Wilson. 2011. **A Kit-of-No-Parts**. Massachusetts Institute of Technology.

302. Dušan Peterc and Simon Weilguny. 1992. *Arahweave*. Arahne. https://www.arahne.si/products/arahweave/ (retrieved December 29, 2022)

303. Andreas Pointner, Thomas Preindl, Sara Mlakar, Roland Aigner, and Michael Haller. 2020. **Knitted RESi: A Highly Flexible, Force-Sensitive Knitted Textile Based on Resistive Yarns**. In *ACM SIGGRAPH 2020 Emerging Technologies*, 1–2. https://doi.org/10.1145/3388534.3407292

304. Mariana Popescu, Matthias Rippmann, Tom Van Mele, and Philippe Block. 2018. **Automated Generation of Knit Patterns for Non-Developable Surfaces**. In *Humanizing Digital Reality*, Klaas De Rycke, Christoph Gengnagel, Olivier Baverel, Jane Burry, Caitlin Mueller, Minh Man Nguyen, Philippe Rahm and Mette Ramsgaard Thomsen (eds.). Springer Singapore, Singapore, 271–284. https://doi.org/10.1007/978-981-10-6611-5_24

305. Irene Posch and Ebru Kurbak. 2016. **CRAFTED LOGIC Towards Hand-Crafting a Computer**. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 3881–3884. https://doi.org/10.1145/2851581.2891101

306. Emanuel Anthony Posselt. 1887. *The Jacquard machine analyzed and explained: with an appendix on the preparation of Jacquard cards*. Pennsylvania Museum and School of Industrial Art, Philadelphia, PA. http://archive.org/details/jacquardmachinea00poss (retrieved January 8, 2021)

307. Ivan Poupyrev, Nan-Wei Gong, Shiho Fukuhara, Mustafa Emre Karagozler, Carsten Schwesig, and Karen E. Robinson. 2016. **Project Jacquard: Interactive Digital Textiles at Scale**. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 4216–4227. https://doi.org/10.1145/2858036.2858176

308. Thomas Preindl, Cedric Honnet, Andreas Pointner, Roland Aigner, Joseph A. Paradiso, and Michael Haller. 2020. **Sonoflex: Embroidered Speakers Without Permanent Magnets**. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, 8. https://doi.org/10.1145/3379337.3415888

309. Miller Puckette. 1996. *Pure Data*. https://puredata.info/ (retrieved January 1, 2023)

310. David Pye. 1978, c1968. *The nature and art of workmanship*. Cambridge University Press, Cambridge.

311. Jie Qi and Leah Buechley. 2014. **Sketching in Circuits: Designing and Building Electronics on Paper**. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1713–1722. https://doi.org/10.1145/2556288.2557391

312. Meghan M. Quirk. 2009. **Inclusion of Fabric Properties in the Design of Electronic Textiles**. Thesis: Virginia Tech. https://vtechworks.lib.vt.edu/handle/10919/36245 (retrieved April 14, 2022)

313. Andrew Quitmeyer and Hannah Perner-Wilson. 2015. **Wearable Studio Practice: Design Considerations for Digital Crafting in Harsh Environments**. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and International Symposium on Wearable Computers*, 1285–1293. https://doi.org/10.1145/2800835.2807926

314. Younès Rabii and Michael Cook. 2023. **Why Oatmeal Is Cheap: Kolmogorov Complexity and Procedural Generation**. In *Proceedings of the 18th International Conference on the Foundations of Digital Games*, 1–7. https://doi.org/10.1145/3582437.3582484

315. Ahmad Rafsanjani, Yuerou Zhang, Bangyuan Liu, Shmuel M. Rubinstein, and Katia Bertoldi. 2018. **Kirigami Skins Make a Simple Soft Actuator Crawl**. *Science Robotics* 3, 15. https://doi.org/10.1126/scirobotics.aar7555

316. Brian Randell. 1994. **The Origins of Computer Programming**. *IEEE Annals of the History of Computing* 16, 4: 6–14.

317. Anna Ridler. 2018. *Myriad (Tulips)*. https://annaridler.com/myriad-tulips

318. Michael L. Rivera and Scott E. Hudson. 2019. **Desktop Electrospinning: A Single Extruder 3D Printer for Producing Rigid Plastic and Electrospun Textiles**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3290605.3300434

319. Michael L. Rivera, Melissa Moukperian, Daniel Ashbrook, Jennifer Mankoff, and Scott E. Hudson. 2017. **Stretching the Bounds of 3D Printing with Embedded Textiles**. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 497–508. https://doi.org/10.1145/3025453.3025460

320. Veronica Roberts. 2012. **"Like a Musical Score": Variability and Multiplicity in Sol LeWitt's 1970s Wall Drawings**. *Modern & Contemporary Drawings* 50, 2: 193–210. https://www.jstor.org/stable/41703378

321. Julian Rohrhuber and David Griffiths. 2017. **Coding with Knots**. *TEXTILE* 15, 2: 142–157. https://doi.org/10.1080/14759756.2017.1298307

322. Jessica Rosenkrantz and Jesse Louis-Rosenberg. 2017. **Dress/Code Democratising Design Through Computation and Digital Fabrication**. *Architectural Design* 87, 6: 48–57. https://doi.org/10.1002/ad.2237

323. Daniela K. Rosner. 2012. **The Material Practices of Collaboration**. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work,* 1155–1164. https://doi.org/10.1145/2145204.2145375

324. Daniela K. Rosner. 2021. *Critical Fabulations: Reworking the Methods and Margins of Design*. The MIT Press, Cambridge, Massachusetts, USA. https://doi.org/10.7551/mitpress/11035.001.0001

325. Daniela K. Rosner and Kimiko Ryokai. 2009. **Reflections on Craft: Probing the Creative Process of Everyday Knitters**. In *Proceeding of the seventh ACM conference on Creativity and Cognition,* 195. https://doi.org/10.1145/1640233.1640264

326. Daniela K. Rosner, Samantha Shorey, Brock R. Craft, and Helen Remick. 2018. **Making Core Memory: Design Inquiry into Gendered Legacies of Engineering and Craftwork**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems,* 1–13. https://doi.org/10.1145/3173574.3174105

327. Thijs Roumen, Bastian Kruck, Tobias Dürschmid, Tobias Nack, and Patrick Baudisch. 2016. **Mobile Fabrication**. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology,* 3–14. https://doi.org/10.1145/2984511.2984586

328. Andreas Roye and Thomas Gries. 2007. **3-D Textiles for Advanced Cement Based Matrix Reinforcement**. *Journal of Industrial Textiles* 37, 2: 163–173. https://doi.org/10.1177/1528083707078136

329. Gerard Rubio. 2014. **Building the OpenKnit Machine**. *Instructables*. https://www.instructables.com/Building-the-Open-Knit-machine/ (retrieved September 13, 2022)

330. Gerard Rubio. 2017. **About**. *Kniterate | The Digital Knitting Machine*. https://www.kniterate.com/about/ (retrieved September 13, 2022)

331. Rundong Tian, Sarah Sterman, Ethan Chiou, Jeremy Warner, and Eric Paulos. 2018. **MatchSticks: Woodworking Through Improvisational Digital Fabrication**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems,* 1–12. https://doi.org/10.1145/3173574.3173723

332. Richard Rutt. 1987. *A history of hand knitting*. Interweave Press, Loveland, Colo.

333. Kimiko Ryokai, Stefan Marti, and Hiroshi Ishii. 2004. **I/O Brush: Drawing with Everyday Objects as Ink**. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 303–310. http://doi.org/10.1145/985692.985731 (retrieved August 21, 2021)

334. Vanessa Sanchez, Conor J. Walsh, and Robert J. Wood. 2021. **Textile Technology for Soft Robotic and Autonomous Garments**. *Advanced Functional Materials* 31, 6: 2008278. https://doi.org/https://doi.org/10.1002/adfm.202008278

335. Sandoz Software. 2001–2020. *WIF "n" Proof*. https://sandozsoftware.com/WnP/WnP.html

336. Kurt Schaefer. 2019. **Making a DIY Computer Controlled Loom**. https://retrotechjournal.com/2019/06/17/making-a-diy-computer-controlled-loom/ (retrieved September 3, 2020)

337. Alice Schlein. 1994. *Network Drafting: An Introduction*. Bridgewater Press, Greenville, SC, USA.

338. Alice Schlein and Bhakti Ziek. 2006. *The Woven Pixel: Designing for Jacquard and Dobby Looms Using Photoshop®*. Bridgewater Press, Greenville, SC, USA.

339. Tenley Schmida and Paolo Pedercini. 2018. **GlitchScarf**. *Molleindustria*. https://www.molleindustria.org/blog/glitchscarf/ (retrieved September 13, 2022)

340. Sherry Schofield-Tomschin and Mary A. Littrell. 2001. **Textile Handcraft Guild Participation: A Conduit to Successful Aging**. *Clothing and Textiles Research Journal* 19, 2: 41–51. https://doi.org/10.1177/0887302X0101900201

341. Eldon Schoop, Michelle Nguyen, Daniel Lim, Valkyrie Savage, Sean Follmer, and Björn Hartmann. 2016. **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*, 1607–1614. https://doi.org/10.1145/2851581.2892429

342. Daniel V. Schroeder. 2013. *A Lattice-Boltzmann Fluid Simulation in JavaScript, Using HTML5 Canvas for Graphics*. https://physics.weber.edu/schroeder/fluids/ (retrieved September 3, 2021)

343. Neal A. Schwarzberger. 2016. *Innova AutoPilot*. ABM International, Montgomery, TX, USA. https://innovalongarm.com/software/autopilot

344. Kristin A. Searle, Deborah A. Fields, Debora A. Lui, and Yasmin B. Kafai. 2014. **Diversifying High School Students' Views about Computing with Electronic Textiles**. In *Proceedings of the tenth annual conference on International computing education research*, 75–82. https://doi.org/10.1145/2632320.2632352

345. Joseph Seering, Saiph Savage, Michael Eagle, Joshua Churchin, Rachel Moeller, Jeffrey P. Bigham, and Jessica Hammer. 2017. **Audience Participation Games: Blurring the Line Between Player and Spectator**. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, 429–440. https://doi.org/10.1145/3064663.3064732

346. Phoebe Sengers, Kirsten Boehner, Shay David, and Joseph `Jofish'Kaye. 2005. **Reflective Design**. In *Proceedings of the 4th decennial conference on Critical computing: between sense and sensibility*, 49–58. https://doi.org/10.1145/1094562.1094569

347. Fabienne "fbz" Serriere. 2015–2020. **KnitYak**. https://knityak.com/pages/about-knityak

348. Ticha Sethapakdi, Laura Huang, Vivian Hsinyueh Chan, Lung-Pan Cheng, Fernando Fuzinatto Dall'Agnol, Mackenzie Leake, and Stefanie Mueller. 2023. **Polagons: Designing and Fabricating Polarized Light Mosaics with User-Defined Color-Changing Behaviors**. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3544548.3580639

349. Ticha Sethapakdi and James McCann. 2019. **Painting with CATS: Camera-Aided Texture Synthesis**. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–9. http://doi.org/10.1145/3290605.3300287 (retrieved August 21, 2021)

350. Shaper. 2012. *Shaper Origin*. https://www.shapertools.com/en-us/origin (retrieved August 5, 2024)

351. Samantha Sherer. 2018. **Glitch 3D Printed Ceramics**. https://3dpclay.ca/#/weite/

352. Ken Shirriff. 2019. **Software Woven into Wire: Core Rope and the Apollo Guidance Computer**. *Ken Shirriff's Blog*. https://www.righto.com/2019/07/software-woven-into-wire-core-rope-and.html (retrieved August 2, 2024)

353. Kevin Simler. 2019. **Going Critical**. *Explorable Explanations*. https://meltingasphalt.com/interactive/going-critical (retrieved December 31, 2022)

354. Robin Sloan. 2014. **Dancing the Flip-Flop**. https://www.robinsloan.com/notes/flip-flop/

355. Asim Smailagic and Daniel P. Siewiorek. 1993. **A Case Study in Embedded-System Design: The VuMan 2 Wearable Computer**. *IEEE Design & Test of Computers* 10, 3: 56–67. https://doi.org/10.1109/54.232473

356. Gillian Smith. 2015. **An Analog History of Procedural Content Generation**. In *Foundations of Digital Games 2015*, 6.

357. Gillian Smith. 2017. **Generative Design for Textiles: Opportunities and Challenges for Entertainment AI**. In *Proceedings of the Thirteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 7. https://dl.acm.org/doi/abs/10.5555/3505326.3505343

358. Irwin Sobel. 1968. **An Isotropic 3x3 Image Gradient Operator**. atStanford A.I. Project.

359. Olga Soffer. 2004. **Recovering Perishable Technologies Through Use Wear on Tools: Preliminary Evidence for Upper Paleolithic Weaving and Net Making**. *Current Anthropology* 45, 3: 407–413.

360. Paul Soldner. 1990. **American-Style Raku**. *Ceramic Review 124*. http://www.paulsoldner.com/essays/american-raku (retrieved January 2, 2023)

361. Samantha Speer, Ana P Garcia-Alonzo, Joey Huang, Nickolina Yankova, Carolyn Rosé, Kylie A Peppler, James Mccann, and Melisa Orta Martinez. 2023. **SPEERLoom: An Open-Source Loom Kit for Interdisciplinary Engagement in Math, Engineering, and Textiles**. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–15. https://doi.org/10.1145/3586183.3606724

362. David J. Spencer. 2001. *Knitting Technology: A Comprehensive Handbook and Practical Guide*. Woodhead, Cambridge, England.

363. Sarah Spencer. 2021. **Auto Changer**. *Heart of Pluto*. https://www.heartofpluto.co/autochanger (retrieved September 13, 2022)

364. Robert T. Spillane and Mike S. Kowalski. 1995. **Warp Knitted Textile Spacer Fabric, Method of Producing Same, and Products Produced Therefrom**. https://patents.google.com/patent/US5385036A/en (retrieved September 17, 2020)

365. Lesley Stanfield and Melody Griffiths. 2010. *The Essential Stitch Collection: A Creative Guide to the 300 Stitches Every Knitter Really Needs to Know*. Reader's Digest, Pleasantville, N.Y.

366. Jessica Stanley, John A. Hunt, Phil Kunovski, and Yang Wei. 2021. **A Review of Connectors and Joining Technologies for Electronic Textiles**. *Engineering Reports* 4, 6: e12491. https://doi.org/10.1002/eng2.12491

367. Phillip David Stearns. 2012–2019. **Glitch Textiles**. https://www.glitchtextiles.com/home

368. Anne Sullivan, Joshua Allen McCoy, Sarah Hendricks, and Brittany Williams. 2018. **Loominary: Crafting Tangible Artifacts from Player Narrative**. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, 443–450. https://doi.org/10.1145/3173225.3173249

369. Anne Sullivan, Anastasia Salter, and Gillian Smith. 2018. **Games Crafters Play**. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, 1–9. https://doi.org/10.1145/3235765.3235802

370. Ruojia Sun, Ryosuke Onose, Margaret Dunne, Andrea Ling, Amanda Denham, and Hsin-Liu (Cindy) Kao. 2020. **Weaving a Second Skin: Exploring Opportunities for Crafting On-Skin Interfaces Through Weaving**. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, 365–377. https://doi.org/10.1145/3357236.3395548

371. James Utama Surjadi, Libo Gao, Huifeng Du, Xiang Li, Xiang Xiong, Nicholas Xuanlai Fang, and Yang Lu. 2019. **Mechanical Metamaterials and Their Engineering Applications**. *Advanced Engineering Materials* 21, 3: 1800864. https://doi.org/10.1002/adem.201800864

372. Haruki Takahashi and Jeeeun Kim. 2019. **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*, 43–51. https://doi.org/10.1145/3332165.3347896

373. Ye Tao, Yi-Chin Lee, Haolin Liu, Xiaoxiao Zhang, Jianxun Cui, Catherine Mondoa, Mahnoush Babaei, Jasio Santillan, Guanyun Wang, Danli Luo, Di Liu, Humphrey Yang, Youngwook Do, Lingyun Sun, Wen Wang, Teng Zhang, and Lining Yao. 2021. **Morphing Pasta and Beyond**. *Science Advances* 7, 19. https://doi.org/10.1126/sciadv.abf4098

374. TensorFlow Authors. 2022. **Image Classification**. https://github.com/tensorflow/docs/blob/bea089a0113b0459fec7f67c652e1658641cda41/site/en/tutorials/images/classification.ipynb (retrieved September 6, 2022)

375. G. Peter Tessa. 2024. **Global Textile Market Set for Major Growth by 2030**. *Textile Today*. https://www.textiletoday.com.bd/global-textile-market-set-for-major-growth-by-2030 (retrieved August 1, 2024)

376. Lavender Tessmer, Ganit Goldstein, Guillermo Herrera-Arcos, Volodymyr Korolovych, Rachel Bellisle, Cody Paige, and Christopher Shallal. 2022. **3D Knit Spacesuit Sleeve With Mulitfunctional Fibers and Tunable Compression**. In *Hybrids & Haecceities*.

377. Rundong Tian. 2019. **Lucid Fabrication**. In *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 170–173. https://doi.org/10.1145/3332167.3356881

378. Rundong Tian and Eric Paulos. 2021. **Adroid: Augmenting Hands-on Making with a Collaborative Robot**. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 270–281. https://doi.org/10.1145/3472749.3474749

379. Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. 2019. **Turn-by-Wire: Computationally Mediated Physical Fabrication**. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 713–725. https://doi.org/10.1145/3332165.3347918

380. Skylar Tibbits. 2014. **4D Printing: Multi-Material Shape Change**. *Architectural Design* 84, 1: 116–121. https://doi.org/10.1002/ad.1710

381. Cesar Armando Torres. 2019. **Hybrid Aesthetics: Bridging Material Practices and Digital Fabrication Through Computational Crafting Proxies**. UC Berkeley.

382. Cesar Torres, Wilmot Li, and Eric Paulos. 2016. **ProxyPrint: Supporting Crafting Practice Through Physical Computational Proxies**. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, 158–169. https://doi.org/10.1145/2901790.2901828

383. Cristen Torrey, Elizabeth F. Churchill, and David W. McDonald. 2009. **Learning How: The Search for Craft Knowledge on the Internet**. In *Proceedings of the 27th international conference on Human factors in computing systems*, 1371. https://doi.org/10.1145/1518701.1518908

384. Jasper Tran O'Leary, Eunice Jun, and Nadya Peek. 2022. **Improving Programming for Exploratory Digital Fabrication with Inline Machine Control and Styled Toolpath Visualizations**. In *Proceedings of the 7th Annual ACM Symposium on Computational Fabrication*, 1–12. https://doi.org/10.1145/3559400.3561998

385. Jasper Tran O'Leary, Chandrakana Nandi, Khang Lee, and Nadya Peek. 2021. **Taxon: A Language for Formal Reasoning with Digital Fabrication Machines**. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 691–709. https://doi.org/10.1145/3472749.3474779

386. Jasper Tran O'Leary and Nadya Peek. 2019. **Machine-o-Matic: A Programming Environment for Prototyping Digital Fabrication Workflows**. In *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 134–136. https://doi.org/10.1145/3332167.3356897

387. Jasper Tran O'Leary, Thrisha Ramesh, Octi Zhang, and Nadya Peek. 2024. **Tandem: Reproducible Digital Fabrication Workflows as Multimodal Programs**. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–16. https://doi.org/10.1145/3613904.3642751

388. Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2021. **Tools, Tricks, and Hacks: Exploring Novel Digital Fabrication Workflows on #PlotterTwitter**. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3411764.3445653

389. Hannah Twigg-Smith and Nadya Peek. 2023. **Dynamic Toolchains: Software Infrastructure for Digital Fabrication Workflows**. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–20. https://doi.org/10.1145/3586183.3606802

390. Hannah Twigg-Smith, Emily Whiting, and Nadya Peek. 2024. **KnitScape: Computational Design and Yarn-Level Simulation of Slip and Tuck Colorwork Knitting Patterns**. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–20. https://doi.org/10.1145/3613904.3642799

391. Jenny Underwood. 2009. **The Design of 3D Shape Knitted Preforms**. PhD Thesis: RMIT University.

392. Van De Wiele. 2020. **VSi32**. https://www.vandewiele.be/en/activities/vsi32 (retrieved December 29, 2020)

393. Ty Van de Zande and Dan Lockton. 2017. **Printerface: Screen Printed Electroluminescent Touch Interface**. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, 450–453. https://doi.org/10.1145/3132272.3132286

394. Madelyn Van der Hoogt. 1993. *The Complete Book of Drafting for Handweavers*. Shuttle-Craft Books.

395. Joshua Vasquez, Hannah Twigg-Smith, Jasper Tran O'Leary, and Nadya Peek. 2020. **Jubilee: An Extensible Machine for Multi-Tool Fabrication**. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3313831.3376425

396. Bret Victor. 2011. **Up and down the Ladder of Abstraction: A Systematic Approach to Interactive Visualization**. *Kill Math*. http://worrydream.com/LadderOfAbstraction/

397. Nancy Walsh. 2008. *Knit Foundry*. https://www.knitfoundry.com/knitvisualizer.html (retrieved December 30, 2022)

398. Guanyun Wang, Tingyu Cheng, Youngwook Do, Humphrey Yang, Ye Tao, Jianzhe Gu, Byoungkwon An, and Lining Yao. 2018. **Printed Paper Actuator: A Low-Cost Reversible Actuation and Sensing Method for Shape Changing Interfaces**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3173574.3174143

399. Jinfeng Wang, Hairu Long, Saeid Soltanian, Peyman Servati, and Frank Ko. 2014. **Electromechanical Properties of Knitted Wearable Sensors: Part 1 – Theory**. *Textile Research Journal* 84, 1: 3–15. https://doi.org/10.1177/0040517513487789

400. Lijing Wang. 2008. **The Use of Spacer Fabrics for Absorbent Medical Applications**. *Journal of Fiber Bioengineering and Informatics* 1, 4: 321–329. https://doi.org/https://doi.org/10.3993/jfbi03200910

401. Noah Wardrip-Fruin. 2007. **Playable Media and Textual Instruments**. In *The Aesthetics of Net Literature*, Peter Gendolla and Jörgen Schäfer (eds.). transcript Verlag, University of Siegen, 211–254. https://doi.org/10.1515/9783839404935-010 (retrieved August 29, 2021)

402. Noah Wardrip-Fruin and Nick Montfort. 2003. *The New Media Reader*. MIT Press, Cambridge, Massachusetts, USA.

403. Michael Wehner, Ryan L. Truby, Daniel J. Fitzgerald, Bobak Mosadegh, George M. Whitesides, Jennifer A. Lewis, and Robert J. Wood. 2016. **An Integrated Design and Fabrication Strategy for Entirely Soft, Autonomous Robots**. *Nature* 536, 7617: 451–455. https://doi.org/10.1038/nature19100

404. Jiaying Wei, Daniel Cardoso Llach, and Joshua Bard. 2023. **Responsive Robotic Assembly with Heterogeneous Materials: A Case Study with Unprocessed Wood**. In *ACADIA 2023 Habits of the Anthropocene: Scarcity and Abundance in a Post-Material Economy*. https://papers.cumincad.org/data/works/att/acadia23_v2_352.pdf (retrieved September 18, 2024)
405. Christian Weichel, John Hardy, Jason Alexander, and Hans Gellersen. 2015. **ReForm: Integrating Physical and Digital Design Through Bidirectional Fabrication**. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 93–102. https://doi.org/10.1145/2807442.2807451
406. Tali Weinberg. 2015–2019. *Woven Climate Datascapes*. http://www.taliweinberg.com/datascapes
407. Pierre Wellner. 1993. **Interacting with Paper on the DigitalDesk**. *Communications of the ACM* 36, 7: 87–96. https://doi.org/10.1145/159544.159630
408. Kristin Williams, Jessica Hammer, and Scott E. Hudson. 2021. **An Upcycled IoT: Building Tomorrow's IoT Out of Today's Household Possessions**. *XRDS: Crossroads, The ACM Magazine for Students* 27, 4: 19–25. https://doi.org/10.1145/3466872
409. Karl D.D. Willis, Cheng Xu, Kuan-Ju Wu, Golan Levin, and Mark D. Gross. 2011. **Interactive Fabrication: New Interfaces for Digital Fabrication**. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*, 69. https://doi.org/10.1145/1935701.1935716
410. Hannah Perner Wilson. 2014. **Igne Oyasi Motor**. *How to Get What You Want*. https://www.kobakant.at/DIY/?p=5277 (retrieved August 2, 2024)
411. Margaret B. Windeknecht and Thomas G. Windeknecht. 1980. **Microcomputer Graphics and the Color-and-Weave Effect in Handweaving**. In *Proceedings of the 18th annual Southeast regional conference*, 174–179. https://doi.org/10.1145/503838.503866
412. Wireshark Foundation. 1998–2020. *Wireshark*. https://www.wireshark.org/
413. Matthijs Witsenburg. 2016. **Laser Cut Plywood Bandweaving Loom**. *Hackster.io*. https://www.hackster.io/matthijs-witsenburg/laser-cut-plywood-bandweaving-loom-32e22c (retrieved December 28, 2020)
414. Anthony S.W. Wong and Y.I. Li. 2006. **Prediction of Clothing Sensory Comfort**. In *Clothing Biosensory Engineering*, Y. Li and A.S.W. Wong (eds.). Woodhead Publishing, Online, 178–188. https://doi.org/https://doi.org/10.1533/9781845691462.178
415. Kui Wu, Hannah Swan, and Cem Yuksel. 2019. **Knittable Stitch Meshes**. *ACM Transactions on Graphics* 38, 1: 10:1–10:13. https://doi.org/10.1145/3292481
416. Rundong Wu, Joy Xiaoji Zhang, Jonathan Leaf, Xinru Hua, Ante Qu, Claire Harvey, Emily Holtzman, Joy Ko, Brooks Hagan, Doug James, François Guimbretière, and Steve Marschner. 2020. **Weavecraft: An Interactive Design and Simulation Tool for 3D Weaving**. *ACM Transactions on Graphics* 39, 6: 210:1–210:16. https://doi.org/10.1145/3414685.3417865
417. Shanel Wu and Laura Devendorf. 2020. **Unfabricate: Designing Smart Textiles for Disassembly**. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3313831.3376227

418. Xiang 'Anthony' Chen, Stelian Coros, Jennifer Mankoff, and Scott E. Hudson. 2015. **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*, 73–82. https://doi.org/10.1145/2807442.2807498

419. Xiang 'Anthony' Chen, Jeeeun Kim, Jennifer Mankoff, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2016. **Reprise: A Design Tool for Specifying, Generating, and Customizing 3D Printable Adaptations on Everyday Objects**. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 29–39. https://doi.org/10.1145/2984511.2984512

420. Xiang 'Anthony' Chen, Ye Tao, Guanyun Wang, Runchang Kang, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2018. **Forte: User-Driven Generative Design**. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 496:1–496:12. https://doi.org/10.1145/3173574.3174070

421. Ting Xu and Henri J. Lezec. 2014. **Visible-Frequency Asymmetric Transmission Devices Incorporating a Hyperbolic Metamaterial**. *Nature Communications* 5, 1: 4141. https://doi.org/10.1038/ncomms5141

422. MIAO Xu-hong and Ge Ming-Qiao. 2008. **The Compression Behaviour of Warp Knitted Spacer Fabric**. *Fibres & Textiles in Eastern Europe* 16, 1: 90–92.

423. Mayu M. Yamashita, Junichi Yamaoka, and Yasuaki Kakehi. 2013. **Enchanted Scissors: A Scissor Interface for Support in Cutting and Interactive Fabrication**. In *ACM SIGGRAPH 2013 Posters on - SIGGRAPH '13*, 1. https://doi.org/10.1145/2503385.2503422

424. Zeyu Yan, Hsuanling Lee, Liang He, and Huaishu Peng. 2023. **3D Printing Magnetophoretic Displays**. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–12. https://doi.org/10.1145/3586183.3606804

425. Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. **Investigating How Experienced UX Designers Effectively Work with Machine Learning**. In *Proceedings of the 2018 Designing Interactive Systems Conference*, 585–596. https://doi.org/10.1145/3196709.3196730

426. Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. **PneUI: Pneumatically actuated soft composite materials for shape changing interfaces**. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, 13–22. https://doi.org/10.1145/2501988.2502037

427. Katherine Ye, Wode Ni, Max Krieger, Dor Ma'ayan, Jenna Wise, Jonathan Aldrich, Joshua Sunshine, and Keenan Crane. 2020. **Penrose: From Mathematical Notation to Beautiful Diagrams**. *ACM Transactions on Graphics* 39, 4. https://doi.org/10.1145/3386569.3392375

428. Nur Yildirim, Alex Kass, Teresa Tung, Connor Upton, Donnacha Costello, Robert Giusti, Sinem Lacin, Sara Lovic, James M O'Neill, Rudi O'Reilly Meehan, Eoin Ó Loideáin, Azzurra Pini, Medb Corcoran, Jeremiah Hayes, Diarmuid J Cahalane, Gaurav Shivhare, Luigi Castoro, Giovanni Caruso, Changhoon Oh, James McCann, Jodi Forlizzi, and John Zimmerman. 2022. **How Experienced Designers of Enterprise Applications Engage AI as a Design Material**. In *CHI Conference on Human Factors in Computing Systems*, 1–13. https://doi.org/10.1145/3491102.3517491

429. Xiaoxiang Yu, Surya Girinatha Nurzaman, Utku Culha, and Fumiya Iida. 2014. **Soft Robotics Toolkit Tendon Module Box**. https://softroboticstoolkit.com/hot-glue-kit/design/tendon-module-box/electronics

430. Cem Yuksel, Jonathan M. Kaldor, Doug L. James, and Steve Marschner. 2012. **Stitch Meshes for Modeling Knitted Clothing with Yarn-Level Detail**. *ACM Transactions on Graphics* 31, 4: 1–12. https://doi.org/10.1145/2185520.2185533

431. Qiji Ze, Xiao Kuang, Shuai Wu, Janet Wong, S. Macrae Montgomery, Rundong Zhang, Joshua M. Kovitz, Fengyuan Yang, H. Jerry Qi, and Ruike Zhao. 2020. **Magnetic Shape Memory Polymers with Integrated Multifunctional Shape Manipulation**. *Advanced Materials* 32, 4: 1906657. https://doi.org/10.1002/adma.201906657

432. Mengjia Zhu, Amirhossein H. Memar, Aakar Gupta, Majed Samad, Priyanshu Agarwal, Yon Visell, Sean J. Keller, and Nicholas Colonnese. 2020. **PneuSleeve: In-Fabric Multimodal Actuation and Sensing in a Soft, Compact, and Expressive Haptic Sleeve**. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3313831.3376333

433. David Zicarelli. 2018. *Max*. Cycling '74. https://cycling74.com/products/max (retrieved January 1, 2023)

434. John Zimmerman, Jodi Forlizzi, and Shelley Evenson. 2007. **Research Through Design As a Method for Interaction Design Research in HCI**. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 493–502. https://doi.org/10.1145/1240624.1240704

435. Amit Zoran. 2016. **A Manifest for Digital Imperfection**. *XRDS: Crossroads, The ACM Magazine for Students* 22, 3: 22–27. https://doi.org/10.1145/2893491

436. Amit Zoran and Joseph A. Paradiso. 2013. **FreeD: A Freehand Digital Sculpting Tool**. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2613. https://doi.org/10.1145/2470654.2481361

437. Amit Zoran, Roy Shilkrot, Suranga Nanyakkara, and Joseph Paradiso. 2014. **The Hybrid Artisans: A Case Study in Smart Tools**. *ACM Transactions on Computer-Human Interaction* 21, 3: 1–29. https://doi.org/10.1145/2617570

Acknowledgements

I have been incredibly well-supported in my PhD studies. Formulating sentences that can truly convey the experience of the thing is frankly beyond me, and I am inevitably accidentally omitting somebody. Nonetheless. **This thesis would not have been possible without...**

My advisors:
- Scott Hudson, who treats his students as peers and his hobbies as research, who is equally thoughtful and open-minded about both, and who is shaping up to be a pretty decent basket-weaver.
- Lining Yao, who is a master of the enthusiastic "yes, and" across disciplines, even in the face of tight deadlines.

My thesis committee:
- Mark Gross, who invited me into the community of his lab and handed me unlimited access to the lab's private laser cutter when I was a scrappy little undergraduate architecture student, and who has continued to be a positive influence—shaping our field for the better and weirder as the head of ATLAS, and supporting me as a committee member—ever since then.
- Jess Hammer, who is comprehensively an inspiration to me as a researcher, as a teacher, and as a community member.

My bonus research mentors(!):
- Laura Devendorf, whose own research was a catalyst for my understanding of what HCI research could be, and who was an insightful collaborator and generous lab host when I needed help finding my research-heart.
- Jim McCann, who can be counted on for a productive argument and to never take "I don't know how to do that" as an answer, and who very generously materially supported me with unquestioned access to Textiles Lab resources.
- Olivia Robinson, whose reflective development of the term "Soft Technologies" has clearly been influential on how I think about my research and my teaching.

My communities:
- Everyone who has supported my professional development:
  again Jim "we'll make a researcher of you yet, Lea" McCann, Jessica Hodgins, the STUDIO for Creative Inquiry (Golan, Marge, Linda, Harrison, Tom, Bill), IDeATe (Susan, Olivia, Zach, Kelly, Ryan), and the HCII's very own Queenie Kravitz and Leah Buffington.
- Everyone I could talk to about Computer Problems:
  first and most influentially my father, of course, and also: Glenn Willen, Trevor Burns, Jason C. Reed, Ross Hatton, William Lovas and the entire Recurse Center community (especially Lindsey Kuper, Alex Rudnick, Julia Evans, Emilia Lazer-Walker, Lyndsey Moulds).
- Everyone I could talk to about Research:
  HCII friends in my cohort and in my labs (DevLab, Morphing Matter Lab, Textiles Lab and the 407 S Craig community), especially: L. Kirabo Byabashaija and Stephanie Valencia^2, Sam Reig, Julia Cambre, Erica Cruz,

Nur Yildirim, Jenny Lin, Vidya Narayanan, Gabrielle Ohlson, Tucker Rae-Grant, Humphrey Yang, Jesse Gonzalez, Michael Vinciguerra
...and inspiring research buddies in far-flung locales: Yi-Chin Lee, Danli Luo, Ticha Sethapakdi, Max Kreminski, Sam Goree, Jack Forman, Taylor Tabb, Cedric Honnet, Gillian Smith, Madeline Gannon, Claire Henschtker, and everyone in the UW Machine Agency and the CU Unstable Design Lab.

- Friends I could talk to about things that aren't Computer Problems or Research:
  Trevor Burns, Kathleen Brousseau, Akiva Leffert, Vincent Zeng, Brian Saghy, Tom7, Rose Turner, Sophia Wu, Kat Hagen, Sumana Harihareswara
- Friends drawn with a very fine camel hair brush:
  Chris Martens and Rob Simmons
- Friends who can be relied upon to handle dry ice and to pour concrete at 2AM:
  none other than the inimitable Rebecca Grabman
- Small friends who supplied warmth and routine in a cold and aimless time:
  Johnny, Junebug, and Lula Rat

And, always:

- Linda Benenati and Michael Albaugh: who genuinely do *not* push me to work on overambitious and stressful projects, but who do seem to believe I can accomplish them anyway (while assiduously ignoring any evidence to the contrary)
- David Renshaw: who provides a nutritious diet, a cozy nest, and enrichment in my habitat.

***

This document was set in Eames Century Modern by House Industries, with intertitles in Lust Script by Neil Summerour (Positype) and section headers in Degular by James Edmondson (Ohno).

It was generated with Pandoc and Weasyprint, with help from Zotero and the CiteBibtex plugin for Sublime Text.