Metamaterial Devices

by Alexandra Ion, MSc

a dissertation submitted in partial fulfillment
of the requirements for the degree of

Dr. rer. nat.

Computer Science, Human-Computer Interaction
defended on 26. April 2019

Hasso Plattner Institute
Digital Engineering department, University of Potsdam

Thesis committee:
Advisor: Prof. Dr. Patrick Baudisch (Hasso Plattner Institute)

Reviewers:
Prof. Dr. Scott E. Hudson (Carnegie Mellon University)
Prof. Dr. Sriram Subramanian (University of Sussex)

Members of the committee:
Prof. Dr. Marc Alexa (TU Berlin)
Prof. Dr. Felix Naumann (Hasso Plattner Institute, secondary advisor)
Prof. Dr. Andreas Polze (Hasso Plattner Institute, head of the committee)

Published online at the
Institutional Repository of the University of Potsdam:
https://doi.org/10.25932/publishup-42986
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-429861

To David,

you know why .)

Abstract

Digital fabrication machines such as 3D printers excel at producing arbitrary shapes, such as for decorative objects. In recent years, researchers started to engineer not only the outer shape of objects, but also their internal microstructure. Such objects, typically based on 3D cell grids, are known as metamaterials. Metamaterials have been used to create materials that, e.g., change their volume, or have variable compliance.

While metamaterials were initially understood as materials, we propose to think of them as *devices*.

We argue that thinking of metamaterials as devices enables us to create internal structures that offer functionalities to implement an *input-process-output model* without electronics, but purely within the material's internal structure. In this thesis, we investigate three aspects of such metamaterial devices that implement parts of the input-process-output model: (1) materials that process analog inputs by implementing mechanisms based on their microstructure, (2) that process digital signals by embedding mechanical computation into the object's microstructure, and (3) interactive metamaterial objects that output to the user by changing their outside to interact with their environment. The input to our metamaterial devices is provided directly by the users interacting with the device by means of physically pushing the metamaterial, e.g., turning a handle, pushing a button, etc.

The design of such intricate microstructures, which enable the functionality of metamaterial devices, is not obvious. The complexity of the design arises from the fact that not only a suitable cell geometry is necessary, but that additionally cells need to play together in a well-defined way. To support users in creating such microstructures, we research and implement interactive design tools. These tools allow experts to freely edit their materials, while supporting novice users by auto-generating cells assemblies from high-level input. Our tools implement easy-to-use interactions like brushing, interactively simulate the cell structures' deformation directly in the editor, and export the geometry as a 3D-printable file. Our goal is to foster more research and innovation on metamaterial devices by allowing the broader public to contribute.

Zusammenfassung

Digitale Fabrikationsmaschinen, wie 3D-Drucker, eignen sich hervorragend um beliebige Formen zu produzieren. Daher sind sie bei Endnutzern für die Erstellung von dekorativen Elementen sehr beliebt. Forscher hingegen haben in den letzten Jahren damit begonnen, nicht nur die äußere Form zu betrachten, sondern auch Mikrostrukturen im Inneren. Solche Strukturen, die meist auf einem 3-dimensionalen Gitter angeordnet sind, sind als "Metamaterialien" bekannt. Metamaterialien wurden entwickelt, um Eigenschaften wie Volumenänderung oder lokalisiert die Steifheit des Materials zu steuern.

Traditionell werden Metamaterialien als Materialien betrachtet, wir hingegen betrachten sie als *Geräte*.

In dieser Arbeit zeigen wir, dass die Betrachtung von Metamaterialien als Geräte es erlaubt Strukturen zu kreieren, die Geräte nach dem *Eingabe-Verarbeitung-Ausgabe* Prinzip realisieren – und das gänzlich ohne Elektronik. Wir untersuchen 3 Aspekte von solchen funktionsfähigen Metamaterial-Geräten die jeweils Teile des EVA Prinzips implementieren: (1) Materialien, die analoge Eingabe als Mechanismen, die durch ihre Mikrostruktur bestimmt sind, verarbeiten, (2) Materialien, die digitale Eingabe verarbeiten und mechanische Berechnungen in ihrer Mikrostruktur durchführen und (3) Materialien, die ihre äußere Textur dynamisch verändern können um mit dem Nutzer zu kommunizieren. Die Eingabe für Meta-

material-Geräte ist in dieser Arbeit direkt durch den Nutzer gegeben, der mit dem Gerät interagiert, zum Beispiel durch Drücken eines Griffs, eines Knopfes, etc.

Das Design von solchen filigranen Mikrostrukturen, die die Funktionalität der Metamaterial-Geräte definieren, ist nicht offensichtlich oder einfach. Der Designprozess ist komplex, weil nicht nur eine Zellstruktur gefunden werden muss, die die gewünschte Deformation durchführt, sondern die Zellstrukturen zusätzlich auf wohldefinierte Weise zusammenspielen müssen. Um Nutzern die Erstellung von diesen Mikrostrukturen zu ermöglichen, unterstützen wir sie durch interaktive Computerprogramme, die wir in dieser Arbeit untersuchen und implementieren. Wir haben Software entwickelt, die es Experten erlaubt die Mikrostrukturen frei zu platzieren und zu editieren, während Laien durch automatisch generierte Strukturen geholfen wird. Unsere Software beinhaltet einfach zu bedienende Interaktionskonzepte, wie zum Beispiel das aufmalen von funktionalen Eigenschaften auf Objekte, eine integrierte Vorschau der Deformation, oder der 3D-druckfähige Export der erstellten Geometrie. Das Ziel dieser Arbeit ist es langfristig Forschung und Innovation von Metamaterial-Geräten zu fördern, so dass sich sogar die breite Masse in das Thema einbringen kann.

Acknowledgments

First, I would like to thank my advisor Patrick Baudisch. He shaped my thinking about research early on and encouraged be to big, wild and far ahead. While we might not always achieve this, it is certainly great to always keep this as the goal. His contribution of ideas for valuable research directions is one of the reasons, this thesis exists and for that I am incredibly thankful. (The other reason is hard work.)

I would also like to thank Michael Haller, who advised my master thesis and with that introduced me to the field of HCI. Without him I would certainly not be here finishing my PhD. I would also like to thank Mark Hancock and Stacey Scott, who hosted me for 5 months and who I had wonderful discussion with that further shaped my future path in research.

Without my colleagues at the HPI lab, I could not have done my work. I want to thank Pedro Lopes, who always took the time to give me deep and insightful feedback, whenever I asked. He also always had an open ear when things got tough. I appreciate this a ton, thank you! Robert Kovacs can also not go unmentioned, as we shared an office, laughter and insightful discussions about each others projects. I want to thank Stefanie Müller for encouraging me, I always felt like she believes in me. I want to thank the entire lab: Mirela Alistar, Thijs Roumen, Lung-Pan Cheng,

Oliver Schneider Sebastian Marwecki, Jotaro Shigeyama; and former labmates Dominik Schmidt and Tim Chen for discussions and fun times! I was also lucky to work with many wonderful students, in particular I want to mention Ludwig Wall and Johannes Frohnhofen.

Many people at HPI helped create a productive evironment and cannot go unmentioned here. I want to thank the lab's administrative assistant Frau Tholen for her support. Furthermore, I like to thank Daniela Roick-Schütze, the HPI research school that I was a part of for 4 years (Prof. Andreas Polze and Sabine Wagner), our IT-support (especially Andy Weber) and Christian from the security team.

I also want to shout out a special thanks to my friend Philipp Herholz, who was never tired of answering my technical questions. I also would like to mention Marc Alexa and Felix Naumann, who I occasionally bugged with questions about career and who patiently bared with me.

I want to thank my parents and my sister for their support. Although my parents came as immigrants to Austria and had tough jobs, they always encouraged me to study whatever I wanted to study and to strive for the best. Thank you for not setting borders!

Lastly, I have to thank my wonderful husband, love of my life, best friend, and even colleague David. Everyone in their doctoral studies knows that it can be hard sometimes. Without you, I could not have done this. You are my rock, my save haven, you are the one who believes in me when no one else (not even me) does. Thank you for your love who made me a happier, optimistic and more generous person. You definitely made me be the best version of myself. Everyday I am looking forward to our future together.

Publications

Some ideas and figures have appeared previously in the following publications. Specific chapters and sections that directly derived from each publication are listed:

Chapter 3 was published and presented as: Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. Metamaterial Mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 529–539, New York, New York, USA, 2016. ACM Press. ISBN 9781450341899. doi:10.1145/2984511.2984540

Chapter 3.8 was published and presented as: Alexandra Ion, David Lindlbauer, Philipp Herholz, Marc Alexa, and Patrick Baudisch. Understanding metamaterial mechanisms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 647:1–647:14, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-5970-2. doi:10.1145/3290605.3300877

Chapter 5 was published and presented as: Alexandra Ion, Ludwig Wall, Robert Kovacs, and Patrick Baudisch. Digital Mechanical Metamaterials. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 977–988, New York, New York, USA, 2017. ACM Press. ISBN 9781450346559. doi:10.1145/3025453.3025624

Chapter 6 was published and presented as: Alexandra Ion, Robert Kovacs, Oliver S. Schneider, Pedro Lopes, and Patrick Baudisch. Metamaterial Textures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, New York, New York, USA, 2018. ACM Press. ISBN 9781450356206. doi:10.1145/3173574.3173910

Declaration

I hereby confirm that

- this dissertation is the result of my own work, it was prepared without unauthorized help and using only the given literature,
- this dissertation has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification
- I am aware of the doctorate regulations of the department for Digital Engineering of the University of Potsdam from April 24, 2017

Ich erkläre hiermit, dass

- ich die vorliegende Dissertationsschrift selbständig und ohne unerlaubte Hilfe angefertigt sowie nur die angegebene Literatur verwendet habe,
- die Dissertation keiner anderen Hochschule in gleicher oder ähnlicher Form vorgelegt wurde,
- mir die Promotionsordnung der Digital Engineering Fakultät der Universität Potsdam vom 24. April 2017 bekannt ist.

Alexandra Ion – November 2, 2018

Contents

1

Introduction

Personal fabrication machines, such as 3D printers, have increasing impact on many areas including academia, aerospace industry, clothing or simply end-user products. The reason why 3D printing is so influential is that, in contrast to other manufacturing technologies such as CNC milling, molding, forming or laser cutting, 3D printing is the only technology that allows us to *arrange matter freely in space.* A 3D printer is a generic machine that creates a physical representation of potentially any digital 3D model that users design. Such machines usually start with an empty build space and build up the object layer by layer. Due to this layer-by-layer approach the tool head that extrudes material always has access to the object and can deposit material anywhere on top of the last layer, which allows for arbitrary complexity.

Naturally, the first 3D prints revolved around arbitrary shapes. However, researchers quickly found that by leveraging the freedom in material arrangement and altering how the material on the inside of objects is distributed, unique physical properties in their printed artifacts are enabled.

From the outside shape...

Researchers in human-computer interaction started early on to investigate fabrication systems that are accessible for novice users. They reduced the hurdle of designing 3D models by presenting sketch-based CAD tools. Furthermore, researchers also enabled interactive fabrication systems by allowing users to directly interact with the machine [96, 116, 182], increasing the number of iteration cycles [98, 99], use real-world objects for content creation [180, 181], embed mechanisms [192] and electronics [71, 179], or even fabricate things on the go [135]. These were mainly concerned with creating the outside shape of 3D models.

to internal material distribution...

Since 3D printing allows for creating physical objects with complex geometry, researchers investigated the redistribution of material on the inside of objects. They proposed tools to save material by optimizing the strength to weight ratio of 3D objects [72], allow arbitrarily shaped objects to spin reliably [9], or to float in pre-defined poses [123]. The availability of commercial tools [5, 6] to calculate the optimal material distribution for static objects even let industry adopt 3D printing as a production technique for special parts [147].

...to engineered microstructures

Researchers in physics and mechanical engineering work on a similar concept, but they push the boundaries of engineered structures even further. They started to divide materials into a large number of cells on a regular grid. Since each cell is designed to perform a specific deformation [108], objects that entirely consist of such cells literally offer thousands of degrees

of freedom. Such structures are also known as mechanical *metamaterials*. Metamaterials are artificial structures with mechanical properties that are defined by their usually repetitive cell patterns, rather than the material they are made of [16, 27, 113].

Such metamaterials can exhibit extreme properties. Researchers developed nano-structures that yield ultra-lightweight materials [84, 85], they investigated structures that can change their volume, i.e., they expand in two dimensions upon one-dimensional extension (so-called 'auxetics' [15, 61, 69]), recoverable materials that damp impacts [35, 149], and even materials that hide enclosed objects from being felt [19], that are tunable wave guides [8, 81] or as a thermal conductance managing materials [120]. As these examples demonstrate, metamaterials allow researchers to create materials with very desirable and controllable properties.

> **So far, metamaterials have been understood as materials. The main contribution of this thesis is that we think of them as *devices*.**

1.1 METAMATERIAL DEVICES

While metamaterials were already demonstrated to create materials with extreme properties, we push the field further by creating metamaterials that function like devices. The functionality of our metamaterial devices is solely defined by their cell geometry. We build devices from cells that *play together* in a well-defined way to create interactive devices without electronics, but are purely defined by their material structure. The key benefit of our approach is that the resulting devices can be 3D printed from

a single material and in one piece and thus do not require assembly. This means that they can be 3D printed as a part of a larger structure, such as full door including handle, latch mechanism and lock.

To instantiate this concept of metamaterial devices, we took inspiration from the input-process-output model. In this thesis, we present novel cell structures that implement mechanical devices that can process and output information depending on mechanical input from the user.

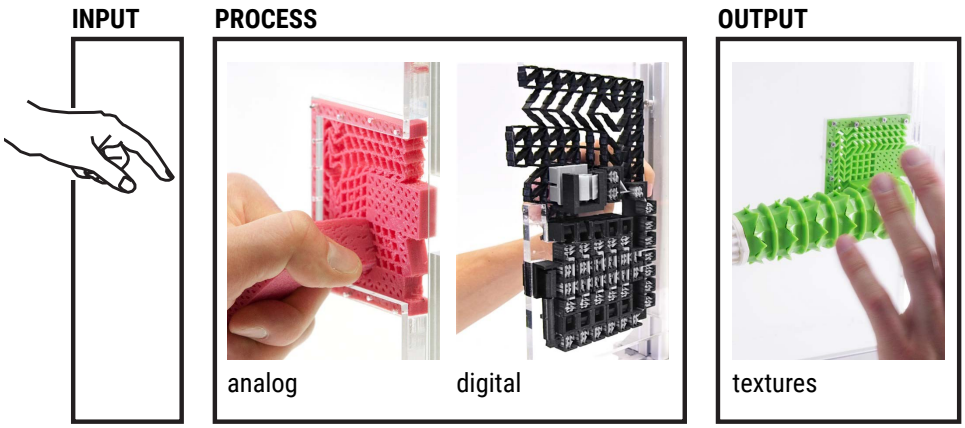

Figure 1.1: Metamaterial devices can process users' input and output to them again, thereby implementing an *input-process-output* system without electronics, but purely within the material structure.

Figure 1.1 illustrates the core areas that we investigated in this thesis: we show (1) how we process *analog* information in the context of metamaterials that transform forces and movements input by the user into a different set of forces and movement (i.e., mechanisms), (2) a system of cells that process information *digitally*, and (3) cells that output information to the user or interface with the environment by appropriating dynamic textures.

Since designing such complex systems of cell structures is difficult, we additionally focus our work on providing accessible software to enable users to create such metamaterial devices.

1.1.1 Processing analog signals

We push the concept of metamaterials further by creating metamaterials that implement devices that transform input forces and movement into a desired set of output forces and movement—also known as mechanisms. Such metamaterial mechanisms consist of a single block of material the cells of which play together in a well-defined way in order to achieve macroscopic movement.

Figure 1.2 shows an example of a metamaterial mechanism: a door latch mechanism. Our metamaterial door latch transforms the rotary movement of its handle into a linear motion of the latch. The key to transforming analog input are our shearing cells as they are only able to shear when subjected to an external force and thereby apply a controlled force to their neighboring cells. While the traditional door latch mechanism consists of several parts, including an axle, bearings, springs, etc., the metamaterial door latch in Figure 1.2 consists of a single block of material, as it is groups of cells inside the object that perform the mechanical function.

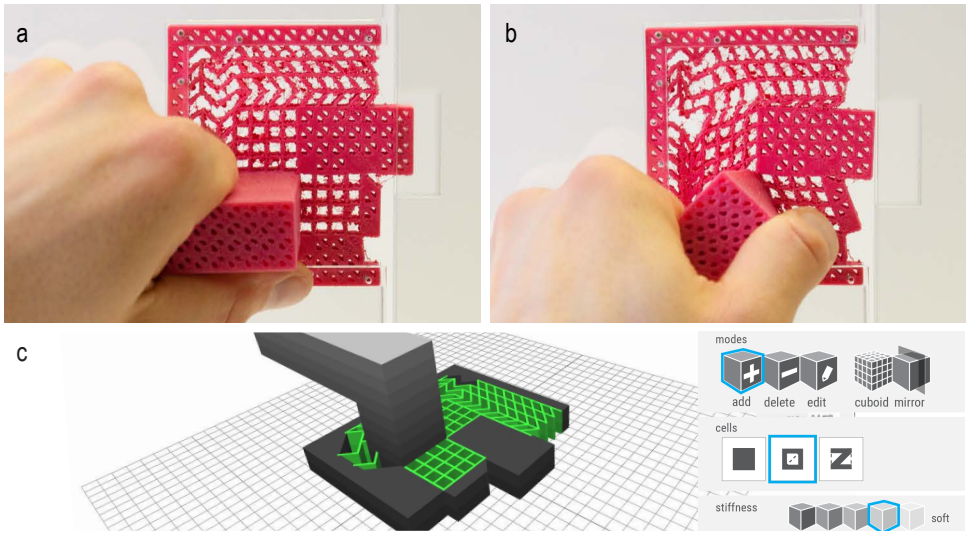

Figure 1.2: (a) This door latch demonstrates a metamaterial device that transforms user input, i.e., pushing down the handle, (b) to pulling the latch inwards. (c) We developed a custom editor to assist their design.

The design of metamaterial mechanisms is highly challenging and the extent of mechanisms that can be realized with metamaterials remains unclear. We show in Figure 1.3 how a small change can lead to a vastly different output. The reason is that a mechanism consists of many cells that are interconnected and impose constraints on each other, the behaviour of the cell structure is unobvious and non-linear. To enable novice users to design such metamaterial mechanisms, we implemented a computational design tool takes user-drawn paths as input and optimizes the cell configuration which implements the transformation.

To achieve this, we dig deeper into the underlying topological constraints of such cell structures and their influence on the resulting mechanism. We find that by representing the constraints as a graph we can easily distinguish unique cells configuration.

For example, Figure 1.3 illustrates that one cell can prevent 7 cells from shearing such that any permutation within these 7 cells creates the same output. We bypass these $2^7 = 128$ configurations by only operating on the constraint graph (i.e., merging and splitting connected components), which reduces our search space by several orders of magnitude and ultimately enables such an automatic tool.

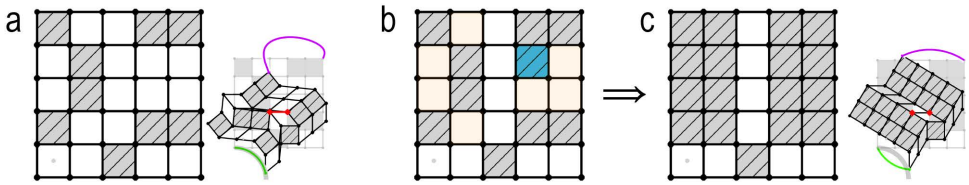

Figure 1.3: (a) In this example, (b) setting one cell rigid (c) prevents 7 cells from shearing, which changes the output drastically. (c) The green input path from (a) cannot be followed anymore.

1.1.2 Processing digital signals

Such analog machines, however, are limited in terms of complexity. As forces are passed on from one cell to the next, they are damped and the activation energy dissipates, causing the mechanical "signal" to decay. This limits the number of mechanisms that can be concatenated and therefore the complexity of the machine. While this decay can be minimized, it can never be eliminated. This motivated the question 'Can we create metamaterials that process digital input, which have no signal decay?'

To explore this concept, we introduce a new type of cell that propagates a digital mechanical signal, i.e., it counteracts signal decay and thus allows signals to pass through an arbitrary number of cells. Our cell embeds a bistable spring, which discharges when triggered. The resulting impulse triggers one or more neighboring cells, resulting in signal propagation. These cells allow users to implement combinational circuits within 3D printed metamaterial objects. Figure 1.4 shows a combination lock implemented using digital metamaterials. We contribute 3D printable cells that process signals and make this concept of digital metamaterials accessible to users by extending our specialized 3D voxel-style editor.

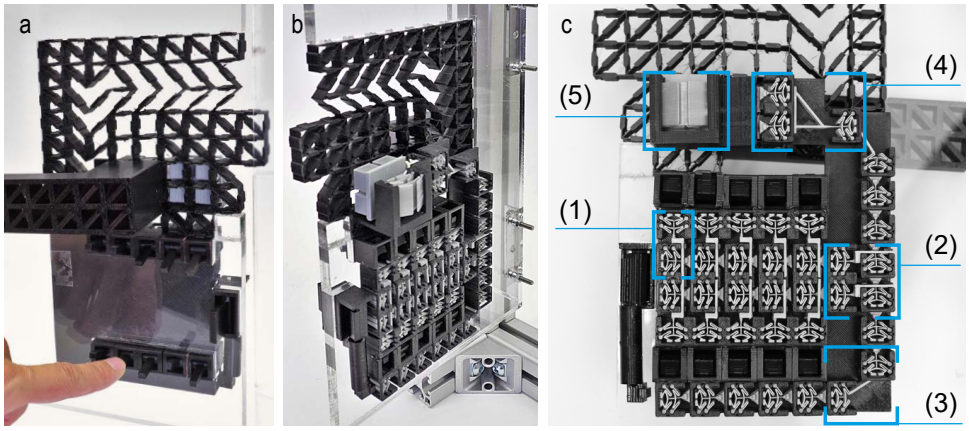

Figure 1.4: (a) This combination door lock consists of cells that process the code and (b) only unlock the latch if users input the correct code. (c) Internally, the lock consists of that implement (1) signal validation, (2) logic gates, (3) signal routing, (4) bifurcation, and (5) amplification.

1.1.3 Output to the user and the environment

After investigating two types of metamaterial structures that process information input by the user, we shifted our focus to create metamaterials that *output* to the user.

We apply the main idea behind metamaterials, i.e., subdivision into a large number of cells and customization on a per-cell basis, to the outsides of 3D printed objects. We introduce metamaterials that undergo a controlled transformation when an external force is applied, allowing users to transition between multiple dynamic textures after fabrication. The resulting metamaterial textures allow designers to shape how the object interacts with the environment and with the tactile sense of the user. Figure 1.5 shows an example. This door handle transforms its outside, allowing the person behind the door to set three levels tactile messages for everyone trying to enter. The inside of the door handle consists of a grid of cells, which controls how the texture on the object's surface will be formed.

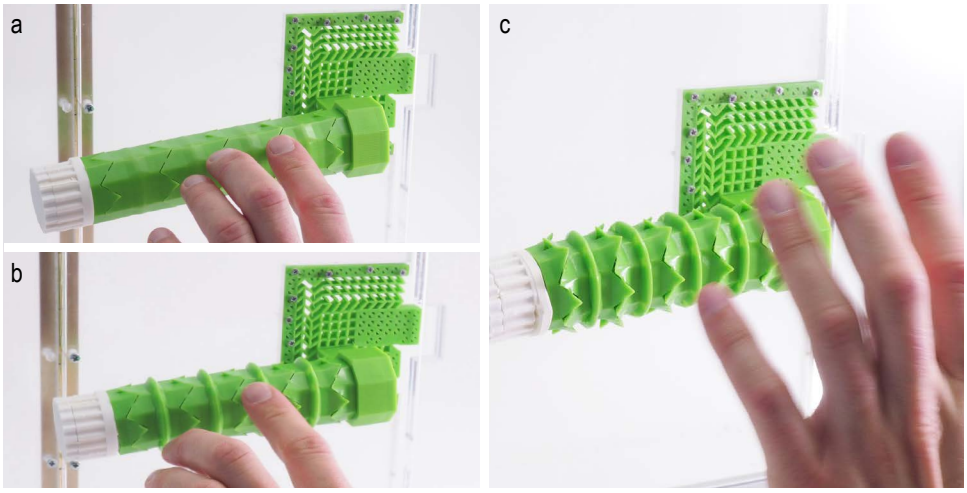

Figure 1.5: When an external force is applied, metamaterial textures undergo a controlled transformation. This door handle, transforms (a) from flat (b) to rippled (c) to spiky, allowing the person behind the door to set a tactile message with three levels of enter/busy/do not enter messages for visually impaired or sighted users trying to enter.

1.2 CONTRIBUTIONS

The concept of metamaterial devices. The main contribution of this thesis is that we apply a radically different thinking to metamaterials. While this emerging field already demonstrated that materials with extreme properties can be engineered by designing their cell structure, we push this further and argue that we can engineer material properties to function as *devices*.

Applications and design space. We investigated the potential impact and relevance of such metamaterial devices by exploring and building tangible prototypes that can be experiences. These demonstrations inform the potential design space for such devices.

Cell designs. We also contribute a set of cells that enable the realization of metamaterial devices. Our cells include cells that enable a well-defined movement for transmitting forces between cells, they include cells that enable the processing and validation of digital signal and cells that change their outside to interface with the user and the environment.

Software. Lastly, to enable users to create such metamaterial devices, we contribute easily accessible[1] software tools that encapsulate our findings. Our software contains automatic generation of cell assemblies for novice users yet allows experts to freely edit their devices.

1.3 STRUCTURE OF THIS THESIS

We first give the reader a deeper insight into the current state of the art regarding digital fabrication and metamaterials in Chapter 2. Then, in Chapter 3, we introduce the concept of metamaterial mechanisms. We dive deeper into the structural properties of metamaterial mechanisms in Chap-

[1] open source editor for Metamaterial Mechanisms: https://github.com/jfrohnhofen/metamaterial-mechanisms

ter 3.8, where we also present a computational design tool that automates the generation of metamaterial mechanisms. We then move on to detailing the structures that enable digital metamaterial devices in Chapter 5, before explaining how to output to the user by appropriating dynamic textures in Chapter 6. In each chapter, we demonstrate the software that enables users to create such metamaterial devices. Lastly, Chapter 7 concludes this thesis by discussing the work on a broader level. Since the concept presented in this thesis is novel, it opens up future directions of research, which we identify and discuss to leave the reader with more food for thought.

2

Related work

Our work builds on previous work in interactive personal fabrication, on techniques that modify the internal structure of 3D printed objects, multi-material printing, and in particular on mechanical metamaterials.

2.1 INTERACTIVE SYSTEMS FOR FABRICATION

Personal fabrication devices such as 3D printers and laser cutters allow users to fabricate personalized physical objects [159]. Researches presented interactive systems that ease the process of designing static objects, creating interactive objects and interacting directly with the machine.

2.1.1 Designing 3D models

The first step to fabricating custom object using 3D printing is to design the digital 3D model. Since Sutherland proposed the first CAD tool back in 1963 [158], many tools have emerged. Currently, the most popular commercial tools include easy-to-use CAD systems such as Tinkercad or SketchUp, expert tools such as SolidWorks and Inventor, or parametric modelling tools such as Rhinoceros or openSCAD [55].

Researchers in HCI and computer graphics investigated systems to ease the process of creating 3D models, e.g., by implementing sketching-based interfaces for simple toy-like models [54], or for more complex models based on assisted 3D sketching [11, 12, 165]. Going beyond creating only the virtual model, researchers integrated the fabrication of the physical model into the design process by simulating e.g., sown plush models [94] or the stability of an interlocked laser cut chair [136]. In general, making 3D objects from laser cut 2D shapes by interlocking them orthogonally is popular, since laser cutting is simpler and faster than 3D printing. Researchers exploited these benefits and developed systems to create 3D model from laser cut pieces [50, 77].

More complex mechanical objects and articulated figures, rather than aforementioned static objects, require special items for construction, such as joints [167] and user-defined motion-ranges [78]. LinkEdit [10] allows users to edit moving linkages to satisfy space constraints. Megaro et al. [79] took this concept further and implemented a system that generates stable motions for legged robots of arbitrary designs and exports 3D-printable geometry.

2.1.2 Interacting with the machine

The traditional workflow for fabricating physical models consists of users creating the 3D model at their computer, then sending the model to the fabrication machine and waiting until the machine is done. In case of 3D

printing, this could mean waiting for hours or days until the object is finished. However, when fit or ergonomics is an issue, this process is not feasible as iterating over a design becomes very time-consuming. For achieving a good fit between a fabricated object and an external object, Weichel et al. [180, 181] proposed using the target objects in an augmented reality environment during the modeling process. Gannon et al. [37] proposed a fabrication system that allows users to design directly on their skin for body-fit items such as casts.

The ability to iterate over a design is crucial for rapid prototyping, which is why researcher investigated how to decrease the time between iterations. This includes integrating building blocks [99], printing only low-fidelity previews [98], or destroying parts of the model for updating instead of reprinting the whole object [160]. *On-the-Fly Print* [116] even allows users to design 3D models digitally while having a low-fidelity physical wireframe model printed in parallel.

Going beyond quick iterations cycles, researchers investigated interactive systems that allows users to interact directly on the machine. This powerful concept was first instantiated by a foam extruder that was integrated with a touch-table [182]. Extending this concept, Müller et al. investigated systems that allow users to draw mechanical parts directly onto the laser cutter [96, 97]. This makes the machine a collaborator, which was taken further by systems such as *FormFab* [13] or *RoMA* [117] that enable a collaboration with a robotic arm for fabrication.

Working directly on the machine also presents another way to creating the digital 3D model. *D-Coil* [115], for example, allows users to create a physical model using a handheld extruder and creates the digital model by tracking the extruder's movements.

2.1.3 Interactive objects and unconventional materials

Researchers not only investigated how to create interactive fabrication systems, but also how to fabricate interactive objects. One way to achieve this is to create systems that allow users to define space inside fabricated objects that includes electronic components, such as microcontroller boards [179], mobile devices [71], or electro-luminescent wire [138]. Using conductive filaments, users can add capacitive sensing capabilities to their 3D printed objects [20, 62, 143, 169]. Researchers also integrated other sensing technologies into 3D printed objects, including pneumatic sensing [48, 138, 156], acoustic sensing [139], printed loudspeakers [60], or optical sensing [137, 183].

Commonly available materials for 3D printers include plastics, resins, metals and plaster; laser cutting typically involves materials such as wood, paper, acrylic, and metal. Broadening this palette, researchers invented machines that go beyond these materials. They demonstrated systems that can produce 3D objects from stacks of paper [103], that can create soft objects from wool [53] or felt [114], that integrate fabric with plastics [119, 133], or print with biological material [173, 188].

2.2 ALTERING THE MATERIAL DISTRIBUTION

Changing the material distribution inside 3D objects allows for manipulating a diverse set of object properties. This includes changing the density of static objects, distributing rigid and soft materials to change how objects deform, or to make them fold from 2D to 3D structures.

2.2.1 Static internal material distribution

Researchers in HCI and computer graphics explored how to define the behavior of printed objects. In particular they optimized the rigid internal structure of objects in order to optimize the object's strength-to-weight

ratio depending on predefined static loads [26, 72, 184]. By defining void areas inside a 3D object, researchers effectively move the object's center of gravity in order for unbalanced shapes to stand [122].

Similar approaches are also effective for dynamic movements. For example, void areas on the inside of objects can be optimized in order to allow arbitrary objects to spin [9]. To give arbitrary shapes multiple stable poses, researchers integrated moving masses [123], which even allows for stable poses while the object is floating in water.

2.2.2 Defining how objects deform

More complex fabrication processes allow one to print with multiple materials and therefore to engineer for properties such as compliance and deformation. Multi-material 3D printers, such as the Objet Connex series, allows to deposit different materials at every voxel.

Localized compliance

Localized compliance allows mixing stiff and soft material arbitrarily across a fabricated object [68]. By applying techniques such as dithering, highly personalized compliance distributions for e.g., sockets of prosthetics [32], can be realized. Another interesting application area of objects with localized compliance is to realistically model how organs feel, which can be beneficial for medical training. For this purpose, Zehnder et al. [191] developed a computational approach to calculate the internal distribution of stiff silicone within soft silicone to resemble a specific feel on the outside of objects.

Reaching a target shape

Optimizing the material distribution can be used to match a pre-defined target shape based on some simple actuation. Skouras et al. [155], for example, used multi-material printing to create articulated characters. They computed the position of stiff and soft voxels within an object while taking into account locations where the material will be pulled. This allowed them

to match the target shapes of a figure. Pérez et al. [118] also matched target shapes, but by optimizing a rod network from a single material. Target-shape optimizations were also demonstrated for inflatable shapes [154], where the stretching of the material after inflation has to be taken into account. To control the resulting target shapes further, researchers printed rigid shapes onto pre-stretched material [42, 119]. The benefit of this approach is that the object is fabricated as a flat piece and only transforms into its target shape once the tension of the base material is released.

Shape-change and self-assembly

Self-assembly (or 4D printing) builds on a similar concept. Objects are designed to be fabricated flat and to fold up into a 3D shape once they are actuated [130, 162]. Typical forms of actuation include pneumatic actuation, the use of thermally responsive, light absorbing, or hydrophilic materials [39, 168].

Pneumatically actuated objects feature strategically placed pockets that are inflated after fabrication to transform and object from 2D to 3D [105, 109]. This type of actuation, however, requires a compressor and pipes. To avoid this, researchers are interested in using materials with properties that are easier to actuate. One example of a simpler actuation method is submerging objects in water. Researchers layered hydrophobic and hydrophilic materials as actuation mechanism [130, 177]. Once an object is submerged in water, the hydrophilic material soaks up water, expands and forces areas to bend. Placing such bending elements strategically allows for self-folding 3D objects.

Since actuation using water is very slow, researchers investigated thermally active materials [38] such as shape memory polymers [189]. Sandwiching a thermally active material between two layers of rigid materials allows controlling the bending angles and direction (mountain vs. valley fold) by generating parameterized gaps in the rigid material [3]. The folding of complex shapes requires to take the sequence of folding of parts into

account, as well as self-collisions. Mao et al. [75] solved this by modifying the composition of shape memory polymers to control the folding sequence. Recently, these capabilities were made available for simple 3D printers using commonly available materials (TPU, PLA, paper) [4, 174].

Hawkes et al. [47] used magnets to sustain the 3D shape of their thermoresponsive self-folding objects. Miyashita et al. [91] included magnets in a self-folding object to enable remote actuation.

While thermoresponsive materials are most common for self-assembly, researchers also investigated light absorbing materials as an actuation technique [163] and even biological materials [176, 187].

The benefits of such programmable, self-assembling objects span from easing packaging and transportation, to remotely actuated robots [91] that can even be ingested [92]. Furthermore, this type of actuation is used by researchers that are working towards entirely soft, autonomous robots [178].

2.2.3 Monolithic mechanisms based on deformation

Compliant mechanisms are deforming structures that allow implementing mechanisms by placing flexures. While traditional mechanisms use very stiff (rigid) parts that are connected by hinges to transform motion or forces, compliant mechanisms consist of (mostly) one part with the hinging parts being very thin [52]. Making the material thin increases its flexibility and allows for hinging behavior. Therefore, these parts are also known as flexures [164]. Since the movement is performed by deformation there is virtually no friction, no need for lubrication, and thus for maintenance. Since they consist of only one part, compliant mechanisms are well suited for miniaturization and are commonly used in micro-electromechanical systems (MEMS) [36].

To create such compliant mechanisms, researchers proposed a set of different topology optimization algorithms [153, 194, 195]. These algorithms essentially yield the material distribution on a voxel-level for simple mech-

anisms such as a gripper. However, topology optimization is also effective on a defined network of struts [73, 74, 140]. Recently, Megaro et al. [80] presented a system that converts a traditional linkage mechanism into complex compliant mechanisms (e.g., a hand with all joints implemented as flexures).

2.3 ENGINEERING NEW MATERIALS

Researchers in mechanical engineering and physics took such compliant structures to the micro- and nano-scale in order to engineer new materials.

They create microstructures that define the properties of materials. Such engineered material properties vary from ultra-lightweight structures to tuning of acoustic, optical, or electromagnetic wave lengths and mechanical properties such as shock absorption. We will focus on mechanical properties, as they are most relevant to this thesis.

2.3.1 Mechanical metamaterials

Metamaterials are understood as artificial structures with properties that are defined by their usually repetitive cell patterns, rather than the material they are made of [16, 27, 113].

Such 'cells' can be designed and engineered to undergo a desired deformation. When many unit cells are connected on a larger grid, the individual deformations of the cells together form unprecedented macro-scale properties. Overvelde et al. [107, 108] demonstrated the concept of metamaterials nicely: they showed how different unit cells, which were modeled as holes in a square with different shapes (circle, cross, star), influence the overall deformation drastically when the material is compressed.

Based on specifying the overall material properties by designing the internal structure, researchers developed many interesting materials, some of which we discuss in the following.

Ultra-lightweight, strong materials

Cells arranged on a regular grid were shown to be very successful in varying bulk material properties, i.e., properties that affect the entire material as opposed to localized properties. One interesting material property to engineer for is an object's density. A traditional material that has amongst the highest strength-to-weight ratio is ceramics. At the same time, however, it is very brittle. Meza et al. engineered a material that consists of octahedral-tetrahedral cells that are made of hollow ceramic beams [83, 84, 93, 193]. Only changing the geometry while still using ceramics as based material allowed the material to compress up to 50% and still recover from the deformation. This approach successfully reduced the brittleness and created an ultra-lightweight yet strong material, which is relevant, e.g., for air- and spacecraft.

Hierarchical metamaterials. Since such architected structures already showed such promising changes in properties, Meza et al. [85] went further and applied them in a hierarchical manner. They made the beams within their nanolattices from self-similar structures and found that introducing hierarchy enables a combination ultra-lightweight and recoverability. Moreover, they observed a near-linear scaling of stiffness and strength with density, which is promising for further miniaturization.

Pentamode materials. Milton [87] showed in 1995 that such structures can be engineered to be rigid in one direction but compliant in another. One especially notable type of material are so-called pentamode materials. While they are rigid against compression in all directions, they are very easy to shear, which makes them behave like solid fluids. Bückmann et al. later showed that such pentamode structures can be used as a 'unfeelability cloak' [18, 19]. They calculated the cells that surround an object to feel like as if there was no object enclosed in the material.

Volume-changing materials ('auxetics')

Another interesting material property is the expansion or contraction ratio. When conventional materials such as wood or metals are stretched, they compress in the orthogonal direction. The amount of this compression is referred to as the *Poisson's ratio* of the material. However, in the late 1980s, researchers suggested structures that would do the contrary, i.e., as a material is stretched vertically, it gets *thicker* horizontally. The material effectively changes its volume. Such materials are known as auxetic materials or materials with a negative Poisson's ratio. After Almgren [1] demonstrated a mechanical structure made from rods, hinges and springs that has a negative Poisson's ratio, Lakes [69] suggested a monolithic 3-dimensional cell based on a similar structure.

These structures are known as re-entrant honeycombs and inspired many researchers to this day. It is a well researched topic due to the unusual nature of this auxetic property. There are several articles that review the developments of auxetic materials [27, 65, 88, 131, 141] as well as an extended study of variations of cell geometries and the impact on the resulting Poisson's ratio [2].

Auxetic materials can be created using re-entrant structures, as mentioned above, or by rotation. Re-entrant structures are, e.g., honeycombs where 2 opposite vertices of the hexagon are pushed towards the center of the structure. This causes them to push their neighboring cells horizontally outwards when they are pulled apart vertically, thus expanding in both directions. The same principle has also been shown on other re-entrant shapes, such as stars. The second mechanism is to arrange cells on a regular lattice and connect them at their corners, such that the cells (e.g., rigid squares, triangles, etc.) rotate around each other to expand in both directions [40, 61]. Such auxetic materials can also be arranged in a hierarchical manner [95, 148], which allows for a better control over the degrees of freedom. These rotating cells can be produced by simply perforating sheets,

such as demonstrated by Shan et al. [150]. Researchers in computer graphics used the auxetic property of rotating triangles to create 3D objects with spherical curvatures (doubly curved) from a perforated sheet [67].

Besides above mentioned auxetic cells, there is also a very simple but unobvious cell geometry that is auxetic: a square cell with a circular hole [15, 100]. Shim et al. [152] show how, again, the lattice plays a big role to the extend of the auxeticity, with the square lattice contributing to the largest auxeticity compared to triangular, trihexagonal or rhombitrihexagonal tiling. Going beyond 2D structures, researchers also found such structures to work in spherical, thus 3-dimensional cells [7, 151].

Shock absorbing materials

Metamaterial structures were also designed to create materials that "pull" in the direction of compression rather than resisting it. These are known as 'negative stiffness' materials and were shown to be effective shock absorbers [29, 30, 46, 127, 132, 149]. Such materials are usually implemented by creating bistable unit cells. These bistable units usually consist of a curved, slender beam that is clamped between rigid elements. When that so-called bistable spring is pushed back, it snaps through to its second stable position, in which it then remains. Researchers used this snap-through to trap energy in the resulting deformation. This trapped energy is absorbed from the impact energy of an, e.g., falling egg [149]. All these shock absorbers are recoverable since they only need to be pulled out for the bistable springs to snap back into their first stable position. Frenzel et al. [35] even suggested beam configurations for such bistable unit cells to be self-recovering.

Hewage et al. showed that these negative stiffness properties can be combined with auxetic properties [49]. Rafsanjani et al. demonstrated that this bistable property can be used to simply lock the volume change of

auxetics [126]. Such bistable cells can also be used to pop out into the third dimension from a flat sheet and can thus be appropriated for shape-change [25, 45].

Wave propagating materials

Bistable structures have also been shown effective in propagating waves through materials. In general, the bistability depends on the spring's geometry and can be varied by its angle [14] and the curvature [124]. This will also vary the force that the snap-through produces [22].

Researchers investigated these parameters to create bistable structures that propagate waves through soft materials [101, 129]—an interesting property, as traditionally energy would simply dissipate within soft materials such as silicone. Asymmetry in terms of force output is an important characteristic in such wave propagating structures, as comprehensively studied by e.g., Kidambi et al. [63] and Wu et al.[185]. Since one bistable spring will trigger its successor, the force output of the first spring must be higher than the force it takes to trigger the second spring. The aforementioned works show how that can be achieved by pre-stressing the spring. As an example, when a 40 mm long spring is placed in a 39.5 mm wide base [63], the material of the spring is already under stress, i.e. it stores potential elastic energy. Raney et al. [129] show a similar strategy to tune their soft bistable springs.

While they also showed that forking such waves is feasible, Zanaty et al. [190] explored bistable structures that change directions and are directly coupled. They are programmed by varying the distance from the base structure by pushing the bistable spring further together or apart. Their approach is similar to our work on Digital Mechanical Metamaterials that we discuss in Chapter 5, not least because it was also inspired by Merkle's mechanical logic [82]. We, however, use printed blockers to implement our logic as opposed to varying the buckling of each bistable spring.

Origami- and kirigami-based metamaterials

By using origami, the Japanese art of folding paper into 3D sculptures, researchers showed that such folded structures also exhibit interesting properties. One of the first folding patterns that is today considered a metamaterial is the Miura-ori pattern [90]. It is an auxetic structure, that was originally designed as a packaging method for solar collectors to efficiently transport them into space. Once in space, they would be actuated by a simple 1D mechanism, yet fold up into large 2D membranes. Schenk et al. [142] demonstrated the versatility of the Miura-ori pattern by showing that transformations from 2D folded configurations to 3D volumes are realizable. They achieved this by stacking different layers of the pattern and even demonstrated self-locking configurations. Organizing the Miura fold in tubes enables stiff yet reconfigurable structures [34]. Such findings are important for, e.g., deployable structures as shelters in emergency situations or for deployment in space.

Kirigami, a variation of origami that includes cutting of paper, was also shown to be an effective technique for increasing the stiffness of a material [125]. While a simple sheet bends easily under load, cutting the sheet into a Miura kirigami sheet (a square array of mutually orthogonal cuts), makes the cells rotate vertically when the sheet is stretched, which increases the bending stiffness of the sheet. Kirigami metamaterials have applications in shape changing structures [102] or even as stretchable batteries [157].

Actuated and reconfigurable metamaterials

While many studies of metamaterials are concerned with the change in properties after the metamaterial was deformed, creating metamaterials that have built-in actuation mechanisms are of increasing interest. Generally, all mechanisms for actuated shape-changing objects or self-assembly, as we discussed above, can be applied to metamaterials, i.e., pneumatics, thermally-responsive or light absorbing materials, or magnetic fields [39].

Overvelde et al. [109] demonstrated a pneumatically actuated 3D metamaterial. They integrated air pockets in the edges of their cubic shearing cells. Inflating the pocket at an edge causes its connected faces to straighten, which in turn causes the cell to shear. Selectively inflating the air pockets allows for controlling the overall tilting direction(s) of the material. This material can be used as a reconfigurable wave guide [8]. Moving beyond the square cell, different prisms as unit cells influence the degree to which the material can be reconfigured [110]. The deformation of the overall material can be used as simple soft robots [186]. Actively creating a vacuum in auxetic pores causes the material between the pores to rotate. Attaching external objects like little beams as arms or paddles allows creating actuated objects such as grippers or swimmers, respectively.

The elastic properties of metamaterials can also be configured using magnets [44]. By embedding electromagnets into the beams of the unit cells of a metamaterial, the stiffness of those beams can be actuated; when the magnets are switched off, the beams can separate, when the magnets are on, the beams stick together, forming a stiffer beam with double the thickness. This makes the overall material stiffer.

2.3.2 Varying cells across an object

Most of the metamaterials that we have discussed so far actually use only one unit cell within one material. Only recently, the potential of metamaterials with varying the cells were explored. Mirzaali et al. [89] varied honeycomb cells over the whole spectrum from re-entrant auxetic to conventional cells, in order to match a curved 2D shape after expanding the material. Such mixing of 3D honeycomb cells are also promising for implants, as such 'meta-implants' can improve implant longevity [66]. Similarly, aforementioned mechanical cloak [18, 19] varies the cell parameter close to the object that is being hidden from the touch of users.

Researchers varied a large number of cells within a 3D model by adopting a computational approach [17, 24, 111, 112, 145]. They varied the parameters of their cells to cover a large area on the Young's modulus/Poisson's ratio spectrum, where the Young's modulus describes how elastic a material is. By selecting the appropriate cells to vary the elasticity locally, they create deformable figures with a prescribed deformation. Such generated elastic cells can also be achieved using multi-material 3D printing and filling the otherwise void areas with soft material [196]. This eliminates the need for support structures, but it also increases the stiffness. While Coulais et al. [31] did not generate the cells, they varied a set of simple cells as to redirect forces and pop out a hidden texture (e.g., a smiley face) when the material is under compression. Recently, researcher even created metamaterials that are initially flat and can curl up in 3D once actuated [106].

Since metamaterial structures also have a certain aesthetic to them. Schumacher et al. [146] recently created an exploration tool that gives designers a choice of different 2D cells and lattices for an elastic property, such that designers can choose the most visually pleasing one.

These works achieve fine-grained differences in Young's modulus/Poisson's ratio by parameterized cells. We, however, use *different types* of cells, where the spatial layout within the metamaterial is key to their functionality.

2.3.3 Tools for creating metamaterials

As the structures that enable metamaterials are growing, the tools to create these do too. Still, many researchers in the fields of mechanical engineering, physics, or material sciences create their geometries using professional computer-aided design tools and commercial simulation packages such as

ABAQUS[1] (e.g., [33, 61, 73, 85, 107, 149]), ANSYS[2] (e.g., [134, 175]), or MATLAB[3] (e.g., [14, 73, 175]). These tools provide complete freedom in design and analysis and are thus preferred by expert users in research.

In order to reduce the effort of modeling microstructures, researchers proposed procedural specification of materials [23, 170, 171] and visual editing tools [51, 86]. Additionally, companies recognized the potential of engineered microstructures and offer software package for specific application domain such as saving material in mechanics [5], for medical use [6], or precision mechanisms [166].

2.4 OUR CONTRIBUTION: ENGINEERING METAMATERIAL *DEVICES*

Our work builds on the advances in fabrication and specifically in metamaterials. The concept of metamaterials is particularly interesting, because it enables us to populate a material with properties of our choosing. Our work is still very distinct from previous work. Previously, metamaterials focused on achieving one property (e.g., change in volume, or energy absorption). We, however, use this cell-based concept to place different properties across the material in such a way that together they form a complete device, rather than a block of material. This enables us to create devices, the functionality of which is solely defined by the cellular microstructure, thus require no assembly or lubrication.

[1] https://www.3ds.com/products-services/simulia/products/abaqus/
[2] https://www.ansys.com/
[3] https://www.mathworks.com/products/matlab.html

3

Metamaterial mechanisms

As discussed in Chapter 2, the rise of digital fabrication machines motivated researchers to explore interactive tools that allow users to define the outer shape of physical objects, as well as their internal material distribution. In engineering disciplines, researchers started to push internal structures even further and created objects that consist internally of large numbers of 3D cells organized on a regular grid [145]. Since these objects allow each cell to be designed differently, the resulting objects literally offer thousands of degrees of freedom. These types of structures have also been referred to as *metamaterials*. Metamaterials are artificial structures with mechanical properties that are defined by their usually repetitive cell patterns, rather than the material they are made of [113].

So far, metamaterials have been understood as materials. The main contribution of this chapter is that we want to think of them as machines.

In this work, we push the concept of metamaterials further by creating objects that allow for controlled directional movement. This allows users to create objects that perform mechanical functions. Our objects thereby implement devices that transform input forces and movement into a desired set of output forces and movement—also known as mechanisms.

3.1 OVERVIEW OF METAMATERIAL MECHANISMS

Figure 3.1a shows an example of a metamaterial mechanism: a door latch mechanism. Its interior is a regular grid of 3D cells; however, the cells are of different types. Figure 3.1b shows how applying a force causes the cells to deform in a controlled way, thereby performing the intended mechanical function. In the example, rotating the door handle causes cells inside of the object to deform, ultimately pulling the latch towards the left and thereby unlocking the door.

While most of the object consists of *rigid cells* (cells that are reinforced with a diagonal), the object also contains several rectangular regions of cells that lack such a diagonal reinforcement. These are the key to creating mechanisms, as they are able to deform in a very specific way: when subjected to an external force, these cells shear and thereby apply a force to their neighboring cells. We will discuss throughout the remainder of this chapter how this basic principle allows the creation of mechanisms.

Metamaterial mechanisms are simple. While the traditional door latch mechanism shown in Figure 3.2 consists of several parts, including an axle, bearings, springs, etc., the metamaterial door latch in Figure 3.1 consists of a single block of material, as it is groups of cells inside the object that perform the mechanical function.

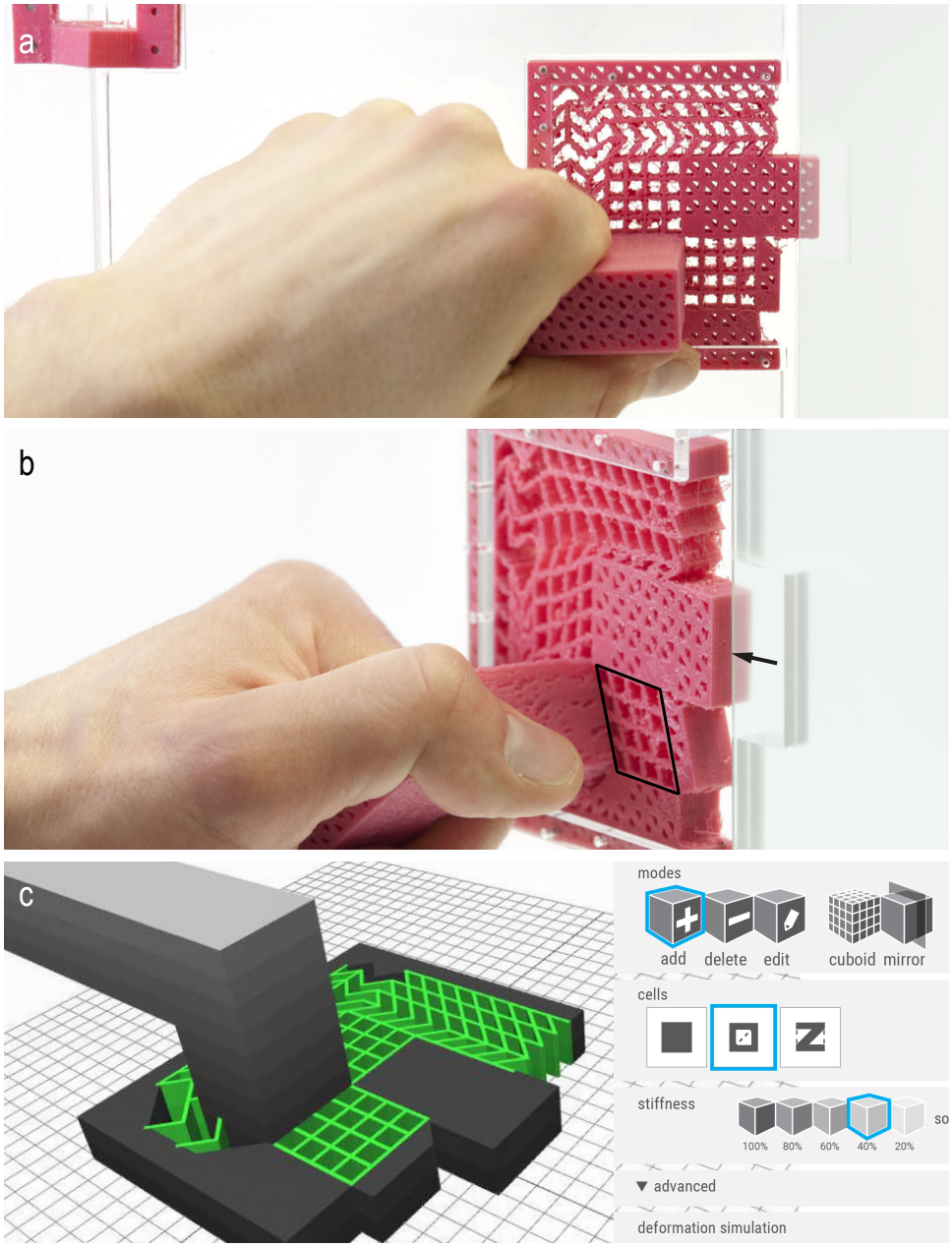

Figure 3.1: (a) This door latch is implemented as a metamaterial mechanism; it consists of a single block of material based on a regular grid of cells that together implement handle, latch, and springs. (b) Turning the handle causes the central hinge array to deform and to pull the latch inwards, which unlocks the door. (c) We created this mechanism in our custom editor. Here, we placed two hinge arrays that mechanically couple the handle to the latch, and cells that couple to the doorframe.

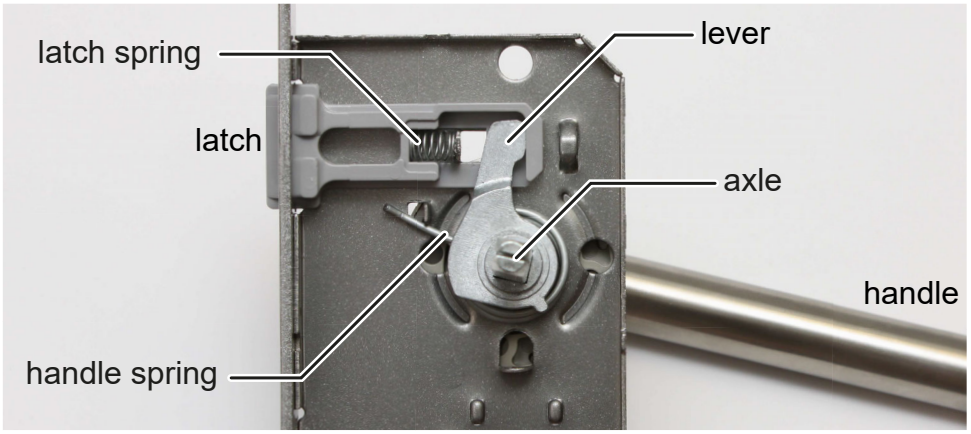

Figure 3.2: This traditional door latch design consists of many parts, thus requires assembly.

While in previous work metamaterials were typically generated by scripts [100, 113], creating mechanisms requires a dedicated design/engineering process that we argue is best performed by means of an interactive editor. Figure 3.1c shows a preview of the custom 3D editor we created specifically to allow users to create and modify metamaterial mechanisms. It contains a range of functions that help users assemble specialized cells into basic mechanisms and to assemble such basic mechanisms into more complex mechanisms and simple machines. Using this editor, we have created a series of demo objects including the door latch from Figure 3.1, a Jansen walker, or a functional pair of pliers. Our examples were printed on an *Ultimaker 2* 3D printer using *NinjaFlex* filament (in pink). The basic mechanisms (hinge, four-bar) were laser cut from 3 mm rubber foam (in black).

3.2 CONTRIBUTION, BENEFITS, AND LIMITATIONS

Our main contribution is the concept of metamaterial mechanisms. Our main software contribution is a specialized editor that allows users to create them. We extend the research field of metamaterials by contributing a general-purpose approach to creating mechanisms. Mechanisms are a new genre of metamaterial structures that is of higher complexity and that

exploits more degrees of freedom than previous work in this field, and that allow metamaterials to tackle problems they have traditionally not been able to address.

Compared to traditional multi-part mechanisms, metamaterial mechanisms offer several benefits. (1) The resulting devices consist of a single part. They can thus be created using particularly simple fabrication processes, such as single-material 3D printers (e.g., FDM printers). (2) As they consist of a single piece, they require no assembly. (3) Since the movement is performed by deformation there is virtually no friction, no need for lubrication, and thus for maintenance [52].

However, the resulting designs are also subject to limitations. Adding more cells increases the stiffness, and as a result, metamaterial mechanisms are not suitable for mechanisms that are to be operated with very small forces. Furthermore, our approach is unable to produce continuous rotation. Objects such as the Jansen walker, for example, thus require a separate axle. Also, cell designs are limited by the quality of the 3D printer. In particular, shear cells work best if their internal hinges are thin, which requires high-resolution 3D printers. Finally, while our editor vastly simplifies the creation of metamaterial mechanisms, any type of mechanical engineering requires experience—and metamaterial mechanisms are no exception here.

3.3 MEMBERS & HINGES BASED ON CELLS

The mechanism behind the door latch in Figure 3.1 ultimately consists of rigid regions (such as the latch itself), which we will refer to as members and compliant regions that implement hinges (such as the region right of the handle). Members and hinges both consist of cells on an evenly spaced grid. This is the general schema behind metamaterial mechanisms. In this section, we take a closer look at these two basic elements.

3.3.1 Members from cells

Members consist of rigid cells. A rigid unit cell is reinforced with a diagonal, as shown in Figure 3.3. The diagonal reinforces the cells against shear forces and thus makes them rigid. Adding two diagonals is possible, but not necessary, hence we use only one diagonal for our member structures in order to save material.

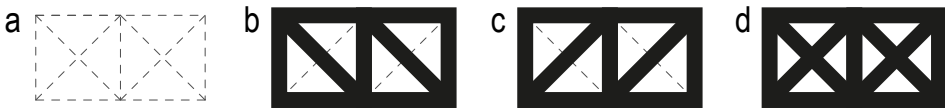

Figure 3.3: (a) The basic grid structure allows reinforcing rigid member cells (b-c) with one diagonal or (d) two diagonals.

While we will focus our discussion on only two types of cells, namely stiff member cells and shearable hinge cells, our research can be combined with any other system of metamaterials already done as long as they are on a cube grid. Just to pick one example, we can make cells soft or hard [28] by weakening their beams so as to allow the cells to compress more easily, or reinforcing their beams to make the cells stiffer (Figure 3.4).

Figure 3.4: Member cells of increasing stiffness.

3.3.2 Hinges

We can create a (naïve) hinge by connecting two cells diagonally as shown in Figure 3.5a. Here the material connecting the two cell blocks forms a thin flexible hinge made from the same material as the two rigid members it connects. As common for this type of contraption, we will refer to it as a living hinge.

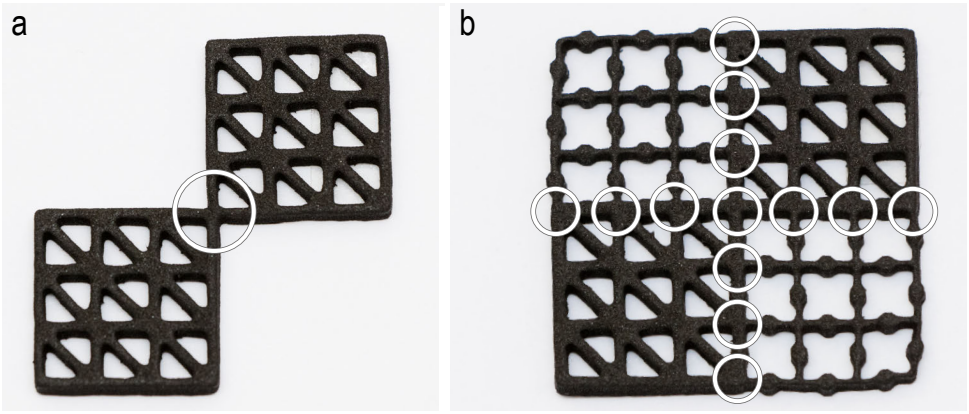

Figure 3.5: (a) A naïve living hinge, (b) reinforced with two arrays of hinges. To showcase the deformation, we laser cut these structures from rubber foam.

Next we add reinforcement to the hinge. Metamaterial cells can, at least in theory, be arbitrarily small—the hinge connecting two cells can thus arbitrarily thin, making mechanisms based on such naïve living hinges fragile, i.e., pulling them apart with very little force will rip them off. We address the issue by extending the living hinge into an array of living hinges, as illustrated by Figure 3.5b. We call them hinge arrays. Figure 3.6 shows that the array flexes in concert with the main living hinge.

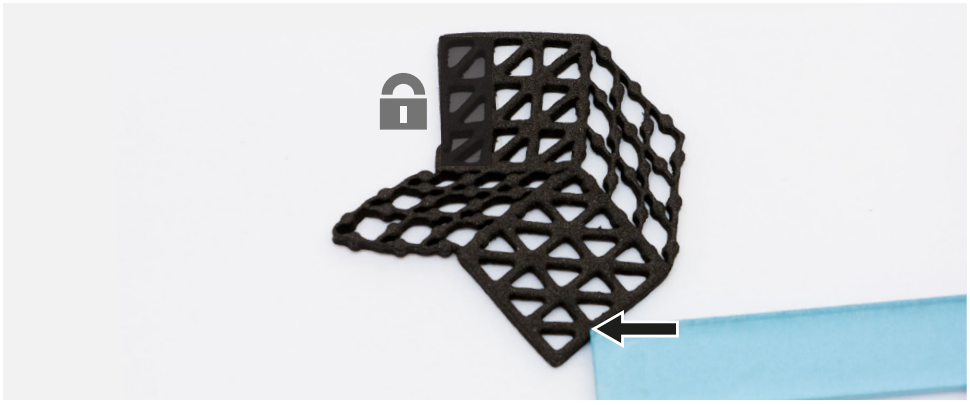

Figure 3.6: The reinforced hinge in action. Note how the hinge array deforms as one while the rigid members remain undeformed. (Cells with gray tinted backgrounds are anchored to the ground, as indicated by the lock symbol. We are pushing in the direction of the arrow using the blue rod).

The hinge array on the inside of the rotation complies by compressing in a shearing motion; the hinge array on the outside complies by stretching in a shearing motion. All hinges together perform a rotation around the original living hinge.

However, when this reinforced hinge is stretched as shown in Figure 3.7a, its hinge arrays evade the tension by shearing. As a result, all tension rests on the central living hinge, causing it to break. As shown in Figure 3.7b, we address this by adding a border of rigid cells to the hinge arrays. These borders force all living hinges to follow the motion of the central living hinge, thereby support the main hinge by assuming part of the load, which distributes the tension across all the hinges.

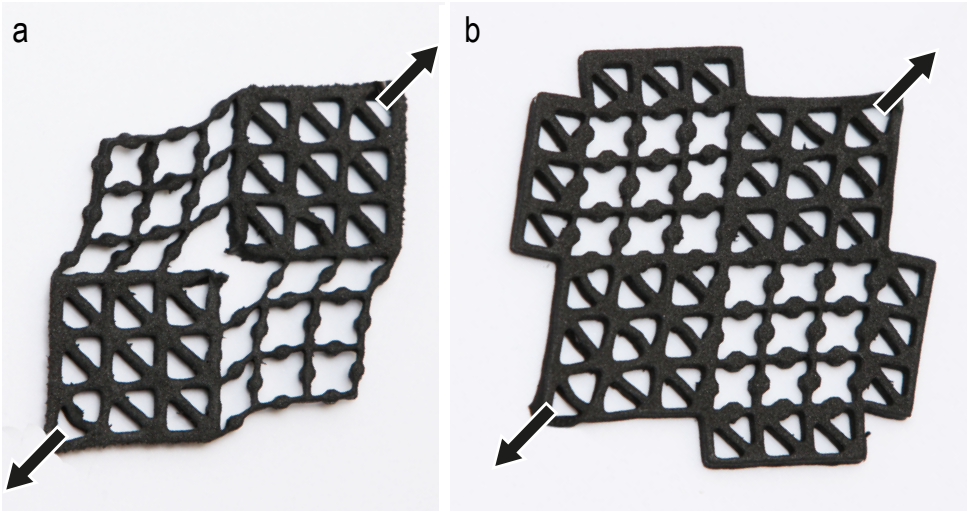

Figure 3.7: (a) When this hinge array is stretched as shown, the central hinge breaks. (b) We address this by adding borders of rigid cells; they help distribute the tension more evenly.

Tensile strength increases linearly with the diagonal of the hinge array, because tension applies to the shortest path of hinges to tear in order to separate the two members. The tensile strength of the hinge array in Figure 3.7, for example, is 7× larger than of a single hinge. Figure 3.8 shows the resulting stress distribution after reinforcement.

Hinge arrays also allow tuning the bending stiffness of a hinge, e.g., to make sure the door latch in Figure 3.1 offers appropriate resistance when pushed down, as well as sufficient force to push the handle back up. Bending stiffness increases proportional to the surface of the hinge array, i.e., $(width + 1) \times (height + 1)$, as all individual hinges need to be bent. The hinge array in Figure 3.7, for example, uses 2 hinge arrays of $(3+1) \times (3+1) = 32$ hinges, thus it takes 32 times more force to bend it than the single hinge in the center.

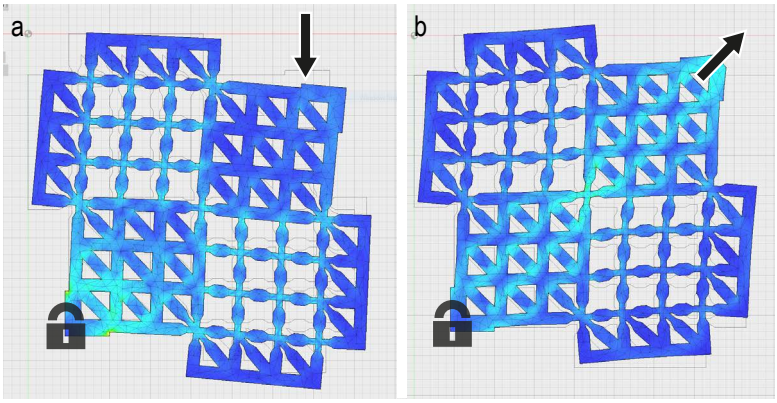

Figure 3.8: Resulting stress distribution on (a) rotation and (b) tension after reinforcement. (Autodesk Fusion 360; bright colors indicate high stress)

3.3.3 Four-bars implement parallel motion

We use the idea of connecting rigid members by hinge arrays to construct a shearing frame of four links, also known as *four-bar* linkages. They consist of four members attached to the *opposite* sides of a hinge array (unlike the reinforced hinge, which consists of two members attached to *adjacent* sides of a hinge array), which keep the two opposing members parallel. Note how in Figure 3.9 the rigid member that is pushed to the left rotates, while the top member moves in parallel with the rigid area underneath it.

In Figure 3.10 we exploit this property of the four-bar in order to create a functional pair of pliers. The hinge array in the center acts as a four-bar that connects the right handle with the left bracket *and* the left handle with the

right bracket, yet allows the two sides to move with respect to each other—somewhat similar to how the axle in a traditional pair of scissors allows the two handle and blade elements to move with respect to each other.

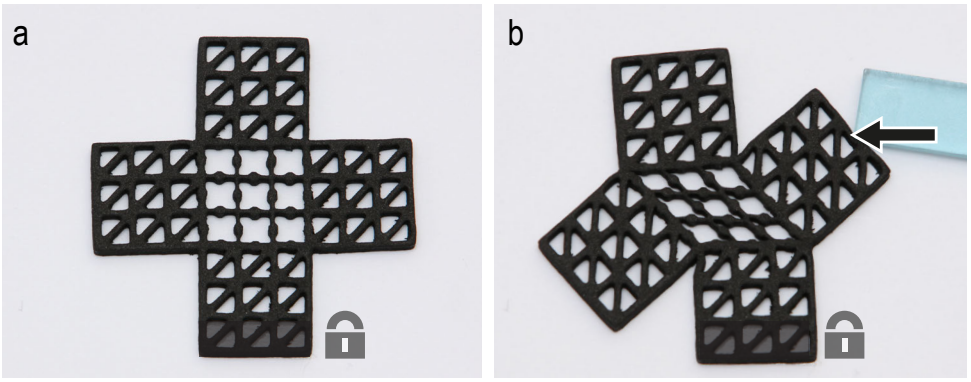

Figure 3.9: A metamaterial four-bar.

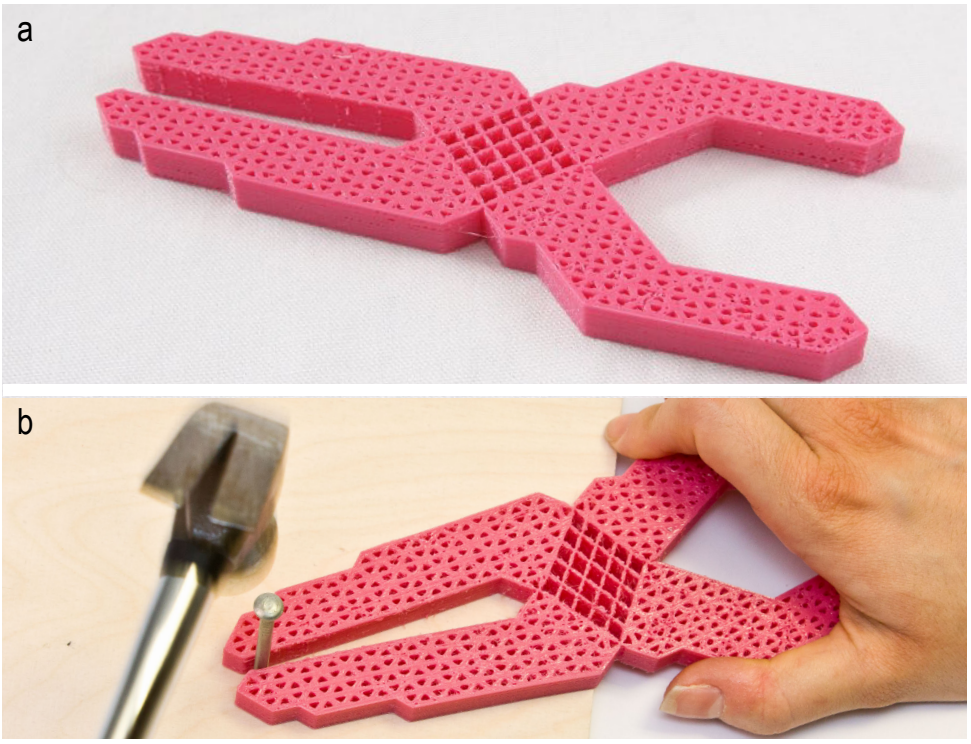

Figure 3.10: (a) This pair of pliers is based on a single metamaterial four-bar. (b) When the user applies a force to the handles, the hinge array in the center transmits this force to the brackets, and the pliers close.

3.3.4 Cascades of hinge arrays

We chain multiple four-bars to create a scissors mechanism, which is a linkage that arranges links in a criss-cross pattern and is used. Figure 3.11 shows how we implement a pantograph by chaining multiple metamaterial four-bars. The pantograph holds two pencils. While one pencil is moved by the user to draw, the second pencil replicates the user's drawing.

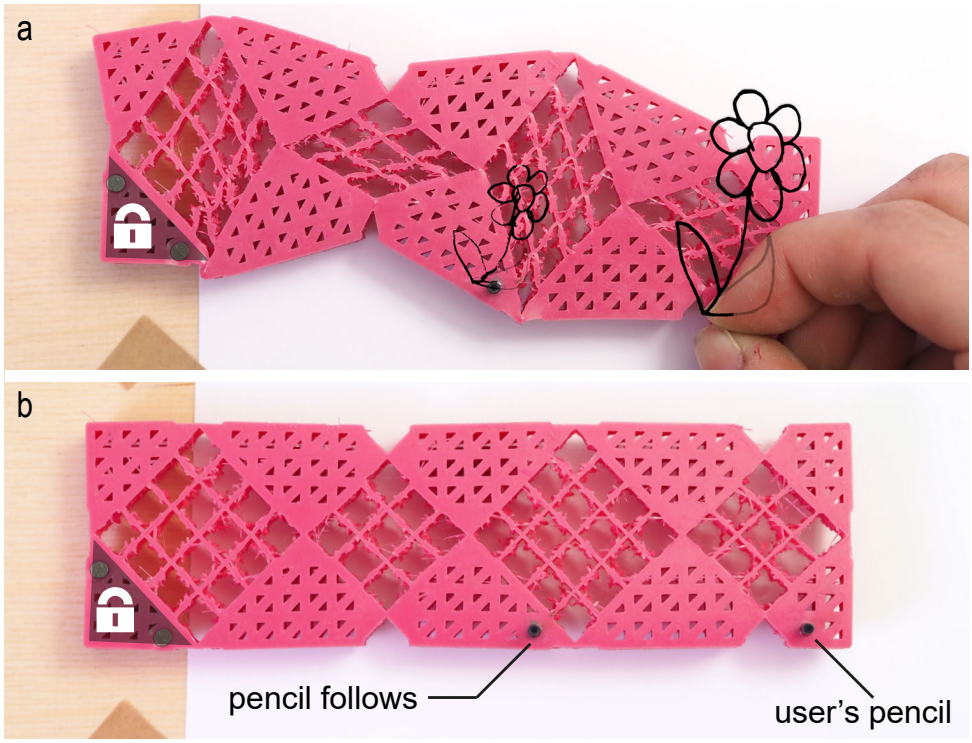

Figure 3.11: The user draws a flower using (a) our metamaterial pantograph, which (b) replicates the flower as the user is drawing.

3.4 THE KEY: THE SHEAR CELL

So far, we only introduced the rigid cell. However, without explicitly mentioning it, the previous section introduced a second type of cell: a cell designed to shear. Unlike the rigid cell, this *shear cell* is designed to deform when a force is applied, more specifically to *shear* (Figure 3.12).

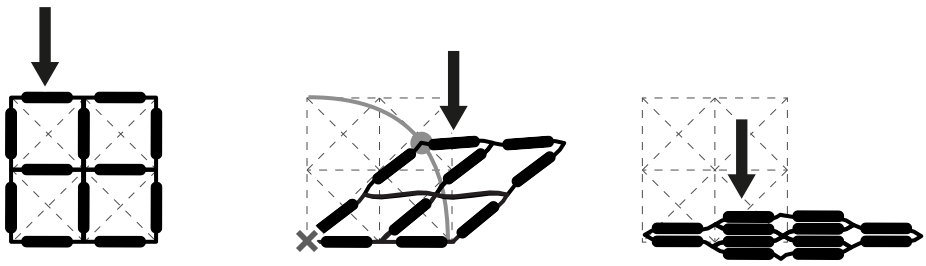

Figure 3.12: When a shear cell in this 2×2 block is subject to compression forces, it complies by shearing on a circular trajectory until its members are packed tightly.

As illustrated by Figure 3.12, shear cells compress when a force is applied. The cell itself consists of miniature members and miniature living hinges—similar to the macroscopic structures we build from them. When a force is applied, the hinges start to bend, while the members largely maintain their shape (increasing the thickness of a beam results in a cubed increase in stiffness). The shear cell resists the external force based on the springiness of its hinges, until eventually the members touch and the resulting tightly packed block of cells is not compressible anymore apart from the compressibility of the material itself.

Since shear cells are based on an unit cell of our grid, they are always oriented along the cardinal directions. To allow handling forces and shearing along additional orientations, we introduce rotated hinges. As illustrated by Figure 3.13, we obtain a rotated design by combining groups of 2×2 cells, each of which contains (only) a diagonal.

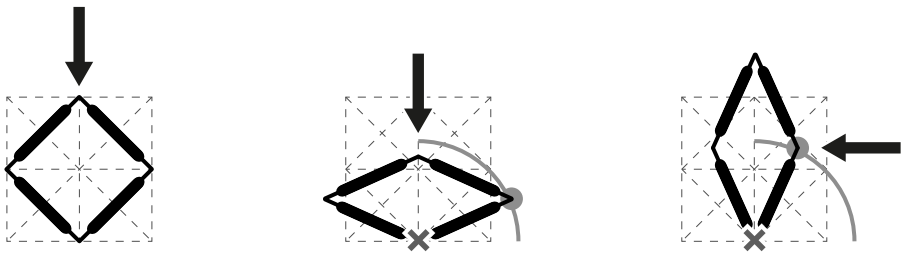

Figure 3.13: The rotated shear cell.

Combining regular and rotated hinges allows us to engineer additional mechanisms, such as the leg pair of a Jansen walker mechanism that is shown in Figure 3.14.

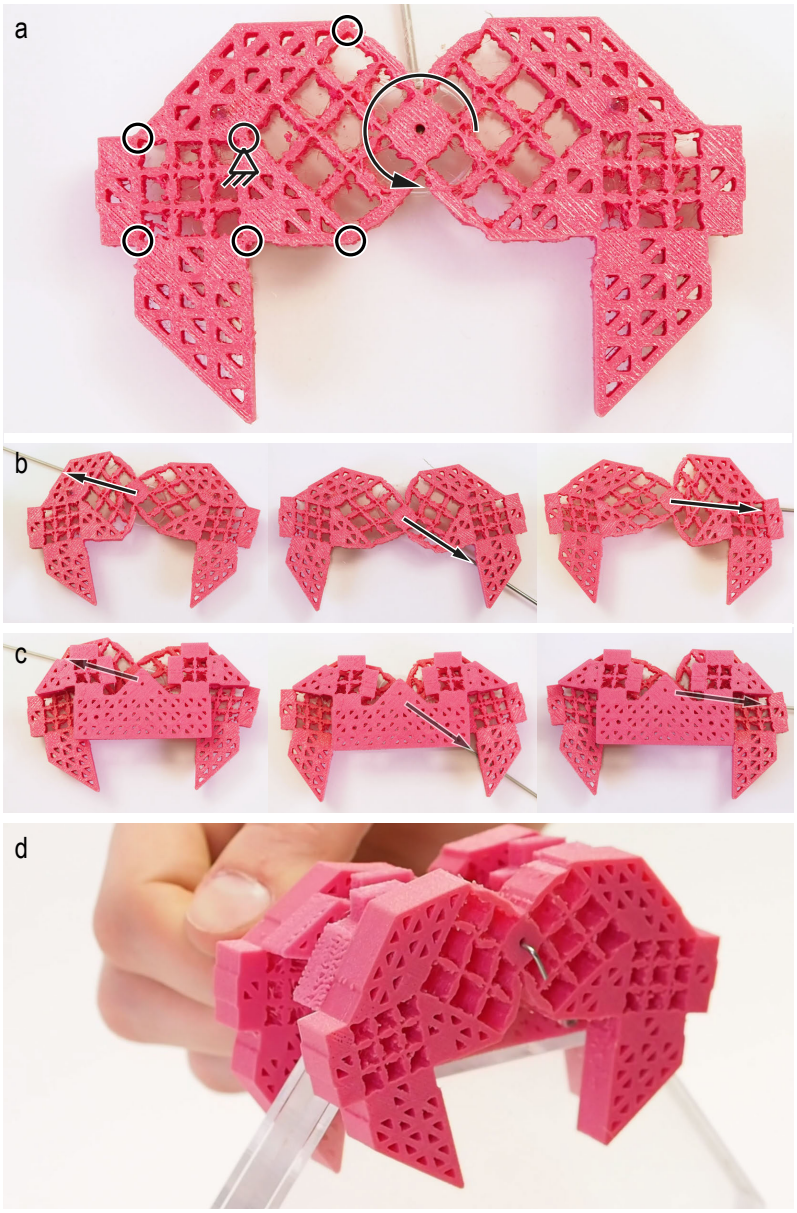

Figure 3.14: (a) Each leg of the Jansen walker has 6 hinges. (b) As the center is actuated by a crank, the legs deform in a walking motion. (c) We constrain two hinges to rotate, but not translate, using a layer of hinges. (d) The complete walker.

Each leg consists of 6 hinges that make up the organic walking motion when the center is actuated on a circular path using a crank. One hinge per leg, however, needs to be constrained so as to only rotate, not translate. We implement this by adding a layer of two metamaterial hinges for the two legs that are connected to stay at constant distance but decoupled to move independently from each other.

Finally, we have created a third type of shear cell by leaving out one of the Cartesian members. It is shown in Figure 3.15 and we refer to it as *z-cell.*

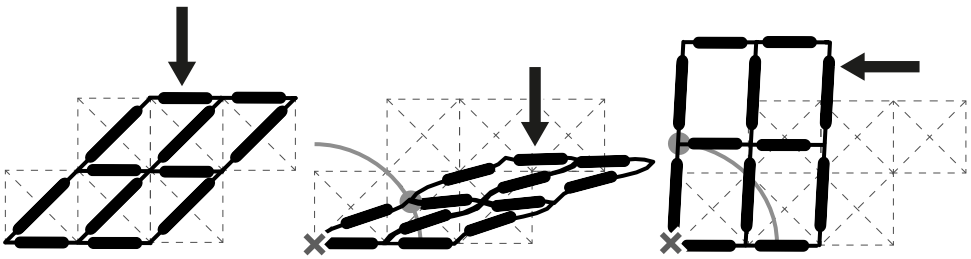

Figure 3.15: "z-cells" shear on a larger circular path than regular shear cells, which is also rotated by 45°.

Z-cells are very similar to regular shear cells, except that they are "pre-sheared", which results in three differences: (1) they have an asymmetric shearing behavior (range of 135° vs. 45°), (2) they expand as they shear against their pre-sheared direction, and (3) compress in the other direction.

Figure 3.16 shows a switch based on z-cells. The button stays parallel while going down to hit the contact point reliably. A similar switch implemented from only soft and hard cells requires making very thin beams to allow the switch to deform. However, this does not implement a well-defined directional movement and closing at the contact point is not ensured.

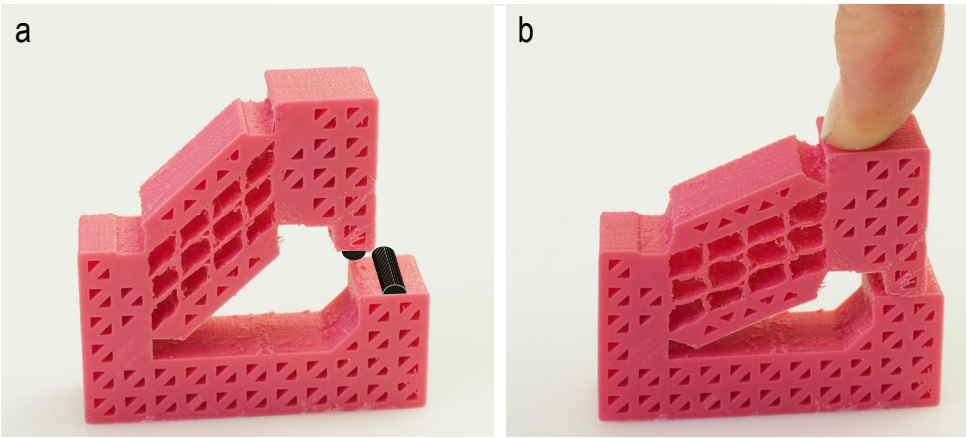

Figure 3.16: (a) This switch is designed based on z-cells, which allow it to compress in a controlled motion and (b) close reliably at the black contact point.

3.4.1 Padding

Metamaterial mechanisms enable directional movement of regions of material, i.e., they occupy space when they are in their deformed state that was not occupied in their undeformed state. Most of our examples—the pliers, the walker, or the pantograph—are self-contained and can deform independently. However, the door latch is not self-contained but embedded in a rigid door. This requires appropriate clearance for the respective region to move into.

We surround the door latch shown in Figure 3.17 with what we call padding areas. Padding areas consist of cells that are not intended to transmit any forces, but rather to receive the forces exerted by the main mechanism. The padding serves two purposes here: (1) to connect the movable door latch with the rigid door and (2) to increase its stability, i.e., to prevent the mechanism from buckling out of plane and add support against operation in unintended directions. In case the handle is pushed up, the z-cells above the latch prevent the latch from moving outward (to the right).

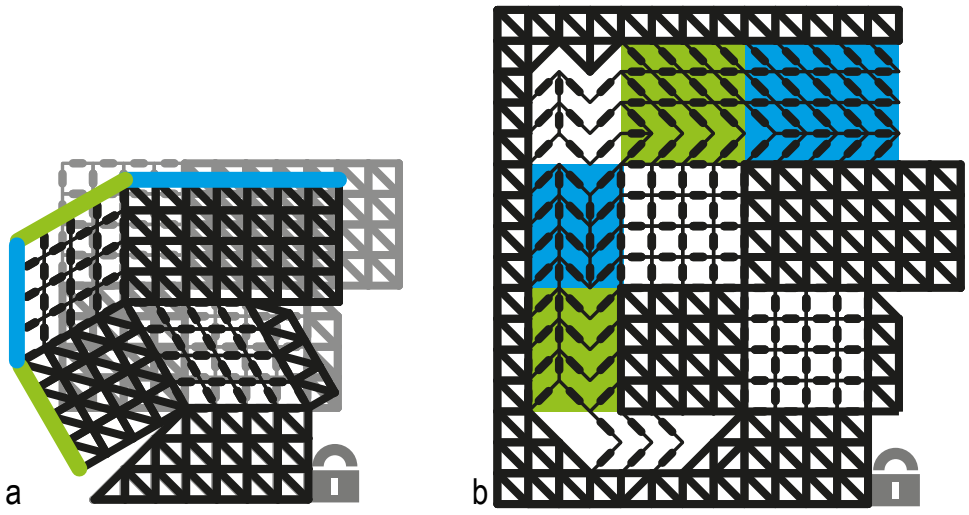

Figure 3.17: (a) When the door latch deforms, the edges labeled in blue remain parallel, while the green edges rotate. (b) Consequently, we reinforce the green regions using padding cells that allow for translation and rotation deformation, while we reinforce the blue regions to only allow for translation.

Figure 3.17a shows that after the door latch mechanism is deformed we see two types of transformations on the outside edges: (1) a parallel movement that is indicated by the blue line and (2) a rotational movement of the edge indicated by the green line. Additionally, all the edges undergo a vertical and horizontal translation. For all outside edges we need to allow translation and for some we need to additionally allow rotation. Therefore, the latch mechanism in Figure 3.17b uses two types of padding cells, chosen to allow for the intended movement, yet to suppress other types of movement.

Z-cell padding absorbs parallel movement, like the movement of the latch. Z-cell padding allows a member on one side to translate in 2D with respect to a member on the other side and keeps the two members parallel. Since z-cells shear on a circular path that is rotated by 45°, they allow for a vertical extension of $\sqrt{2}-1$ cells and a horizontal displacement of 1 cell. We simply stack enough layers of z-cells to compensate for the downward and inward movement of the latch.

Note that we added one row of z-cells that is flipped. Figure 3.18a shows that this additional row is necessary since the shear cells underneath the latch and the z-cells above the latch travel on opposing circular trajectories and cannot pass each other. The flipped row of z-cells however moves upwards (Figure 3.18b) and gives space for the lower row of z-cells to pass and thus the latch to move.

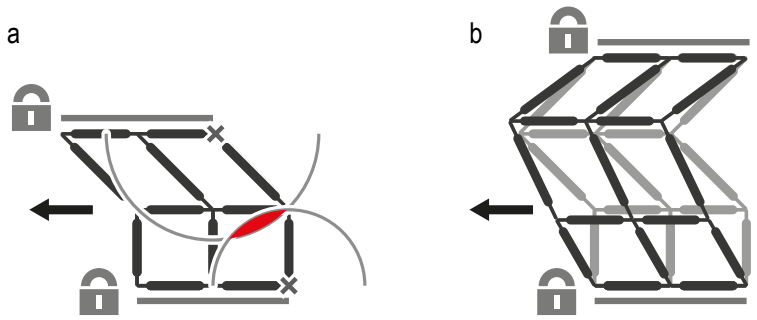

Figure 3.18: (a) The shear cells cannot pass each other (red area), (b) therefore we add a flipped row of z-cells that compress and give space for the other shearing cells to move.

V-cell padding absorbs rotation. V-cells are similar to the flipped z-cell group, but they additionally allow two opposing members to rotate, as shown in Figure 3.19. We add this degree of freedom by removing the vertical beam from the z-cells. We use v-cells in combination with z-cells on the rotating edges illustrated by Figure 3.17b.

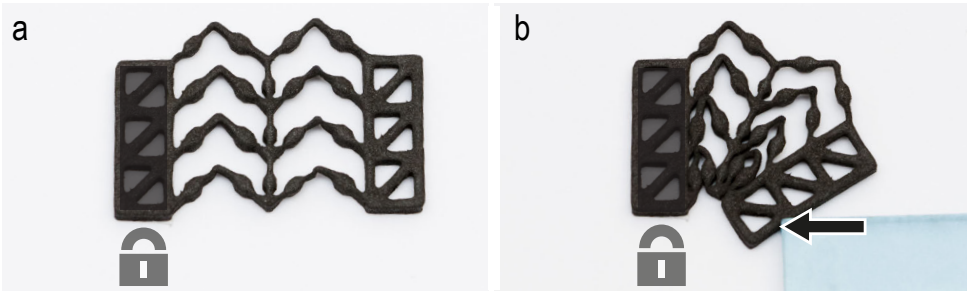

Figure 3.19: V-cells allow for rotation, as well as for compression and extension.

3.5 HIERARCHY OF METAMATERIAL MECHANISMS

In this section, we presented the main elements of metamaterial mechanisms. These elements form the following five-level hierarchy.

1. *Cells* are the lowest-level element of metamaterial mechanisms and perform an elementary mechanical function. We presented shear cells, which implement four-bars on a cell level, as well as z-cells.

2. *Compound cells* still perform an elementary function, yet are implemented using multiple physical cells, such as the rotated shear cell or the v-cell.

3. *Basic mechanisms* are created by repeating cells, in particular the reinforced hinge mechanism and the four-bar mechanism.

4. *Compound mechanisms* consist of multiple basic mechanisms, such as the scissors mechanism implemented in the pantograph.

5. *Machines* perform a mechanical function and consist of one or more basic or compound mechanisms, such as the door latch (Figure 3.1 and Figure 3.17), the pliers (Figure 3.10), the Jansen walker (Figure 3.14).

3.6 EDITING METAMATERIAL MECHANISMS

To allow users to design, fabricate, and test metamaterials containing mechanisms we implemented the specialized editor shown in Figure 3.20. The main intent behind it is not only to make the editing process more efficient than the more traditional script-based editing, but also to provide users with an overview of their design, encouraging design by trial-and-error. Our editor is based on interaction techniques known from voxel editors (such as [51]). However, in addition our software also offers specific supports for creating mechanisms, such as tools for drawing hinge arrays, etc.

In order to allow users to validate their designs, the editor also allows them to apply forces and see how the object deforms in order to then refine their design directly inside the editor, before exporting to the 3D printer.

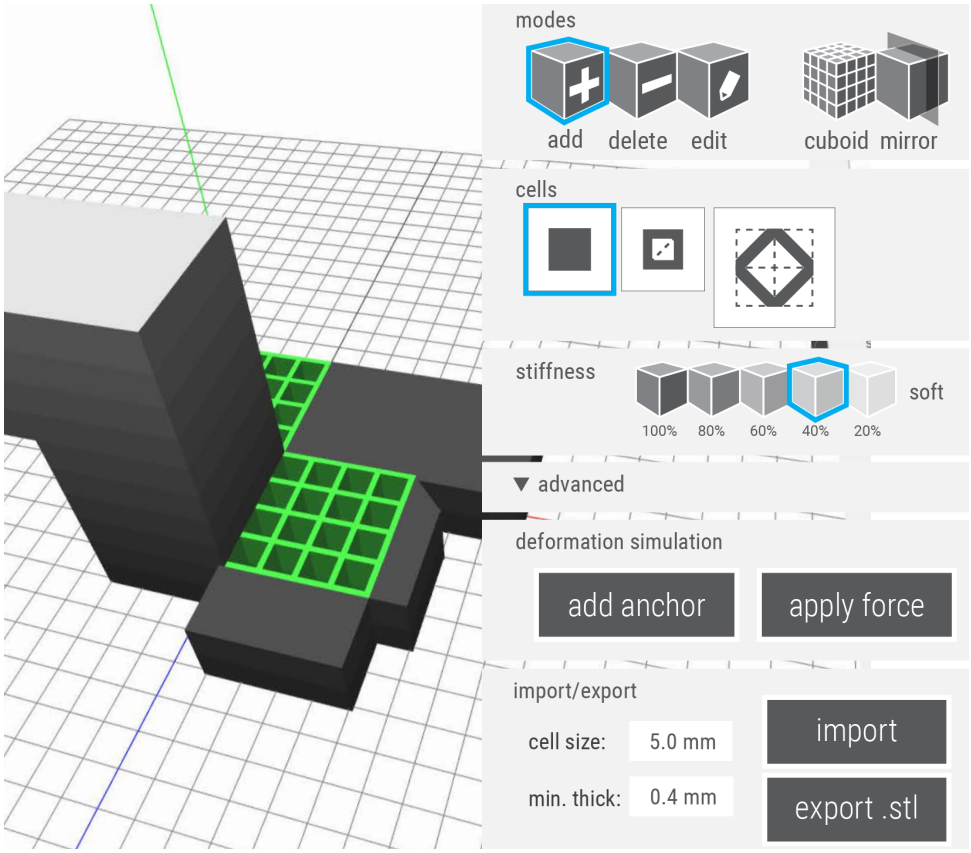

Figure 3.20: Our editor allows users to edit and simulate the deformation of metamaterial mechanisms interactively.

3.6.1 User interaction

Figure 3.21 illustrates how users create the door handle example. (a) Users start by creating a block of rigid cells using the *add brush*. They can remove cells using the *delete brush*. Here, users utilize the tool in *cuboid mode*, which allows them to draw a filled rectangular region at once by just drawing the diagonal. (b) By adding another two cuboids on top, users create the handle.

(c) Next, Users select the *shear brush*. Still in cuboid mode, they paint the central hinge array using a single drag interaction, which causes rigid cells to turn into shear cells. Even though the block of material that users painted on is two cells high, the shear brush paints cells all the way through—as they can tell from the sidewall now being all green. This is one of the features of this brush: since shear cells backed by rigid cells would still be rigid, thus have no effect, the shear brush always cuts shear cells through the entire object.

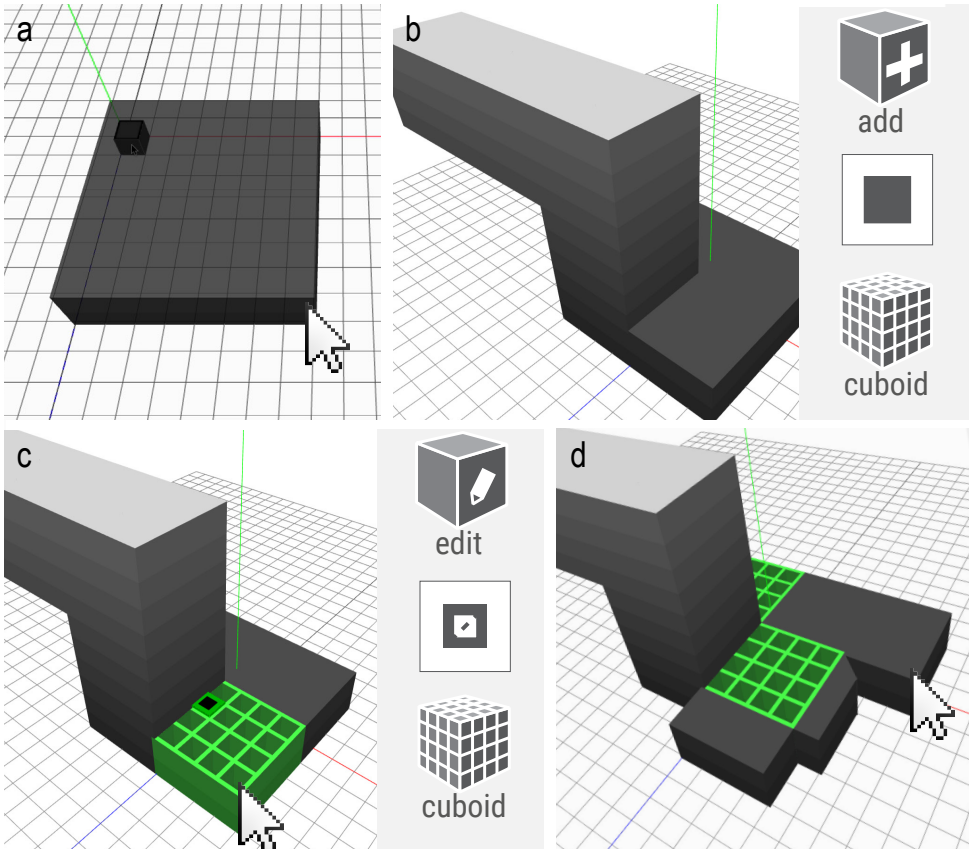

Figure 3.21: Walkthrough of the creating door latch mechanism. The UI elements on the right show the active tools for the respective interaction steps.

3.6.2 Simulating deformation

Users now verify their design directly from within the editor, as illustrated by Figure 3.22.

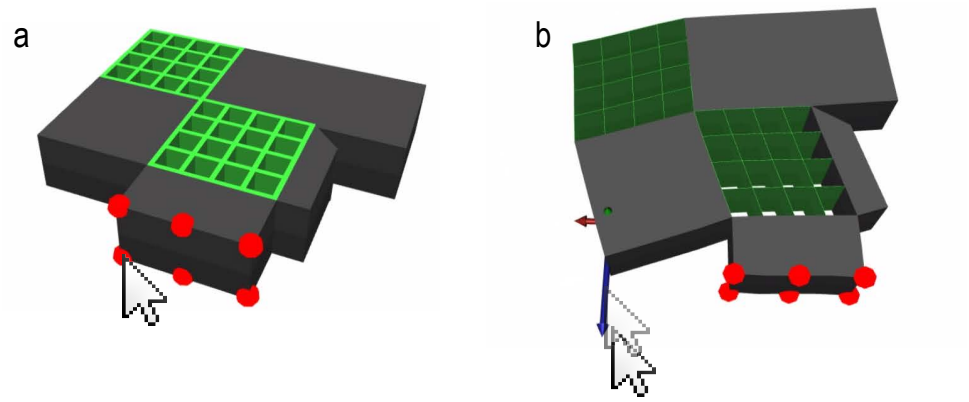

Figure 3.22: To simulate the deformation in real-time in the editor, (a) users set anchor points and (b) adjust forces using the force tool.

(a) They select the anchor tool and use it to place a few anchor points at the bottom, indicating that the door latch is rigidly connected to the doorframe there. (b) Now, they use the force tool to apply a force to the door handle. Users attach a force arrow to one of the handle's cell vertices. As users are building up the force by dragging the force tool the system already responds by showing the resulting deformation of our door latch.

3.6.3 Multiple dimensions

While the door latch mechanism actuates in only two dimensions, our editor also supports placing mechanisms in 3 dimensions. The latch mechanism shown in Figure 3.23, for example, combines a horizontal hinge array (blue) and a vertical hinge array (green) in order to create a mechanism that users operate by pressing down, sliding over, and releasing.

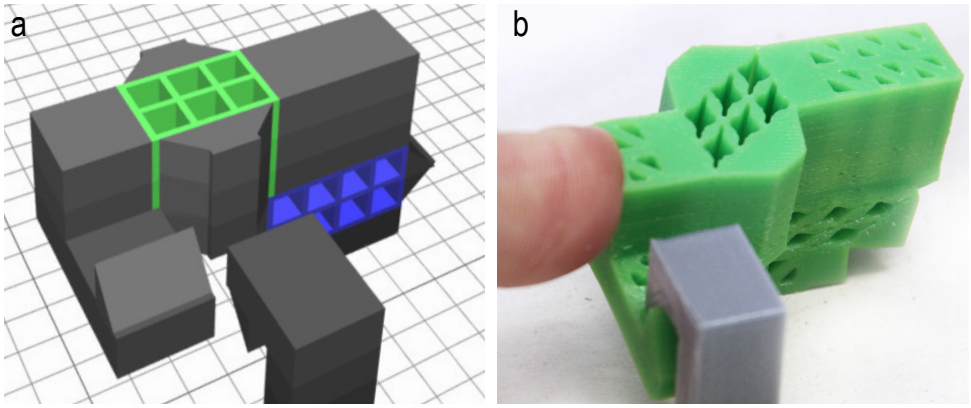

Figure 3.23: This latch requires the ability to shear on two planes, i.e., on the x/z plane denoted in green, and on the on x/y plane indicated in blue.

Our editor color-codes mechanisms automatically according to their orientation in space. This is intended to provide users with a fast overview of the main dimensions of action in their devices and to help recognize hinge arrays from odd viewing angles. In the latch example, green denotes "shearing on the x/z plane" and blue stands for "shearing on the x/y plane". Analogously, red stands for "shearing on the y/z plane".

Note that hinge arrays can overlap. In this case, cells at the intersection bear the combination of all holes. These cells are rendered as the additive mixture of the involved colors, such as yellow, for cells at the intersection between green and red.

3.6.4 Integrating other metamaterial systems

The shear cell is the main element that enables metamaterial mechanisms. However, to allow for the integration with metamaterials by other researchers, the editor can be extended to allow for other cell types.

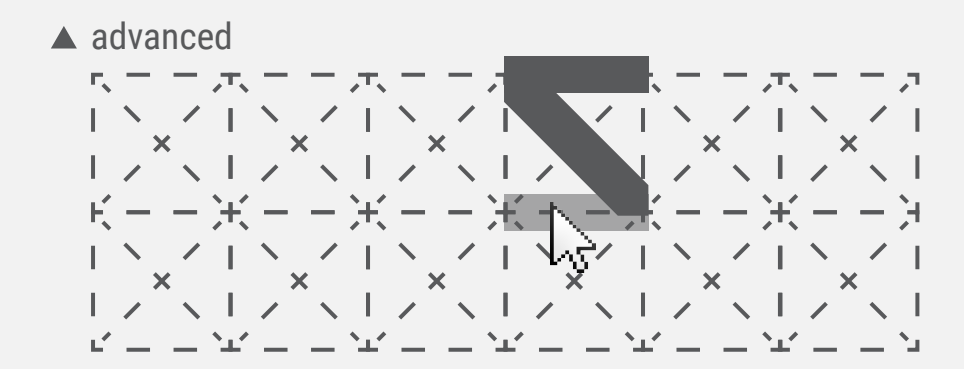

Figure 3.24: Users compose custom cells by adding individual edges in the "advanced" panel.

In order to allow users to explore their own cell types, we offer the advanced panel shown in Figure 3.24. Users compose cells from individual edges by selecting the respective edges. The editor automatically adds all custom cells used in the current model to the cells panel for quick reuse.

Furthermore, users can also create and store groups of cells, for example to create auxetic materials [141], as shown in Figure 3.25. Since metamaterial mechanisms adhere to the standard structure of 3D cell grids that is common for metamaterials, they integrate with earlier research [111, 145].

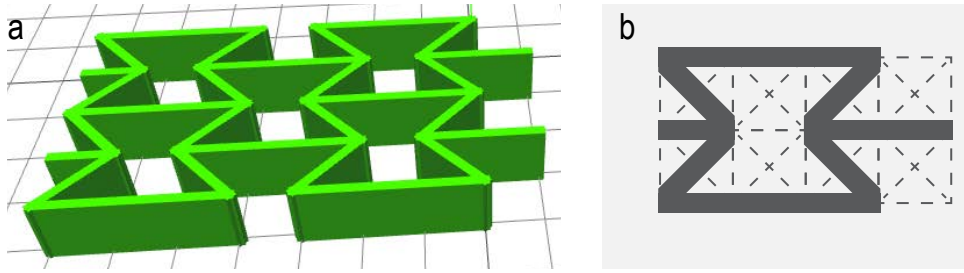

Figure 3.25: Users add groups of cells, here they create an auxetic material from a 4×2 group of cells.

3.6.5 Implementation

In the following, we provide details on the internal processes implemented in our metamaterial mechanisms editor.

Editor (.js)

Our 3D editor is based on WebGL and uses three.js. Internally, the editor creates a dictionary of cells that can be accessed using their position on the grid. Each cell is defined by the 8 vertices making up its bounding box by and the edges that define how the vertices are connected. Note that not all 8 vertices need to be connected by edges.

All vertices lie on our uniform 3-dimensional grid. To generate the 3D cells' structures, i.e., to generate 3D beams from 1D edges, we apply an offset to the vertices' positions on the GPU. Since WebGL does not offer geometry shaders, we use a vertex shader and pass the offset direction and the cell's position with each vertex. The 8 vertices that form a beam are offset uniformly from the two edge vertices on the GPU. To pass additional information about the color and thickness of beams to the shader, we generate a texture where each pixel holds these data for one cell. The color maps directly and the thickness is encoded in the alpha component. In the shader, every vertex looks up its thickness in the texture and calculates the offsets for the new vertices that render a beam from an edge. This enables us to emulate a geometry shader in WebGL and perform all geometry processing on the GPU, which keeps the user experience of our editor smooth.

Simulation (C#)

For simulating the deformation of the user's cell structure, we use the finite elements solver *karamba*[1] [121], which is a plugin-for Grasshopper/Rhinoceros. We implemented a custom C#-Grasshopper-component that receives the mesh data (vertices and edges) and the data for the sim-

[1] http://www.karamba3d.com/

ulation (anchored vertices, force and vertex where the force applies) via a web socket connection. When the simulation is complete, a second custom component receives the transformed mesh vertices and sends them back to the editor. The vertices are kept in the same order within the array as they were received from the editor. We run the simulation on a separate machine to keep the editor running smoothly.

Maintaining the order of the vertices is important to enable geometry processing on the GPU. In the editor, we generate another texture and store the transformed vertices, where XYZ is mapped to RGB. The shader knows the vertex' undeformed position on the grid and looks up the deformed position in the texture.

Depending on the size of the object that is simulated, solving for the deformation can lead to perceivable delays. To compensate for this, our editor interpolates the deformation while the response from the simulation is pending. To do so, we pass the last force where we received the transformation from the server, and the current force that was submitted to the simulation and interpolate the vertex transformation linearly.

Import & export

Users can import mesh geometries directly into our editor. We load .stl meshes using the package 'stl-reader'[2] and voxelize[3] them according to the cell size that the user defined. Once users are satisfied with their metamaterial, they can export their design as a 3D-printable .stl file. We use a constructive solid geometry (CSG) package[4], which we invoke directly from our 3D editor to perform the union operations and generate a triangulated .stl file.

[2] https://www.npmjs.com/package/stl-reader
[3] https://www.npmjs.com/package/voxelize
[4] https://www.npmjs.com/package/openjscad-csg

3.6.6 Software architecture

We designed out software to be loosely coupled. As depicted in Figure 3.26, our editor communicates with the finite element analysis (FEA) tool via a web socket connection. This architecture allow the resource-intensive FEA calculations to be outsourced to a different computer (server) and to exchange the FEA component easily.

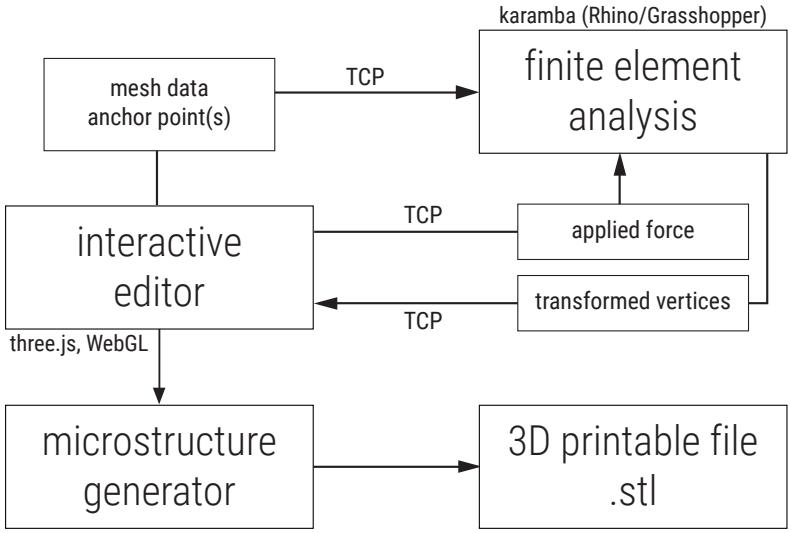

Figure 3.26: The software architecture or our metamaterial mechanism editor.

3.7 DISCUSSION ON MATERIAL & DURABILITY

The ideal material for metamaterial mechanisms is (1) very elastic, i.e., goes back to its original shape without deforming permanently and (2) can withstand high forces without breaking. Spring steel fulfills these criteria well, yet is not possible to fabricate using today's 3D printers. Among commonly available 3D printing materials, flexible materials (such as the TPU *NinjaFlex* and *SemiFlex*) fulfill these criteria well, more so than PLA and certain types of photo-curing resins (such as the *spot-e* resin).

When we created 2D metamaterial mechanisms using our laser cutter, we obtained best results with rubber foam, more so than with ABS sheets and acrylic, which were not compliant enough. Our metamaterial mechanisms worked across fabrication techniques, since the functionality is determined by the cell structure.

In our experience, material fatigue is not a problem, because we operate our metamaterial mechanisms, i.e., bend their living hinges, only to the point where they still fully return to their original shape (the material stays "in the elastic region"). As an example, operating the NinjaFlex-printed Jansen walker shown in Figure 3.27 about ~5000 times (at ~120 rpm) using a 300 W power drill, did not lead to any visible fatigue or damage in the metamaterial structure.

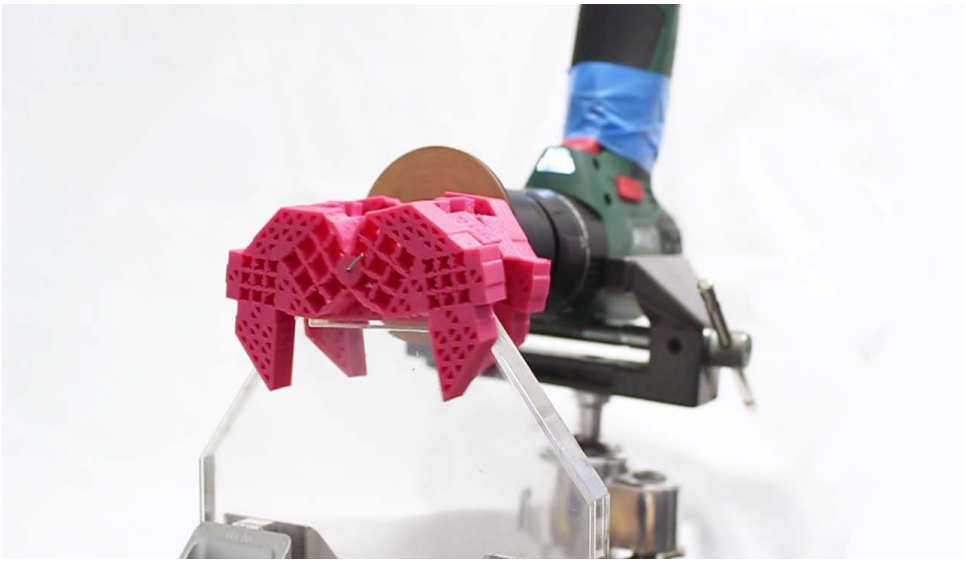

Figure 3.27: We verified the durability of the Jansen walker by operating it ~5000 times using a power drill.

3.8 CONCLUSIONS

In this chapter, we introduced metamaterial mechanisms. While metamaterials so far had been understood as materials, the main contribution of this chapter is that we think of them as *machines.*

On the most basic level, it was the shear cell that allowed us to implement this new perspective on metamaterials. The shear cell allowed us to redirect forces and thus to create basic mechanisms, compound mechanisms, and ultimately simple machines.

While our approach offers tangible benefits for users (e.g., it solves mechanical problems in a single part, thereby eliminates the need for assembly), we see the main promise of this work in that it allows us to achieve a deeper integration between the structural and the mechanical functions of materials.

4

Understanding metamaterial mechanisms

In the previous chapter, we pushed the concept of metamaterials further by going beyond materials and create complete mechanisms from cellular structures. We demonstrated our novel concept with example objects including a door latch and a Jansen walker. However, it remained unclear what types of mechanisms can be implemented with such metamaterials.

In this work, we investigate the underlying working principles of such metamaterial mechanisms. To do so, we analyze the interaction of the two types of cells, identify the underlying topological constraints of metamaterial mechanisms, and ultimately implement this domain knowledge into a computational design tool for non-expert users.

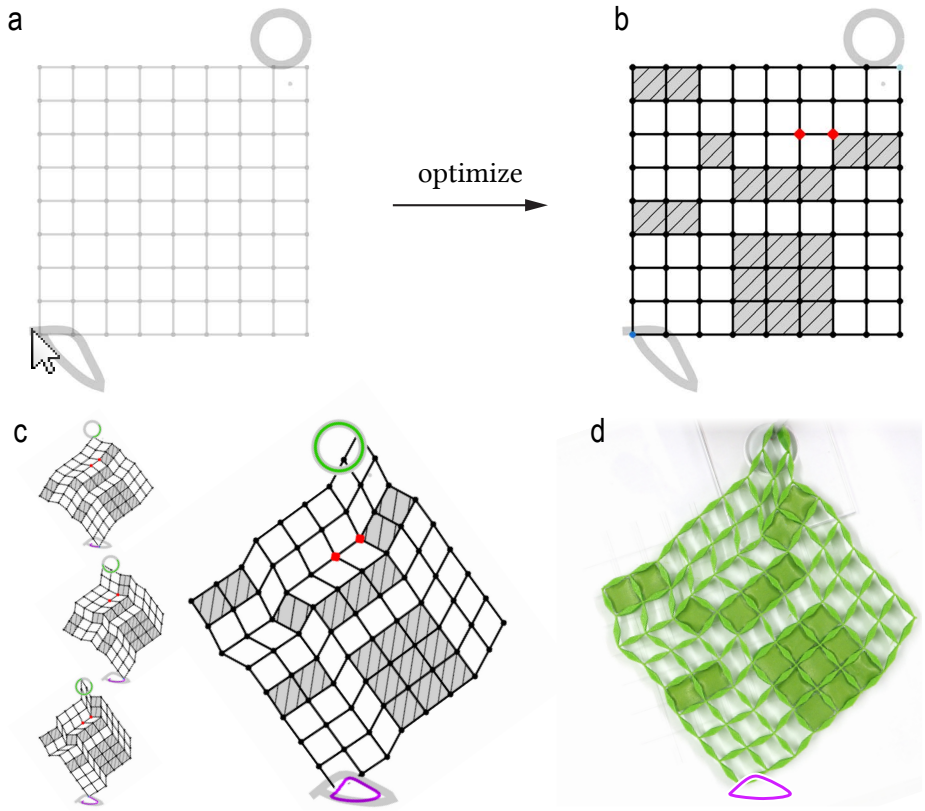

Figure 4.1: We define the underlying working principles of metamaterial mechanisms, which allows us to implement a computational design tool. (a) It takes user-drawn paths and (b) optimizes the cell configuration which implements the transformation. (c) In this example, we show the leg of a walker. (d) The fabricated result matches the optimized motion closely.

4.1 MOTIVATION & CHALLENGES

In this work, we set out to understand the underlying mechanisms that inform the design of metamaterial mechanisms. Such metamaterial mechanisms, as we introduced earlier in Chapter 3, implement a transformation of an input movement to an output movement, such as in Figure 4.2 with the retraction of the bolt when the door handle is pushed down.

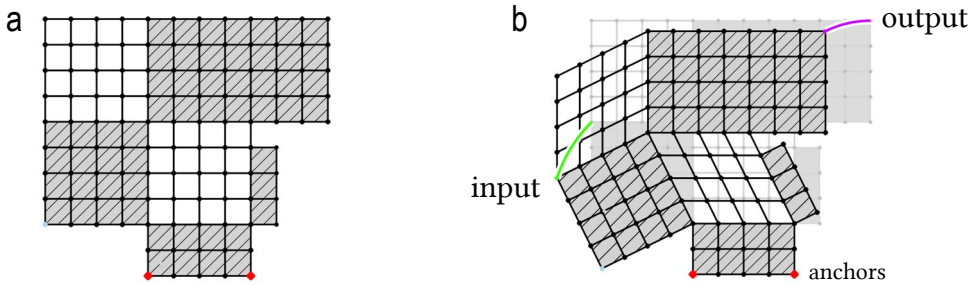

Figure 4.2: (a) Metamaterial mechanisms are implemented by their cell structure to create, e.g., this door latch, from a single block of material. (b) The microstructure implements a transformation from the green input path (pushing the handle) to the pink output path (retracting the bolt).

Metamaterial mechanisms combine two types of cells on a regular grid. The individual cells (see Figure 3), are very simple—they are rigid or can shear. These cells are careful arranged within the material to play together in a well-defined way to implement the mechanism.

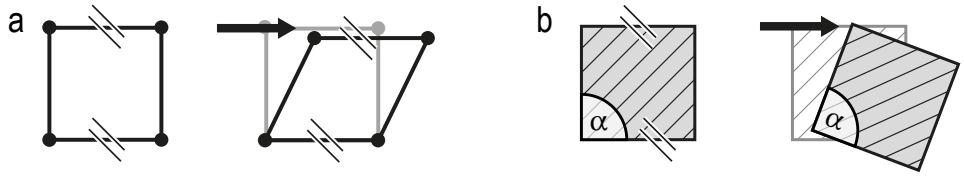

Figure 4.3: Metamaterial mechanisms consist of simple cells: shearing cells and rigid cells. (a) Shearing cells constrain their opposing edges to remain parallel and (b) rigid cells additionally maintain their angle.

In the remainder of this chapter, we will discuss metamaterial mechanisms on a higher level of abstraction, i.e., we will leave out the device applying the metamaterial and instead focus just on the cell structure and the transformation it implements.

4.1.1 Understanding cell constraints and how they interact

To achieve the movement that implements the desired mechanism, the cells need to be arranged on a grid to play together in a well-defined way. More formally, "play together" means that each individual cell has constraints

that it propagates to its neighbors. For example, opposing edges of shear cells remain parallel (parallelism constraints) and rigid cells additionally maintain their angle (angle constraints). Since the cells are connected in two dimensions, their constraints can interact. We illustrate such constraint interactions in Figure 4.4. This example shows how adding a single cell (marked in blue) prevents 7 other shear cells on the grid from shearing.

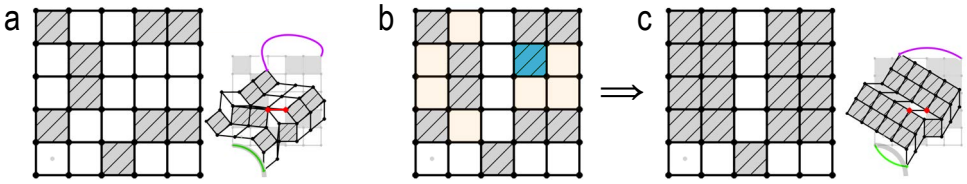

Figure 4.4: (a) In this example, (b) setting one cell rigid (c) prevents 7 cells from shearing, which changes the output drastically. (c) The green input path from (a) cannot be followed anymore.

4.1.2 Large search space

The example depicted in Figure 4.4 illustrates that interactions of constraints are unobvious and that by understanding them, we can reduce the search space drastically. We drill down on this example in Figure 4.5. Here, the naive approach of simply swapping cell types to find a configuration that reaches a user-defined path results in $2^7 = 128$ equivalent mechanisms. This is because any changes within the 7 cells marked in orange (from Figure 4.4) have no effect on the resulting mechanism.

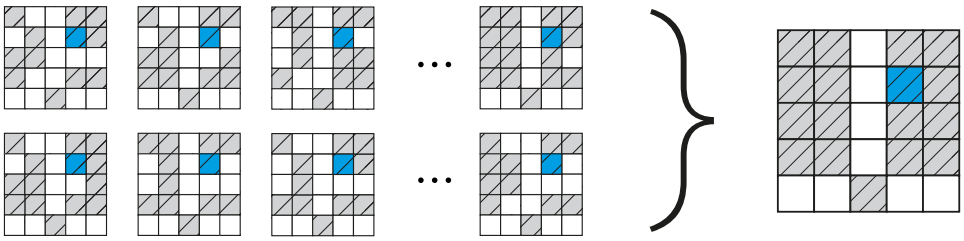

Figure 4.5: In this example, we have 128 configurations that all lead to the same mechanism, because the cell constraints interact.

While this example only concerns one specific scenario, the complete search space would be $2^{25} \approx 3 \cdot 10^7$. We generalize this in Section 4.4 and show how we reduce the search space by several orders of magnitude.

4.1.3 Abstract representation reduces the search space

The key insight of this work is to build an abstract representation of the cell constraints, which defines such distinct mechanisms. We encode constraints that edges impose on each other in a graph whose connected components define their degrees of freedom. We work directly in the reduced space of distinct mechanisms, which are defined by the connected components in the graph, rather than exploring the space of all possible cell configurations.

4.1.4 Computational design tool

Our constraint-based representation of the metamaterial mechanisms and the resulting reduction of the search space makes a computational design tool feasible. We present a computational design tool that optimizes a cell configuration for user-defined paths. Figure 4.1 shows how users define the size of the mechanism they are looking for and draw the input and output paths. Our heuristic optimization searches for a cell configuration that satisfies these boundary conditions.

4.1.5 Novel types of mechanisms

With our computational approach, we not only ease the creation process for users, but we also discover new types of mechanical transformations that were not known before. Metamaterial mechanisms were manually designed to demonstrate useful mechanisms, such as a door latch or pliers. However, we show in Figure 4.6 that the transformations they implemented were basic transformations, such as scaling. In this work, we

demonstrate non-linear transformations, as illustrated in Figure 6b, such as self-intersections, oscillations and smoothing. We believe that the approach will foster more complex metamaterial mechanisms.

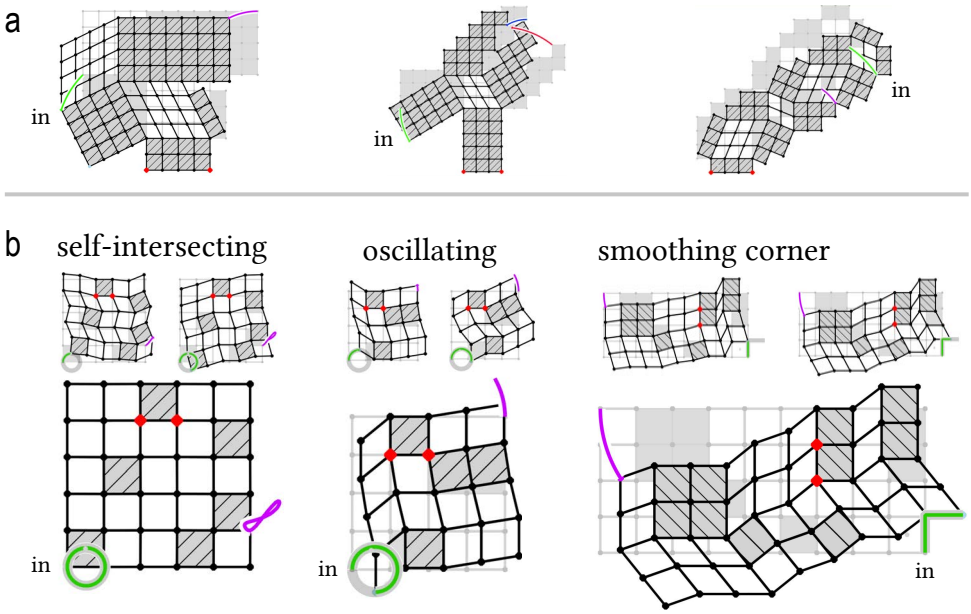

Figure 4.6: (a) The hand-designed mechanisms in only realize simple transformations. (b) In this work, we discover more complex and even non-linear transformations.

4.2 CONTRIBUTIONS & LIMITATIONS

Our main contribution is an understanding of the underlying mechanisms of metamaterial mechanisms. Understanding the constraints that interact within a grid enables a computational design tool that would otherwise not have been possible.

While the naive approach of swapping cells on the grid in order to find a cell configuration that implements a user-defined path transformation is computationally infeasible due to an exponentially growing search space, we contribute an abstract representation of the constraints. Our constraint graph representation reduces the search space significantly.

We show that metamaterials can realize more complex and even non-linear mechanisms—a fact that was unknown before.

However, in this work, we focus on understanding the constraints on the most basic cells, i.e., the square shear and rigid cells. We do not explicitly implement rotated or pre-sheared cells, as suggested in Section 3.4. Since the topology is the same, we show that our constraint graph applied to those cells as well.

4.3 ANALYSIS OF CELL INTERACTIONS

The mechanisms we consider can exhibit intricate behavior even though they are built from very basic building blocks: shearing and rigid cells. Figure 4.4 already demonstrated how drastically a metamaterial can change after performing a small local change, e.g., swapping a shear cell for a rigid cell. These properties are non-obvious but crucial to understand.

In order to understand the movements of a mechanism, we need to simulate its physical behavior numerically. For larger grids the optimization procedure can be time consuming. This can hinder interactive exploration of the space of mechanisms and is especially problematic when sampling and simulating a lot of mechanisms to find one exhibiting some pre-defined behavior.

We describe how we model the constraints of a mechanism, which reduces the number of variables in the physical optimization from all grid points to a few edge vectors. This enables a significantly more efficient implementation and gives insights into the degrees of freedom of a mechanism.

4.3.1 Understanding the constraints

Since our metamaterial consists of shearing and rigid cells, we observe two types of constraints in our cells: (1) parallelism constraints, such that opposing edges always remain parallel and (2) angle constraints, such that

angles of rigid cells remain unchanged. Furthermore, all edges maintain their lengths. This means most edges cannot move independently, e.g. the same vector can represent edges that have to remain parallel. Edges that have to maintain a certain angle can also be represented by a reference edge that needs to be rotated in order to get the second one. To this end we build a constraint graph in which each node represents a cell edge and an arc between nodes the fact that one edge can be constructed by rotating the other (rotations also include the identity transformation).

Figure 4.7 illustrates the graph representation for each cell type individually. For shear cells, opposing cell edges always remain parallel. The constraint graph consequently only contains arcs between opposing cell edges. Figure 4.7b illustrates that rigid cells are represented by a complete graph. This means that transforming one cell edge defines the transformation of all other edges. In other words, if we know how one edge is, e.g., rotated, we know the transformation of the entire cell, because opposing edges remain parallel and adjacent edges maintain their angle.

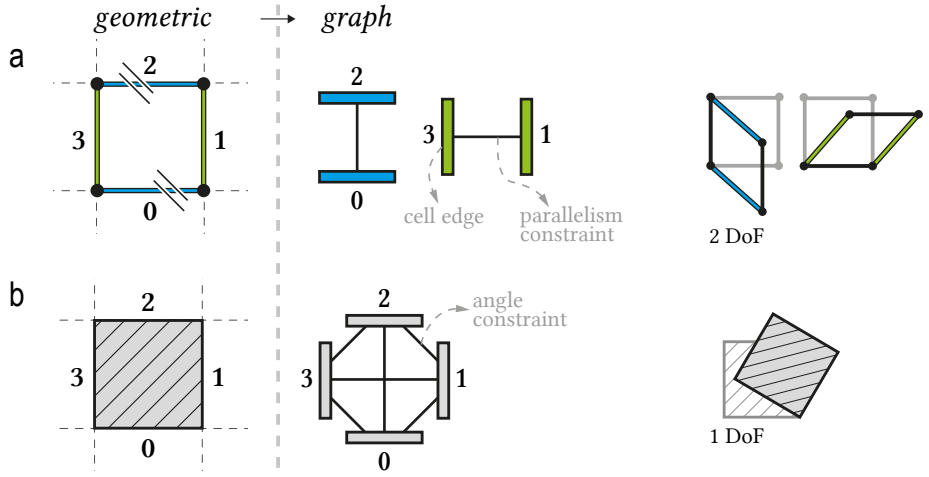

Figure 4.7: We model the constraints as graphs. (a) A shear cell is represented as 2 subgraphs that model the 2 independently moving adjacent edges and the parallelism constraint of opposing edges. (b) A rigid cell is a complete graph, showing that one edge defined the entire cell.

4.3.2 Determining the degrees of freedom

We think of the degrees of freedom (DoF) as a set of edge vectors that can be transformed independently. This property is also illustrated in Figure 4.7, where the constraint graph of the shear cell in (a) consists of 2 connected components, which indicates that the cell has two independently moving parts, i.e., the green and blue edges. The rigid cell in (b) moves as a whole, thus the graph consists of one component. In general, the degrees of freedom of our metamaterial are simply defined by the number of connected components in the constraint graph.

4.3.3 Building the constraint graph

We build the entire constraint graph by connecting the constraints of single cells shown in Figure 4.7 to their neighbor cells, based on coinciding edges. For example, Figure 4.8 shows how to proceed for a metamaterial with 4 cells. We start at the lower left cell and add its vertical edges to the constraint graph. The cell to the right shares the middle vertical edge, thus we link its other vertical edge to the graph, because they need to remain parallel. The two shear cells on the right are processed analogously. The rigid cell, which cannot change its angle, effectively couples edges of the top right and lower left cell. Due to the parallelism constraints, entire rows and columns are linked.

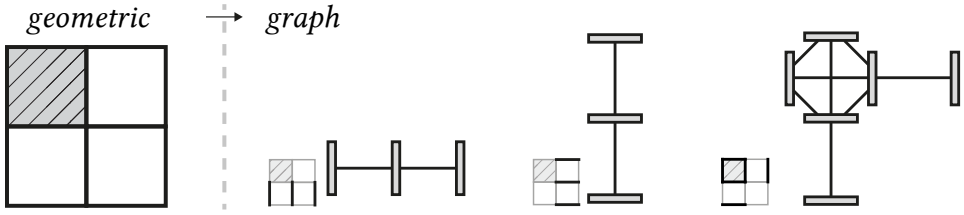

Figure 4.8: We build the constraint graph by connecting the single cell constraints to their neighbor cells based on coinciding edges.

The notion of degree of freedom generalizes to entire mechanisms. The final graph in our example consists of 3 connected components, which tells us that the configuration has 3 DoF. Knowing one edge vector in each com-

ponent uniquely determines all other edge vectors and thereby the whole mechanism. Therefore, we can formulate an optimization problem only involving 3 instead of 12 edges. We will provide details on this idea in Section 4.6.

In a cell grid that consists only of shear cells, the edges in each row and column represent one connected component each. The maximal number of degrees of freedom on an empty $n \times m$ grid is therefore $n+m$. Introducing a rigid cell joins the components of a row and a column into one, reducing the number of degrees of freedom by one.

4.3.4 The influence of anchors

So far, we only considered relative transformations of edges, i.e., edges are transformed with respect to each other. However, since the input of a mechanism is absolute, we need to fixate (i.e., anchor) the cell configuration in order to calculate the vertex positions in absolute space.

Anchoring an edge in a cell configuration reduces its degrees of freedom by one. The same reasoning as for transforming edges, as discussed above, applies; since one edge of a connected component is defined (here, fixed to the ground), all edges in the component are defined. Furthermore, anchors define the scaling of an output path. As shown in Figure 4.9a, the length of the target paths is proportional to the distance between the anchors and the target vertices. Figure 4.9b illustrates, that anchors define a global rotation point.

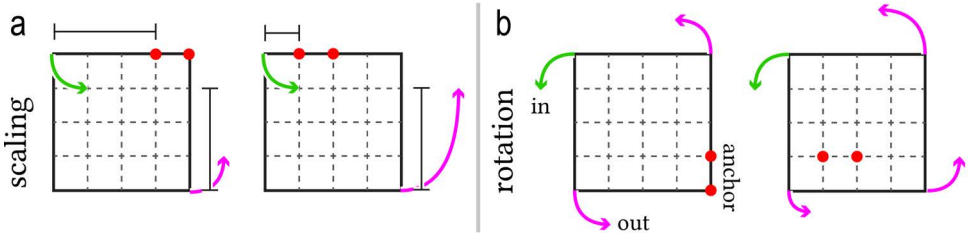

Figure 4.9: The anchor placement influences (a) the scaling of the output path and (b) the global rotation.

4.4 REDUCING THE SEARCH SPACE

Since each cell on a $n \times m$ grid can have 2 states, rigid or shearing, we have 2^{nm} different possible configurations. However, not all configurations will generate a unique mechanism since shearing cells can become rigid because of the constraints. Consider the mechanism in Figure 4.10. The constraint graph reveals that the green and blue cells cannot shear. This is the case when *all edges* of a cell are contained in the *same connected component*. Since the rotation of both potentially independent edges of the shear cell are defined by the same connected component, their angle is constraint to the original 90° and can thus not shear.

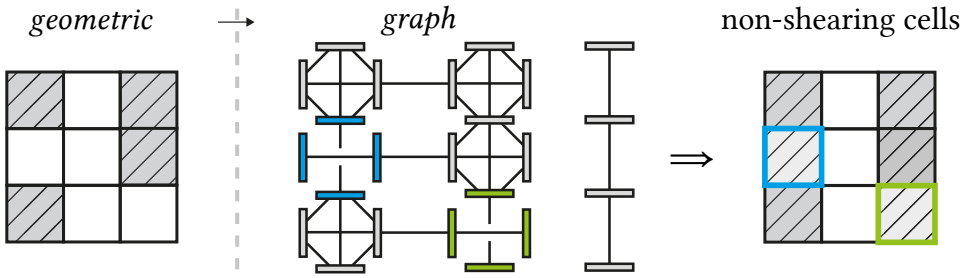

Figure 4.10: It might be non-obvious how many DoF this metamaterial has. Our constraint graph reveals that it contains 2 cells that cannot shear, i.e., the blue and the green one, leaving only 2 DoF.

Since the blue and green cells cannot shear, the two cell configurations in Figure 4.10 are equivalent. We only want to consider *unique mechanisms* in our optimization. But how many of these unique mechanisms exist? To answer this question, we enumerate all possible connected component configurations. It is indeed possible to give an explicit formula for the number of all unique mechanisms on a $n \times m$ grid. We present the derivation of this function in Appendix 4.10. Here, we show the empirical verification the formula for all configurations on grids with $n < 5$. In Figure 4.11, we show the number of all 2^{nn} mechanisms (gray) along with the number of unique mechanisms (blue). While this number still grows rapidly with increasing grid size it reduces the space of configurations on a grid from

growing quadratically with respect to n in the log-plot to linearly, which is already significantly better. This enables us to consider much larger grids when searching for mechanisms with certain properties.

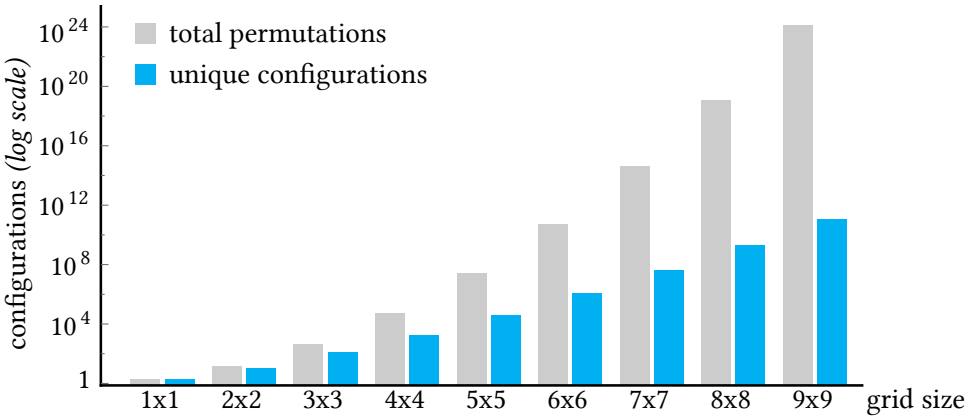

Figure 4.11: The number of all configurations (log scale) on a square grid compared to the number of all unique mechanisms.

4.5 SIMULATING THE MOTION

To simulate the deformation of the object when a vertex of a cell moves, we use a simple elastic material model. We assume the edges are rigid and deformation occurs when two edges incident to a corner are rotated relative to each other.

Through the constraint graph all edges are defined as soon as one representative edge per connected component is known. To define the remaining edges, we traverse the constraint graph, e.g. using breadth-first search, starting at the representative edge. When moving across an arc the appropriate rotation is applied. When all edges are known we can start to reconstruct cell vertices. To this end we traverse the grid itself starting at an anchored vertex. Each vertex is effectively reconstructed by summing all edge vectors along a path connecting the vertex to the anchor.

Direct numerical simulation would try to find a set of grid vertices minimizing this deformation energy subject to a set of non-linear constraints, edges have to maintain their length, opposing edges in a cell have to stay parallel and angles in rigid cells stay at 90°. Additionally, certain vertices are fixed, either in their original position or by an external force driving the mechanism.

Analyzing the constraint graph allows us to formulate a constrained optimization problem that has the same solution but is much more efficient than the direct approach.

The shape of the whole mechanism is determined by fixing the degrees of freedom, which requires two numbers per connected component. The current state of the deformation can therefore be modeled by a state vector $x \in \mathbb{R}^{2\,N_d}$ where N_d is the number of degrees of freedom. Each state vector uniquely determines all edge vectors and is associated with the deformation energy

$$D(x) = \sum_{i=0}^{N-1} \left(\alpha_i(x) - \frac{\pi}{2}\right)^2,$$

where $\alpha_i(x)$ stand for an interior angle of the i-th cell and N for the number of cells. This energy models the fact that each cell will, depending on the material used, resist shearing, even for non-rigid cells. Note, that we only need to take one angle per cell into account because all angles in a cell produce the same energy.

To simulate the deformation, we find a minimizer of D with respect to the following constraints:

C1. The edges are rigid and therefore have to be fixed in length. Constraining all degrees of freedom to unit length will ensure this property for all edges in the mechanism.

C2. Cells cannot invert, i.e., change their orientation. We therefore assure that the cell areas always remain positive.

C3. Anchors are fixed in position.

C4. The vertex that is being dragged is fixed in position.

Note that we do not have to enforce that edges are parallel in shear cells or that rigid cells maintain their interior angles. These constraints are already built into the reduced representation and each state vector induces a valid cell layout. This constitutes another advantage over the direct approach where these constraints have to be taken into account explicitly.

Unfortunately, the energy and the constraints are non-linear and non-convex. However, the constraints are only quadratic and they can be analytically differentiated. The energy D can also be analytically differentiated albeit yielding more complex terms due to the calculation of angles from edge vectors. We solve the problem of minimizing D subject to C1-C4 by employing the non-linear interior point solver IPOPT [172], which gives us a configuration for a specific vertex position moved along a path, which we call handle. To simulate the full deformation, we sample the desired motion path and find the correct configuration by solving the optimization problem starting from the previous configuration as an initial guess.

4.5.1 Invalid input

The range of the handle is limited by the structure of the mechanism, which is fixed by anchors. If a handle position outside this range is prescribed the optimization problem has no solution because C4 cannot be satisfied. To yield an approximate result even for these situations we try to find a handle position, which is close to the desired one but in the range of the handle.

To this end we solve another optimization problem minimizing the distance of the updated handle position $\widetilde{P}_h$ to the prescribed one P_h:

$$D\prime\,(x) = \left\| \widetilde{P}_h - P_h \right\|^2,$$

subject to the constraints C1-C3. The resulting configuration is one that is valid and places the handle close to the desired position. However, the solution represents only one valid configuration, which will not minimize D in general. We therefore optimize D with respect to C1-C4 using the new handle constraint.

Evaluation

As a preliminary evaluation of the simulation, we manually created 3 mechanisms with different inputs in our software and manufactured them from laser cut acrylic and 3D printed hinges (Ninjaflex filament). We tracked the motion of the physical mechanisms using OptiTrack with up to 7 markers on each mechanism and compared it with motion paths from our simulation. We adjusted the OptiTrack data to account for scaling. Figure 4.12 shows at one example that the recorded and simulated data matched closely. These experiments support that the results of our simulation match the behavior of physical mechanisms.

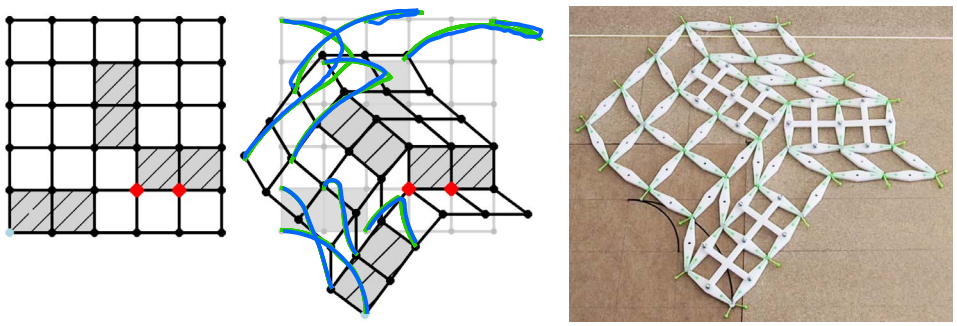

Figure 4.12: Example of our evaluation. The green path represents the paths from our simulation, blue is the data from OptiTrack.

4.6 OPTIMIZATION OF CELL PATTERNS

We aim to alleviate the problem of finding a suitable configuration of cells given a set of desired input and output paths, as well as anchored vertices. In the following, we detail our algorithm that exploits aforementioned properties of metamaterials mechanisms such as the relation between cell transformation and anchors, and the constraint graph. Note that while most of our examples illustrate rectangular cell configurations, the approach is equally applicable for arbitrarily shaped objects. The main requirement for our approach is a simulation that correctly reproduces the deformation of a mechanism given an input path, like the one described in the previous section.

4.6.1 Overview

Our algorithm takes a set of user-defined input and output paths as well as a rough shape of the desired device as input. Our algorithm aims to find a close-to-optimal fit between the motion of the mechanism and the user-defined paths. Generally, input paths are actuated (e.g., by users) and the mechanism transforms this motion to the output path. In context of the optimization, however, we do not need to differentiate between input and output paths but use them as a list of paths which a mechanism should be able to reproduce. As a first step in our algorithm, we automatically determine the positions of the anchors for the cell configuration based on the scale and direction of the paths. We then generate a mechanism that produces the desired motion. An overview of the algorithm is illustrated in Figure 4.13.

Since the behavior of a mechanism is non-linear, we create cell configurations using stochastic optimization, specifically Simulated Annealing [64, 128]. Instead of modifying the cell configuration directly, we modify the constraint graph, i.e., split and merge connected components until the algorithm converges. To speed up computation, we resort to a hierarchical

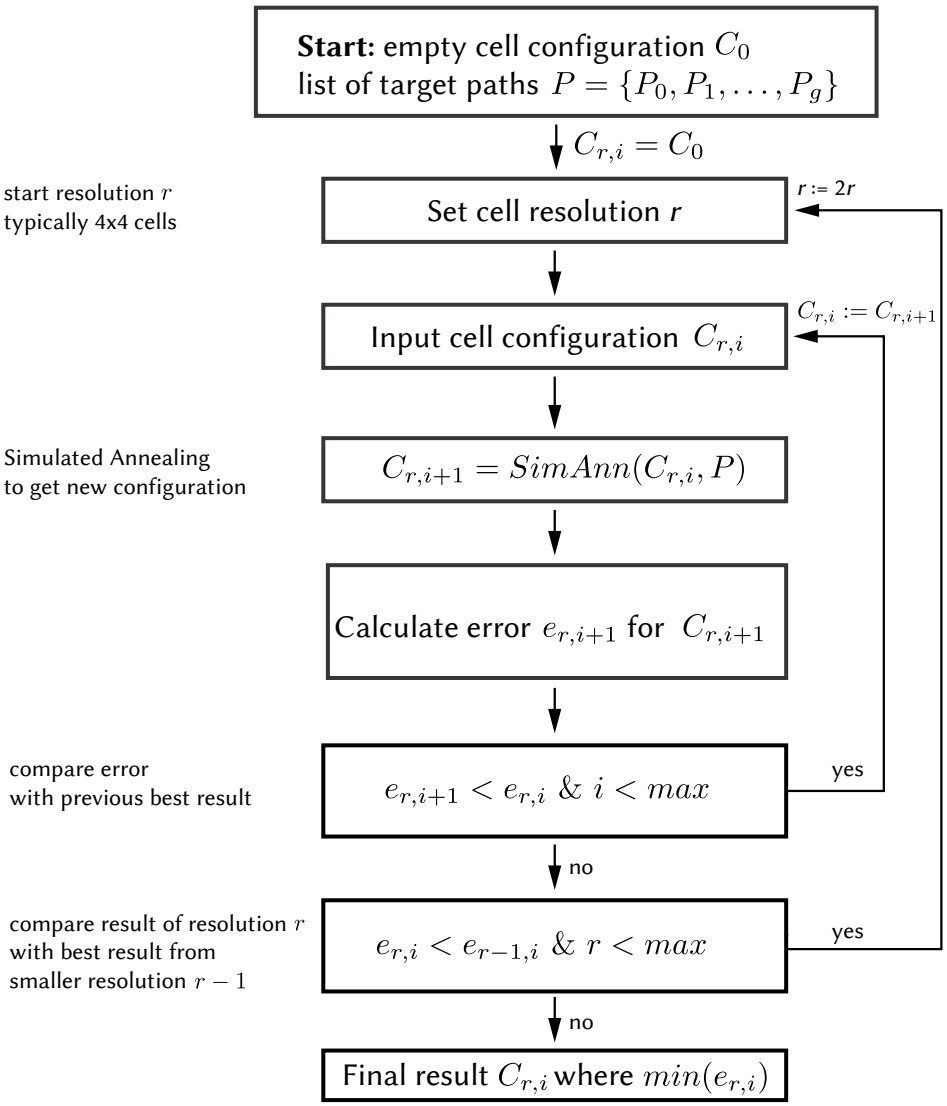

Figure 4.13: Overview of the process of automatically finding cell patterns given user-specified paths.

approach where we first find an optimum configuration for scaled-down versions of the mechanism and use this configuration as seed for larger versions. The output of the algorithm is a cell configuration that produces the desired paths.

4.6.2 Input and representation

Users specify the shape of the cell configuration, denoted as $C_0 \in \{0,1\}^{x,y}$, with x and y being the dimensions of the mechanism. This is a configuration of cells with undefined behavior. In the later optimization, the type of each cell is specified (e.g., they become shear or rigid cells), which ultimately governs the motion of the mechanism. Each mechanism is defined by its cells, which consist of vertices V and edges E. The position of the anchors $A \subseteq V$ governs the motion of a mechanism and fixes the mechanisms absolute position in space. Anchors are automatically determined by our algorithm.

Besides C_0, users specify a set of desired paths $P = \{P_0,\ P_1, \ldots,\ P_m\}$. Each path is a list of points, stored as matrix $P_i \in \mathbb{R}^{2\times n}$. The first point $p_{i,0}$ of a path P_i is a vertex of the cell configuration, i.e., $p_{i,0} \in V$. Note that while we use one input and one output path for clarity, the optimization generalizes to more than one input and output path.

4.6.3 Setting anchors

Our algorithm automatically determines candidates for positions of the anchors based on user-defined paths. This is done by anchoring a single edge e (i.e., setting its two vertices to be anchors). There are three main considerations that govern the positioning of the anchors (i.e., finding which edge to anchor). First, the distances between the paths to the anchors should each be proportional to the distances between the path lengths. For example, if an input path is half the length of an output path (i.e., scale ratio 1:2), then the distance between the input path and the anchor should be half the distance of the output path to the anchors, as illustrated earlier by Figure 4.9. This accounts for scaling between the paths and can be calculated as

$$\min_{l} \sum_{i=0}^{m} \sum_{j=i+1}^{m} \frac{|P_i|}{|P_j|} - \frac{|e_l - P_i|}{|e_l - P_j|}$$

where m denotes the number of user-defined paths. Secondly, anchors are essentially rotation points between paths. We therefore choose the location of anchors to reflect this rotation. Thirdly, to allow for maximum motion range of a path, the anchors should be as far ways from all path points as possible. This is formulated as

$$\max_{l} \sum_{i=0}^{s} |e_l - P_i| \,.$$

Since this operation can be performed on a scale-down version of the configuration, we can exhaustively search the space and choose the best anchor positions. It is possible that no valid positions are found, e.g., if the length of a path exceeds to overall diagonal length of the cell configuration. If this is the case, users can adjust the paths, e.g., decrease the overall length of the paths.

4.6.4 Generation of cell configurations

Given a list of user-defined paths, any cell configuration C will transform the paths in a specific way. For a single path P_i, this transformation is denoted as $C\left(P_i\right)$. We aim at finding a cell configuration C_j with the minimal difference between P_i and $C\left(P_i\right)$ while using as few as possible degrees of freedom to increase mechanical stability. We thus aim to find a solution (i.e., a close-to-optimal cell configuration) given the objective

$$\min_{j} \sum_{i=0}^{m} \omega_i \left|P_i - C_j\left(P_i\right)\right| \,.$$

$\omega_i \in [0,1]\,, \quad \sum_{i=0}^{m} \omega_i := 1$ are user-defined weighting factor for the individual paths. In our implementation, we typically use equal weights for input and output path (i.e., 0.5 for 2 paths).

A trivial approach to the problem would be to randomly sample cell configurations (i.e., switching cell types randomly) and choose the best fit between the user-defined paths and the paths the mechanism produces. This, however, is not feasible, as discussed in Section 4.4, given the large number of possible cell configurations that yield similar movements or are rigid.

In our approach, instead of modifying the cell configuration directly, we manipulate the underlying constraint graph with respect to the degrees of freedom of a configuration. We typically aim for a configuration with as little degrees of freedom as possible that still can reproduce a path well. A too low number of degrees of freedom would not allow for complex motion (e.g., with changes in direction). A too high number of degrees of freedom would lead to mechanically unstable structures. Therefore, we constrain a configuration to be within DoF_{min} and DoF_{max}, which are typically chosen to be 2 and 5, respectively.

For each step in our optimization, we compute the current degrees of freedom DoF_j for a configuration C_j. The degrees of freedom govern the next step, i.e., if the next configuration shall contain more or less degrees of freedom. If the degrees of freedom should be increased, we split a chosen connected component. Conversely, we merge two connected components to decrease the degrees of freedom. If the degrees of freedom are within the limits, an operation is chosen randomly. The connected components that are affected by the operation are chosen randomly.

We use Simulated Annealing for the optimization to avoid converging to a local minimum. We calculate the current temperature T_j for each step as

$$T_j = (1 + e^{\frac{e_k}{T_0 \alpha^k}})^{-1},$$

where e_k denotes the error of the current iteration k. It is calculated with respect the overall objects as the sum of differences between as user-defined paths, i.e. $e_k = \sum_{i=0}^{m} \omega_i \left| P_i - P_i' \right|$. The error is normalized with respect to

the length of the paths. T_0 is the starting temperature, which we typically choose as one third of the number of maximum iterations, as described below. α control the overall falloff of the algorithm, typically $0.85 < \alpha < 0.99$ (in our case, we chose α as 0.95). To avoid that the algorithm converges in a local minimum, we restart the Simulated Annealing process several times. After each run, we compare the best results of the current and previous run. Convergence is reached if the result did not improve compared to the previous run.

Hierarchical generation

We chose a hierarchical approach for the optimization, to avoid users having to estimate the cell resolution of their configuration. We start the Simulated Annealing process for an initial configuration C_0 with a small resolution r, e.g., 4×4 cells. Once the previously described Simulated Annealing procedure converged, we double the resolution and restart the Simulated Annealing with the larger resolution $r + 1$. After convergence, we compare the error of the cell configuration C_r and C_{r+1}. In case the result did not improve, we assume C_r to be the best solution. If the result did improve, we increase the resolution again and restart the process. A typical error function for the whole process is shown in Figure 4.14. Since the process typically converged after 100 to 150 iterations, we set the number of maximum iterations to 200. Similarly, the algorithm typically converges after increasing the resolution 2 to 3 times. We therefore set the maximum number of increasing the resolution to 5.

4.6.5 Evaluation

We randomized a small number of ground truth examples to evaluate if the optimization finds an optimal solution given a user-defined input and output path. We randomly generated 10 cell configurations with 3 or 4 degrees of freedom. We manually set the input and output vertex, an input path and the anchors. Inputting this into the simulation yielded an

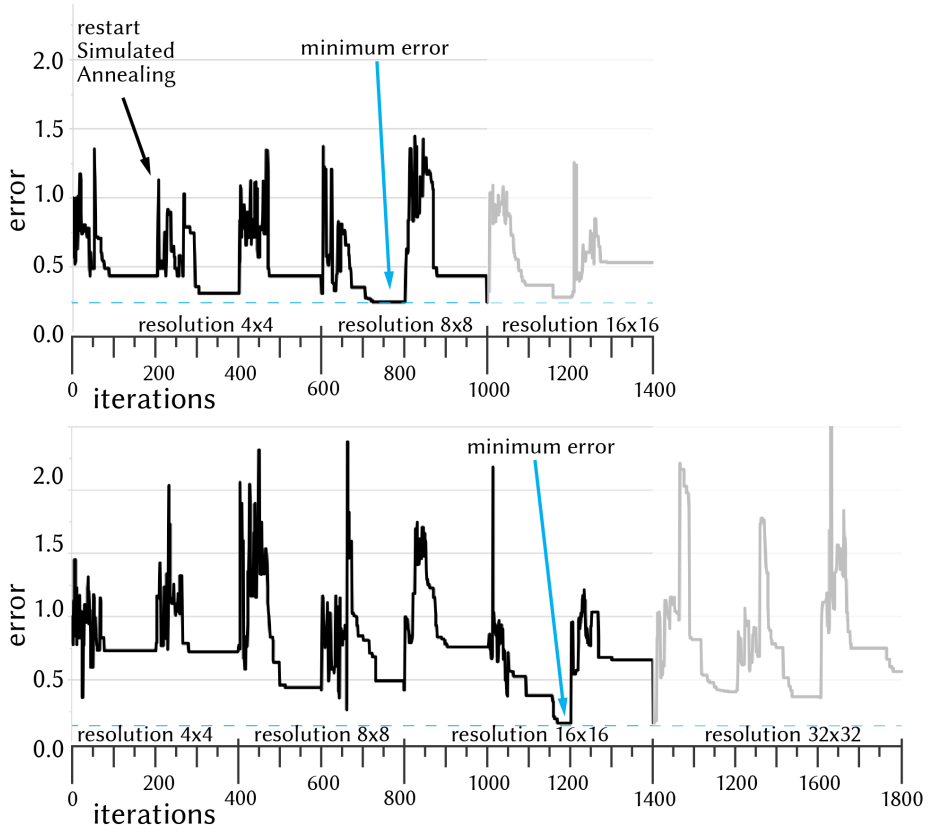

Figure 4.14: Typical error function for two examples. For top, the process converged after a total of 1400 iterations, with the minimum error for a resolution of 8 $\times$ 8 cells. Results with higher resolution (16 $\times$ 16) were discarded due to increased error. The bottom example converged with a resolution of 16 $\times$ 16 cells. For each resolution, Simulated Annealing is restarted multiple times.

output path. The input path and output path, as well as an empty low-resolution cell configuration with correct aspect ratio were given as input for the optimization. The resulting cell configurations, although different in terms of cell configuration and anchoring, could all truthfully reproduce the target movement. Two examples with target and solution are shown in Figure 4.15.

Note that the cell configuration, the anchors and the target resolution were not known to the optimization but were generated.

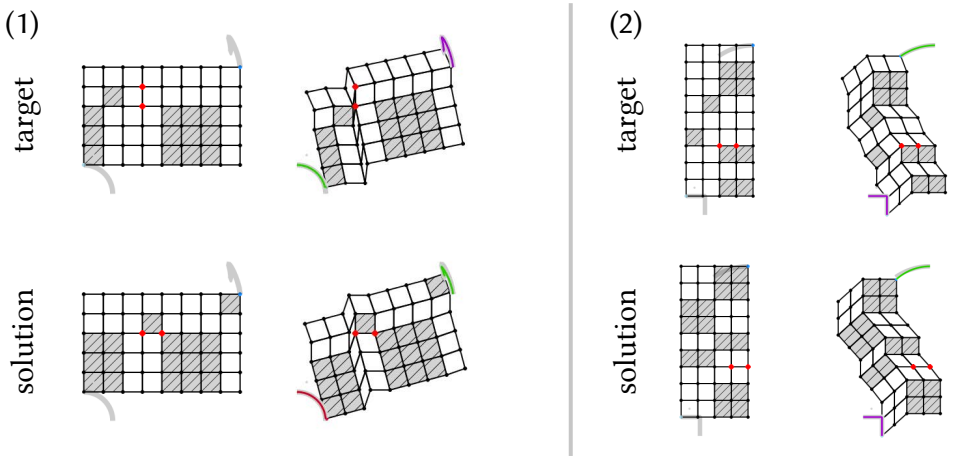

Figure 4.15: Two of ten ground truth examples (top) we used to test the optimization. The results in the bottom were generated without inputting the cell configuration or position of the anchors into the optimization procedure.

4.6.6 Implementation

While we already discussed the optimization and simulation in previous sections, we want to briefly mention the frameworks we used. We integrated the metamaterial mechanism optimization, as described above, into a simple editor. The editor allows users to draw the shape of their desired mechanism and to set input and output paths, which the software will optimize for. Users can use pre-defined paths for precise motion constraints or simply draw rough paths. Time needed to generate a cell configuration depends on whether the solution requires a high-resolution configuration. If the solution is found in a low-resolution grid, the algorithm takes approximately 1 minute to converge on a commodity notebook (MacBook Pro 2015 with Bootcamp). For higher resolutions (e.g., 32×32 cells), finding the best solution takes up to 10 minutes. We are confident that this time can be decreased by parallelizing parts of the algorithm, e.g., running multiple threads of Simulated Annealing at once. The editor is implemented in C# and uses the .NET framework 4.5. We use the Windows Presentation Foundation (WPF) as our GUI toolkit. The optimization procedure is also

implemented in C#, which calls a wrapper to our C++ simulation tool. The simulation is written in C++, because it uses IPOPT [172], as we discussed in Section 4.5.

4.6.7 Limitations of the design tool

The editing capabilities of our editor are currently limited to 2D. Furthermore, we implemented only square cells so far. Triangular, rotated, or pre-sheared cells, as suggested in Section 3.4 are not integrated in the current version of the editor. However, this would be a simple extension. We also currently don't offer mesh export options, as we focused on the optimization. A conceivable option would be to implement the tool as a plugin to existing CAD tools, e.g., Autodesk Fusion 360, as they offer elaborate modeling options for the parts that embed the metamaterial.

4.7 EXAMPLES

The main contribution of this work is the abstract representation of metamaterial mechanisms, which ultimately allows an automatic generation of such metamaterials. While the device that embeds the metamaterial is not the focus of this work, we want to give some brief examples of how our generated metamaterials might be embedded in a device-context. This work allows to create new types of mechanisms, which experts of mechanism design can add to their repertoire. The examples here are merely intended to foster future discussion and research of monolithic cell-based mechanisms. More transformations are listed as examples in Appendix 4.11.

Kinetic sculptures. One example, where metamaterial mechanisms might be embedded into, are kinetic sculptures, toys, or walking automata [161]. As shown in Figure 4.16, our design tool enables users to create custom walk-cycles from metamaterials.

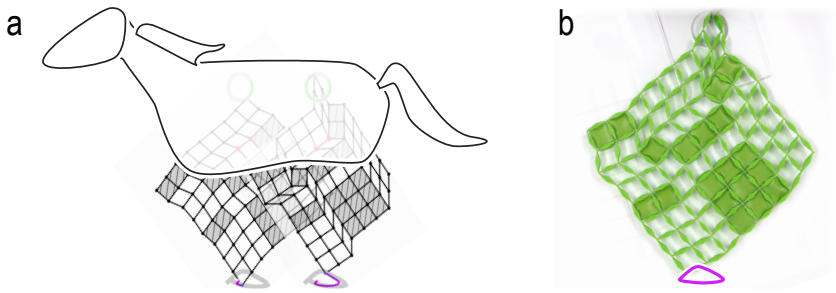

Figure 4.16: Embedding metamaterial mechanisms into kinetic sculptures.

Custom mechanisms. Users might also want to create grippers with a custom motion path. In Figure 4.17, we illustrate embedding metamaterial mechanisms with different motion paths as grippers for collecting items, e.g., for picking-challenge robots. Other applications for such custom paths that come to mind are, e.g., fans with a path cycle, or deflectors for sprinkler which can be customized to sprinkle a specific area optimally.

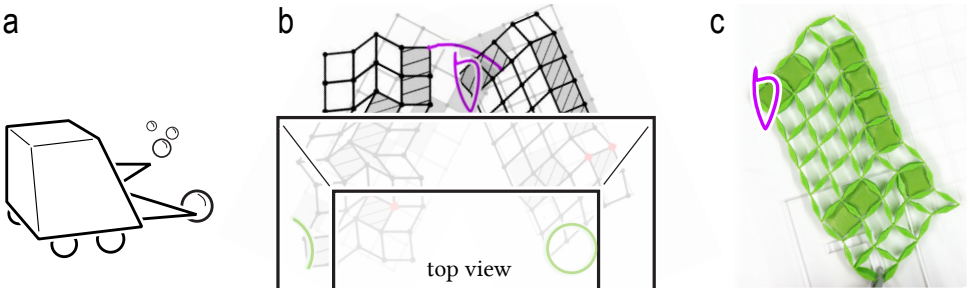

Figure 4.17: Example of metamaterial mechanisms as gripper with custom motions for, e.g., robots.

Clock. Another simple example is an alarm clock, as depicted in Figure 4.18. The metamaterial transforms a rotary input into an oscillating motion to strike the bells.

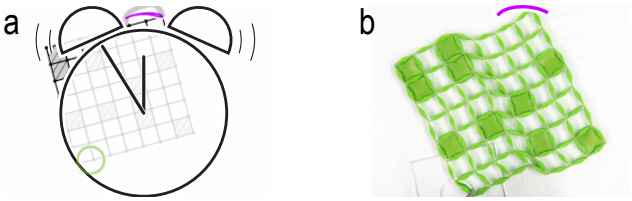

Figure 4.18: A simple example is embedding an oscillating metamaterial into an alarm clock.

4.8 DISCUSSION & OUTLOOK

Our work aims at providing an understanding of metamaterial mechanisms and go beyond purely exploratory work. Our work opens way for researchers to explore, build and use such mechanisms. Moving away from a representation that is based on cells to a higher level—the constraint graph—allowed us to significantly reduce the space of cell configurations, making computational approaches feasible. Note that while the search space is highly reduced, it is still not feasible to fully enumerate the space. As discussed above, a cell configuration of 9×9 cells yields 10^{12} unique transformations (but 10^{24} cell configurations). We are confident, however, that our work informs the research in the space and can provide a foundation to make metamaterial mechanisms accessible to a broader audience.

4.8.1 Limitations and considerations

The constraints graph and the optimization allow us to generate insights into the inner workings of metamaterial mechanisms and create a computational tool that allows for their automatic generation. There are, however, limitations to our approach.

Transformation symmetry

For a given cell configuration, moving an input vertex along the input path yields a transformation of the output vertex, resulting in an output path. If, however, the output vertex is moved along the same output path, the transformation of the input vertex is not equal to the input path. This

behavior has to be taken into account when designing mechanisms that should be actuated from by moving the input and the output vertex. It can, however, also be exploited to create more interesting and complex mechanisms.

Transformation complexity

In our experiments, we observed that transformation between an input and an output path are limited in their complexity. Specifically, we saw that the number of inflection points between the two paths is roughly the same. An input path with one inflection points, for example, usually yields output paths with zero to two inflection points, but not arbitrary number.

Error cases of the optimization

While the optimization generally yields good results, there are cases where no solution (i.e., no good cell configuration) can be found. Examples include a mismatch in complexity between input and output path (e.g., line as input and an ampersand curve as output), or when input and output are mirrored. We will alleviate this problem by warning users of impossible configuration before the optimization.

Extending to 3D mechanisms

The constraint graph is based on the fact that a cell that is constrained in length is fully defined by a single angle. This does not hold true for 3D cell configurations. Therefore, there is no trivial extension of our approach into the 3rd dimension. We already started investigating this interesting future work, including replacing the 2D angle constraint with using the angles of a basis in each cell as constraints. A preview of a deformed 3D mechanism with 2 rigid cells is shown in Figure 4.19.

4.8.2 Practical extensions

There are several practical extensions to our work, that concern the editor, including the simulation and generation of mechanisms.

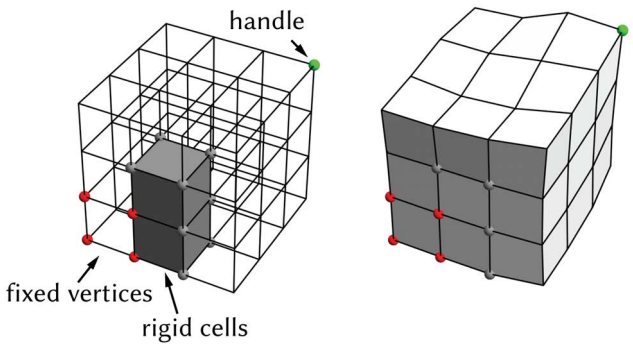

Figure 4.19: Preview of a 3D deformed mechanism with 2 rigid cells.

Adapt to material properties

We used an idealized simulation that does not incorporate engineering factors such as material properties or friction. We did so to focus on the interaction between geometric constraints within a mechanism, which could be missed when taking mechanical factors such as transmission loss into account. While the mechanical properties can be optimized (e.g., through high-resolution 3D printing), the underlying constraints are inherent to the geometry. Material properties could be used to expand the repertoire of transformations. Including other types of simulation such as finite element analysis would be one beneficial extension of the editor. This would also allow us to adaptively change the stiffness, thus actuation force, of a mechanism. This can be achieved by merging cells.

Multiple layers with the same input

Many practical applications such as a multi-legged Jansen walker or a robotic gripper require that multiple layers of mechanisms move in concert with each other, i.e., they use same coinciding input. While our current editor only supports a creating a single layer at a time, they could be merged using any CAD tool.

Extending to different cell types

Our constraint graph, and consequently our optimization algorithm, hold true for other cell suggested previously in Section 3.4, such as rotated and pre-sheared cells. Our editor only needs to be extended slightly to edit such cells.

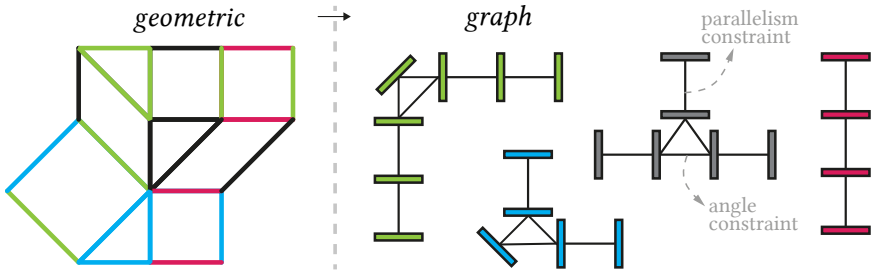

Figure 4.20: Our constraint graph generalizes to the cells we introduces in Section 3.4.

Considering temporal changes

We were concerned with the spatial transformation of metamaterial mechanisms. An extension of the optimization would be to also include a temporal component. Considering the speed of transformation would allow features such as keyframing and easing of motion. We plan to investigate this interesting aspect by adapting our error function and add keyframing to the editor.

4.9 CONCLUSIONS

We analyzed metamaterial mechanisms with respect to their topological constraints. Although the basic cells in this work (shear and rigid cells) are simple, connecting them creates complex interactions. We investigated these interactions, which we modeled as a constraint graph. This abstract graph representation allows us to explore metamaterials on a more abstract level and avoid having to rely on the raw cell structure which ex-

hibits a prohibitively large search space. Consequently, we implemented our knowledge as a computational design tool for the automatic generation of such mechanisms.

On a higher level, we think of our work as the first step towards what we like to call *heterogenous mechanical metamaterials*. We define them as metamaterials that consist of *different types* of cells. Most often metamaterials consist of cells that are topologically equivalent; all cells have the same function, but they can vary in parameter. Metamaterial mechanisms is one example of a material that consists of *topologically different cells*. They exhibit interesting behavior yet the interactions between cells are hard to understand. In the future, we will go step-wise towards investigating more of these heterogenous metamaterials by creating generic tools that allow researchers to investigate combinations of different types of cells from related work.

4.10 APPENDIX A: ENUMERATING ALL UNIQUE MECHANISMS

As noted in Section 4.4, all edges in a row or column necessarily belong to a single connected component. Starting with an empty $n \times m$ grid we have $n + m$ connected components. Introducing a rigid cell joins two connected components and removes one degree of freedom. Suppose we want to enumerate all unique mechanisms with k degrees of freedom. This amounts to k connected components where we differentiate between x empty columns, y empty rows and z components formed by merging rows and columns using rigid cells. There are $\binom{n}{x}$ ways to choose the columns and $\binom{m}{y}$ ways to choose the rows. The remaining $m - y$ rows and $n - x$ columns can be arbitrarily partitioned into z sets. The Stirling number of the second kind counts the number of these partitions as $S(n - x,\ z)$ and $S(m - y, z)$ where we use the convention $S\left(a, b\right) = 0$ for $a < b$. The z sets of rows and columns can be connected in $z!$ ways. This gives

$$G^{n,m}\left(x, y, z\right) = \binom{n}{x}\binom{m}{y} S\left(n - x,\ z\right)\ S\left(m - y, z\right)\ z!$$

different possibilities. Summing over all values for $x,\ y$ and z we obtain the number of all possible unique mechanisms on a $n \times m$ grid. We empirically verified this number by analyzing all possible configurations on grids with $n < 5$.

4.11 APPENDIX B: EXAMPLE TRANSFORMATIONS

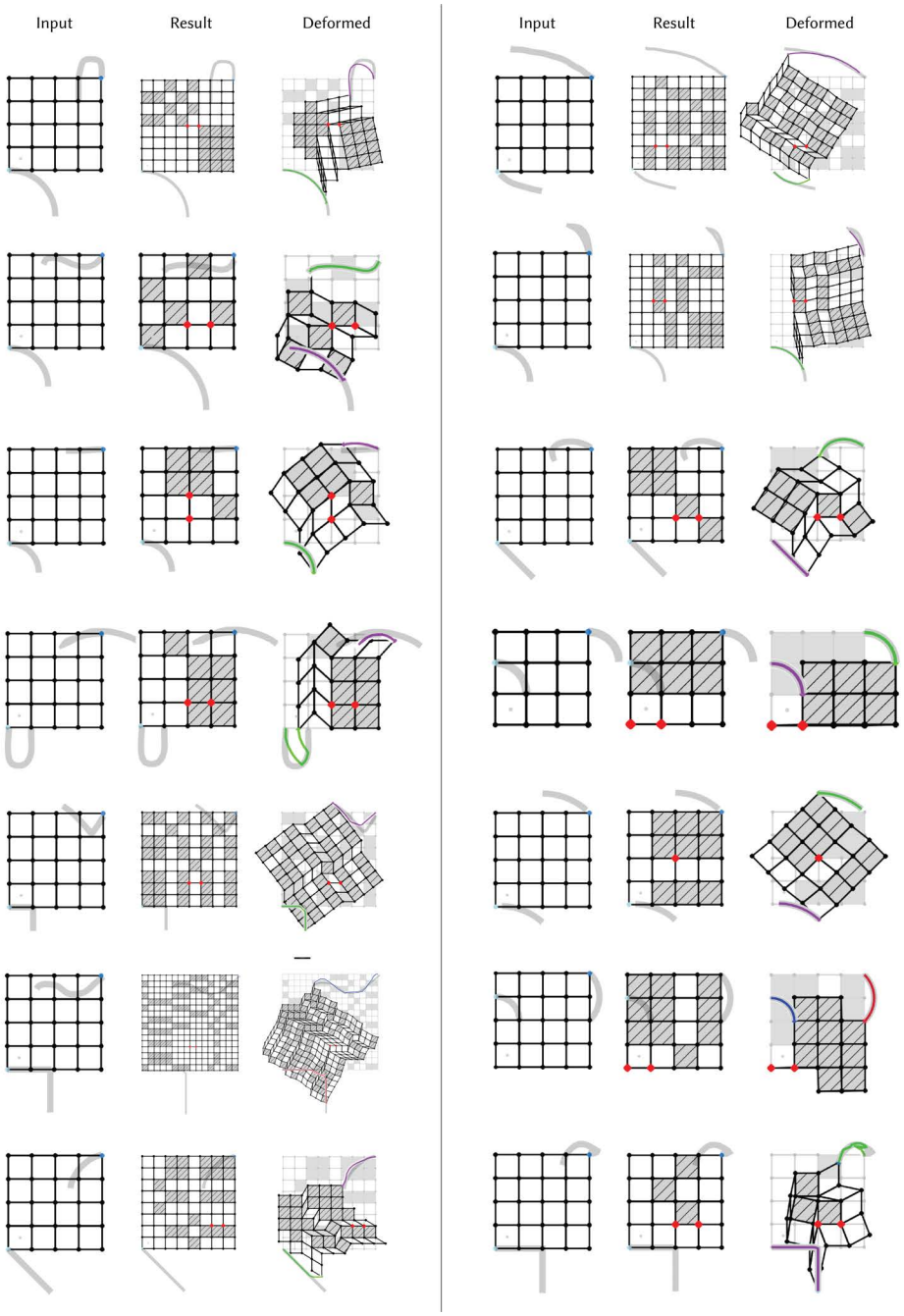

Figure 4.21: Examples of auto-generated metamaterial mechanisms.

5

Digital mechanical metamaterials

Metamaterial mechanisms, like all *analog* machines, are limited in terms of complexity. As forces are passed on from one cell to the next, they are damped and the activation energy dissipates, which causes the mechanical "signal" to decay exponentially. This limits the number of mechanisms that can be concatenated and therefore the complexity of the machine.

In this work, we explore how to extend this concept towards *digital* mechanisms. Combining metamaterial mechanisms with concepts from mechanical computing and mechanical signal propagation [101, 129], we introduce a new type of cell that propagates a digital mechanical signal, i.e., it counteracts signal decay and thus allows signals to pass through an arbitrary number of cells. We extend this basic mechanism to implement simple logic functions.

To illustrate this concept, Figure 5.1 shows a combination lock implemented using digital metamaterials. The device offers ten digit buttons on the front. Users tap these buttons to enter their code, then press the 'open' button to unlock the door.

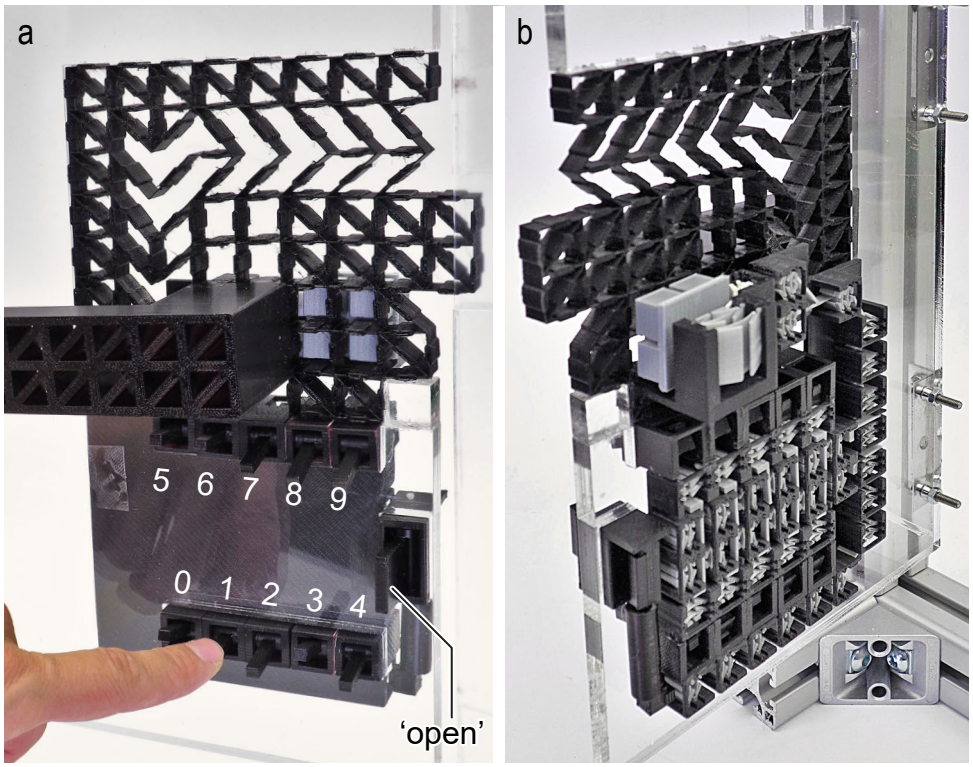

Figure 5.1: (a) This combination door lock is implemented as a *digital mechanical metamaterial*, i.e., a single block of material based on a regular grid of cells. It allows users to input a numeric code, it processes the code, checks its correctness, and unlocks the latch. (b) Under the hood, the lock consists of an array of cells that transmit and process a mechanical signal.

5.1 BASIC CELLS OF DIGITAL MECHANICAL METAMATERIALS

Digital metamaterials are based on a new type of cell that propagates a mechanical signal reinforced by an embedded bistable spring.

5.1.1 The bit cell is the main underlying mechanism

Figure 5.2 shows the key element behind digital metamaterials, which we call *bit cell.* Bit cells contain a bistable spring, which allows them to take on two discrete states. Figure 5.2a shows the bit cell in its *tense state.* When triggered, the spring discharges, causing the cell to switch from its (a) *tense* state to its (b) *relaxed* state.

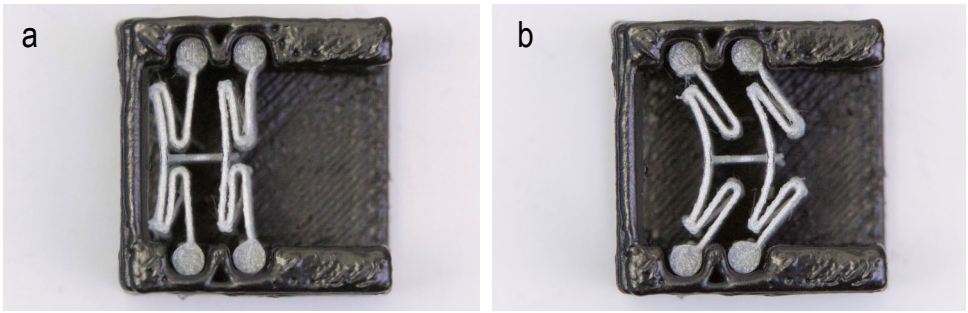

Figure 5.2: When triggered, this *bit cell* changes its state from (a) tense to (b) relaxed.

As shown in Figure 5.3, bit cells feature an input port and output port. A mechanical impulse that reaches the input port triggers the cell, which creates an impulse at the output port. Because discharging the spring releases mechanical energy, the impulse at the output port is larger than the required trigger impulse at the input port.

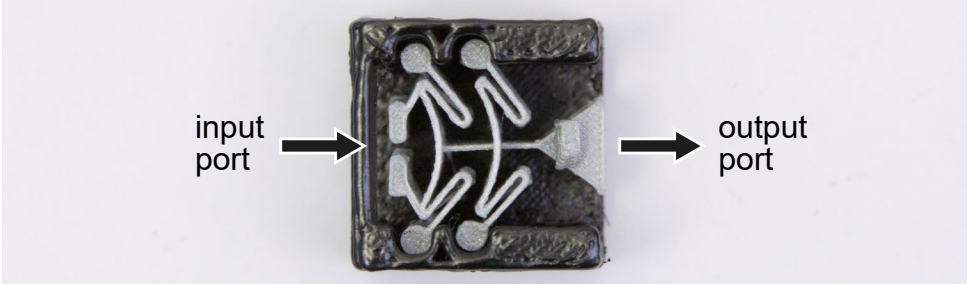

Figure 5.3: Bit cells offer an input and an output port.

As illustrated by Figure 5.4, this allows us to concatenate bit cells in a way that allows cells to trigger their immediate neighbors, resulting in a simple signal propagation mechanism similar to [129].

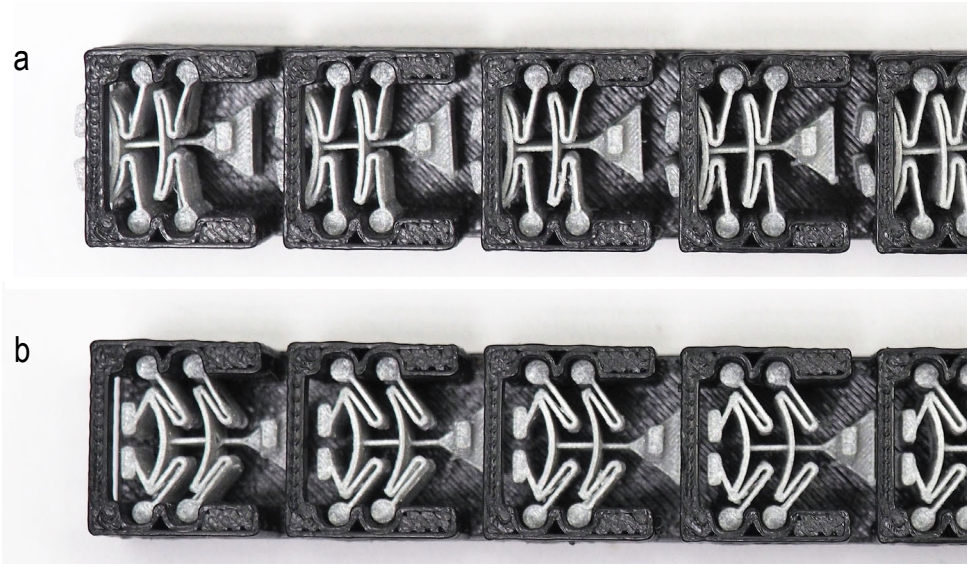

Figure 5.4: Concatenating bit cells creates a signal transmission. (a) Initially all bit cells are in their tense position. (b) Triggering the leftmost cell causes the signal to propagate through all cells from left to right.

5.1.2 The combination lock example

Bit cells and the resulting concept of signal propagation allow us to implement a hierarchy of digital mechanisms of increasing complexity. We discuss these logic functions and mechanisms in full detail later in this chapter, as well as a simple manual recharge mechanism to set discharged springs back into their tense state. However, Figure 5.5 provides a rough overview of the different elements that implement the combination lock.

(1) To input the code, users tap one of the digit buttons on the front, which changes the state of the *digit evaluation* cells. The device contains 10 of these—one for each possible digit. (2) When the user pushes the 'open' button, three signal transmission lines are set off simultaneously; two of

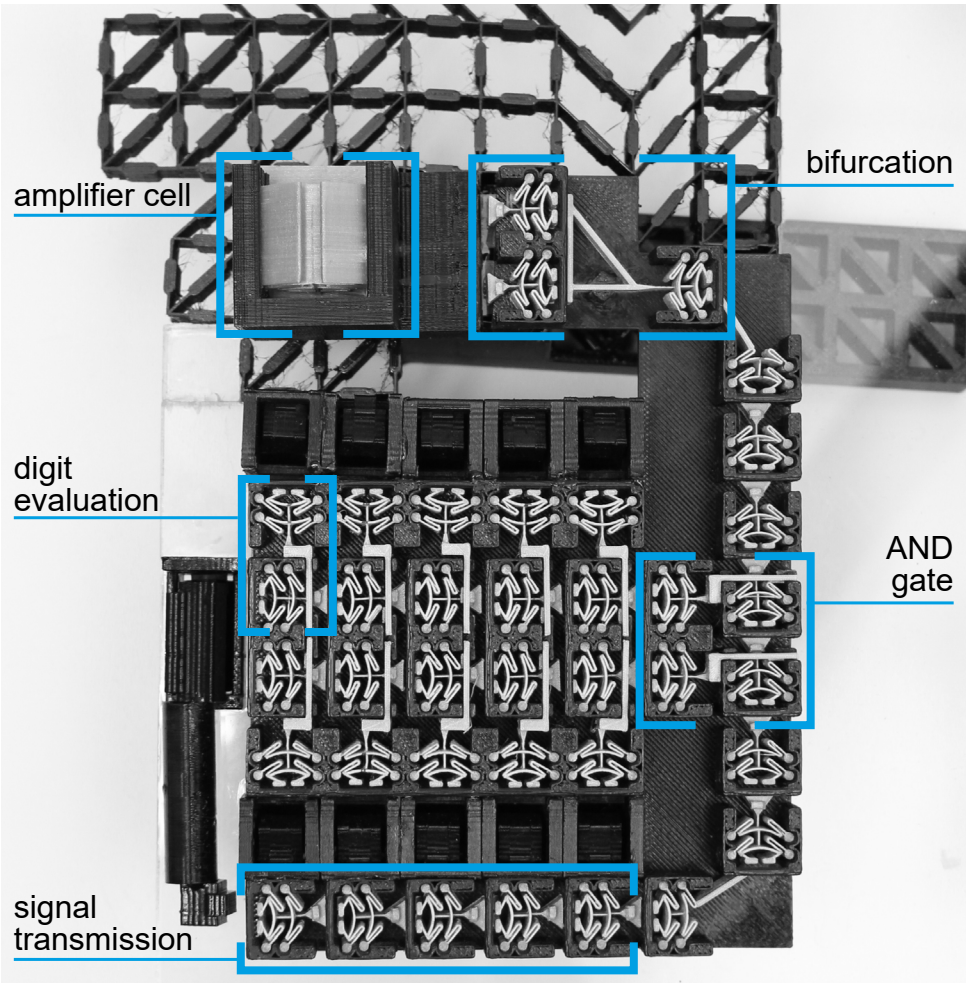

Figure 5.5: Our door lock consists of 82 cells, which implement the signal transmisison, the evaluation of each digit input by the user, an AND gate, and one amplifier cell with a pre-amplification step to move the blocking bolts sufficiently.

which run through the digit evaluation units and (3) set the state of the *AND gate.* The AND gate evaluates the correctness of the code by computing a logical AND the two rows of digits input by the user. The third signal transmission line runs from the bottom left towards the right, around the corner, and upwards where (4) the signal is *bifurcated.* This allows triggering (5) a double-sized *amplifier cell* that actuates the bolts to unblock the door.

Figure 5.6 shows a close-up of these bolts. (a) As long as the bolts are in place, they prevent the shearing cells in the middle from shearing, thereby blocking the door. (b) When the bolts are retracted, the shearing cells can shear and pushing down the handle retracts the latch—as we discussed in Chapter 3. This is where our digital metamaterials connect to the analog metamaterial door latch mechanism.

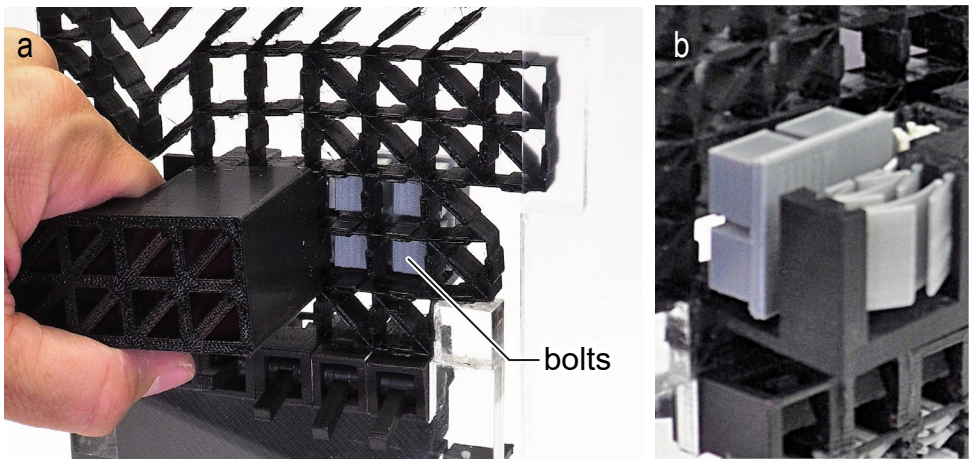

Figure 5.6: We effectively lock the latch mechanism by stiffening the shearing area that enables it. We do so by inserting bolts. Once users entered the key code correctly, our lock signal retracts the bolt and enables the latch mechanism.

5.2 CONTRIBUTION, BENEFITS AND LIMITATIONS

Our main contribution is the concept of digital mechanical metamaterials. They allow integrating computational abilities into the structure of 3D printed objects. We provide a modular system consisting of digital cells (hardware) and an editor (software) that provides a toolkit to users, enabling them to create new digital mechanisms.

While analog metamaterial mechanisms are subject to damping, which causes the mechanical "signal" to decay exponentially and limits the number of 'steps' that can be performed, *digital* mechanical metamaterials enable transmitting signals through an *arbitrary* number of cells.

When we contrast digital mechanical metamaterials to the traditional approach of augmenting objects with electronic microcontrollers, sensors, and actuators [138], our approach results in an entirely mechanical solution and can be produced entirely using a 3D printer. However, since our approach lacks loops, clocks, and memory, our approach is limited to much simpler devices.

5.2.1 Scope of this concept

This work, more than the other works in this thesis, is a thought experiment, a what-if question, rather than a potential solution to a potential problem. Our digital metamaterials are by no means intended to replace computers. We aimed to investigate the question, whether the habit of using microcontrollers for very simple evaluation or computations is really necessary, or if we can perform simple operations within the metamaterial structure.

The scope of our current implementation spans *combinational circuits* (ger. Schaltnetz). Our cell-based material can perform any combinational logic, as we will demonstrate a NAND gate in Section 5.3.2. Furthermore, like combinational circuits our metamaterials present time-independent logic, i.e., the output is a pure function of the present input only.

This work could be expanded to sequential circuits (ger. Schaltwerk) by engineering a protected region that is unaffected by recharging the mechanism and can thus act as memory, and by introducing an external clock signal.

5.3 SIGNAL PROCESSING BASED ON CELLS

5.3.1 Routing signals

In this and the following section, we now show the individual cells that implement the combination door lock we showed in Figure 5.5. We begin with the cell types that allow us to route signals through 3D objects. We already looked at signal propagation along a straight line (Figure 5.4); in this section, we demonstrate how to route signals around corners, across other signal lines, and how to bifurcate signals.

Routing signals is important because 3D printed objects can have arbitrary shape and routing allows transmitting a signal from where it emerges to where the information is needed. For the door lock, for example, we route users input from the digit inputs to the door latch mechanism—which is located elsewhere in our object.

The more specialized routing cells are all based on *bit cells.* However, we position their output ports to be oriented towards the neighbor cell we want to trigger. So while the bit cells in Figure 5.4 feature an output port on the side opposite to the input port, the cell shown in Figure 5.7 redirects the signal by 90° by adding a beam to the arm of our bistable spring. This beam rotates with the arm of the spring, allowing it to tap the input port of the rotated cell on the top right.

As illustrated by Figure 5.8, we can route signals in 3 dimensions by concatenating multiple such mechanisms. Here we route the signal from the x/y plane to the x/z plane to the y/z plane.

Figure 5.9 shows a specialized three-cell mechanism that allows two signals to pass each other in minimal space. We use a crossbar that reaches from the output port of the left cell to the input port of the right cell that spans across the middle cell.

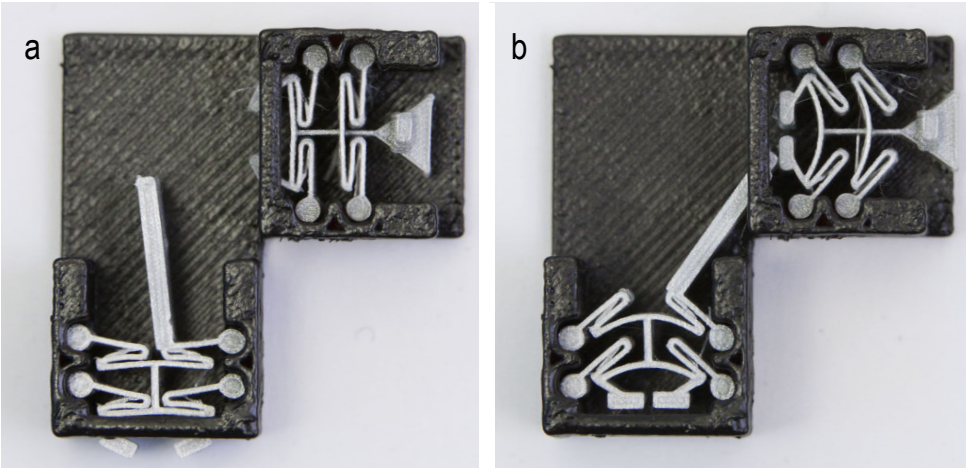

Figure 5.7: We use a new type of output port to redirect the signal by 90°. We exploit the rotational movement of the spring and attach a beam that taps its neighbor cell.

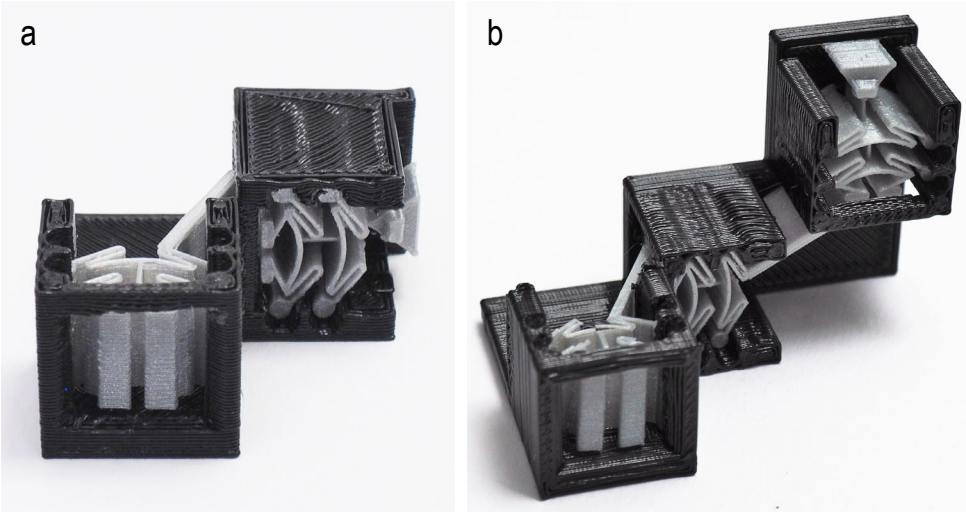

Figure 5.8: (a) Rotating the receiving cell allows us to redirect signals from one plane to another. (b) Concatenating three assemblies routes signals in 3D.

Figure 5.9: We cross signals by running a crossbar across another cell.

Figure 5.10 shows two mechanisms that bifurcate signals. The design shown in (a) triggers two parallel signal lines. The design shown in (b) triggers two signal lines oriented in opposite directions. Both designs exploit the fact that our bistable springs require less energy to be triggered than they output, which allows triggering two cells from one.

Figure 5.11 shows how we merge two signals. This is an interesting construct, because it implements an OR gate.

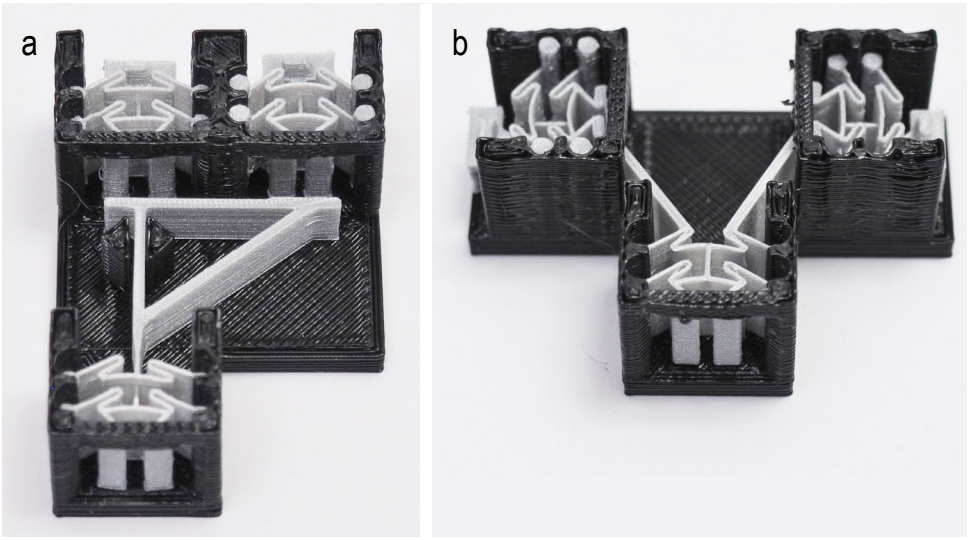

Figure 5.10: We can bifurcate signals (a) in a parallel manner or (b) let the two signals run in opposite directions.

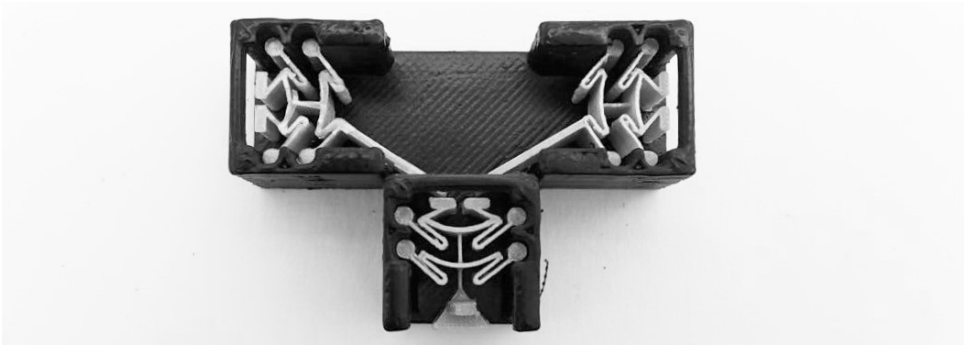

Figure 5.11: We use the opposite assembly to merge signal as we did to bifurcate them. This implements an OR gate.

5.3.2 Logic functions

To implement logic functions, we need to go beyond merely transmitting signals to also evaluating signals, which we achieve by selectively blocking them. In the combination lock from Figure 5.5, we block signals for wrong digit inputs so that the door stays blocked. Later in this section, we present cell arrangements that implement basic logic, such as AND or NAND.

Blocking signals using gate cells

To allow for asynchronous input, we have designed cells capable of storing the first input that reaches them and do not act until the last signal has been received. We call these cells *gate cells*. Our approach is based on rod logic [82].

As illustrated by Figure 5.12a these cells work by placing a "blocker" across their neighboring cell. When the cell on the right is triggered before the cell on the left, the blocker is aligned with the output port of the right cell so that it cannot pass and signal is blocked. However, triggering the left cell, as shown in Figure 5.12b, moves the blocker out of the way and the signal can pass through. The position of the blocker can also be defined to initially let signals through and only after actuation to block signals, as shown in Figure 5.12c-d.

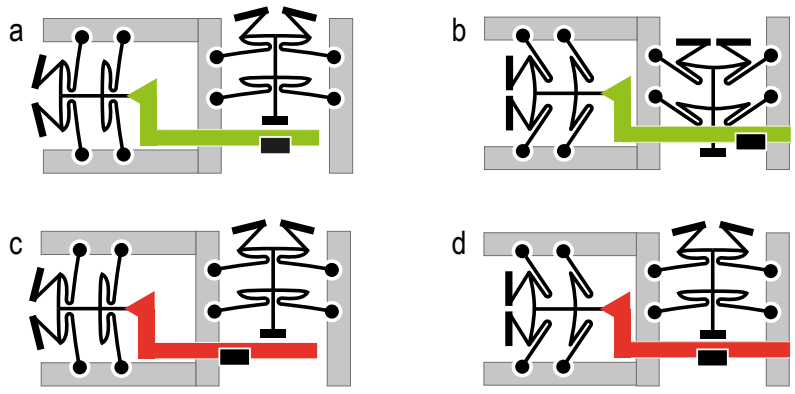

Figure 5.12: *Gate cells* validate signals and can be configured to block signals (a-b) or let signals pass (c-d) in their tense state.

Figure 5.13 shows the design of the two cells that form the gate cell. Each crossbar has a blocker attached on its underside.

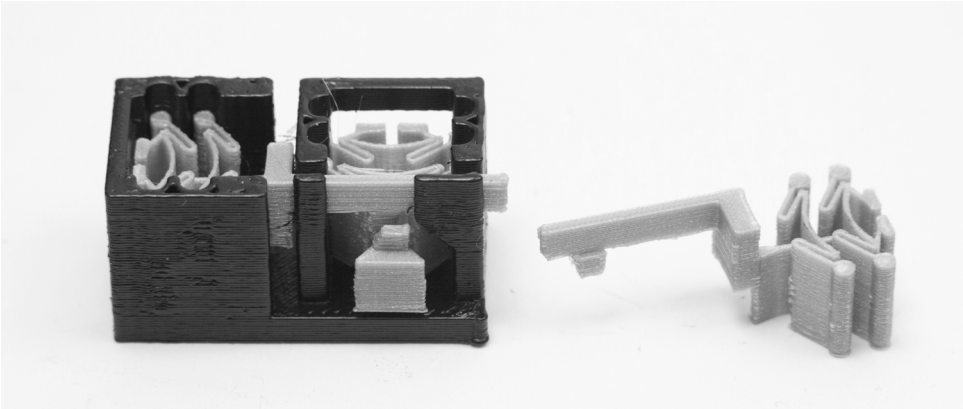

Figure 5.13: We position a blocking element that is intended to either block the signal output or let it pass.

Logic functions based on gate cells

We can concatenate multiple gate cells to create combinational logic functions. Figure 5.14 uses simplified symbols to illustrate how the positions of the blockers are configured to implement the function $A \wedge \neg B \wedge C \wedge D \wedge \neg E$. The positive input cells A, C, and D need to be triggered to move the blocker out of the way and let the signal pass. The negated inputs B and E are implemented by positioning the blocker so that they let the signal pass when they are *not* triggered and block the signal otherwise.

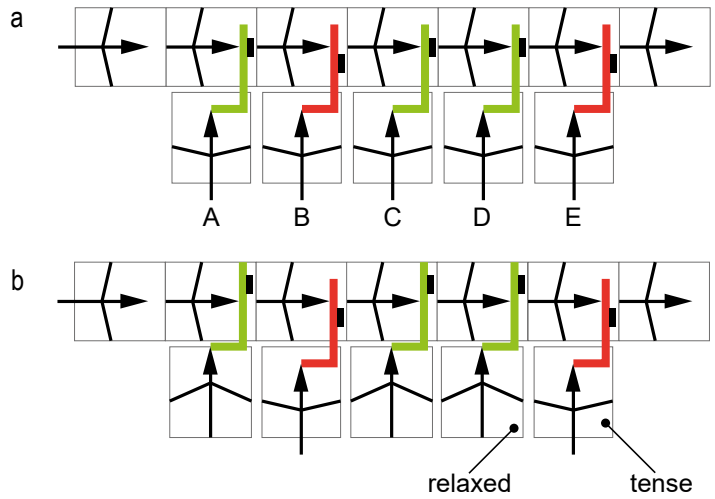

Figure 5.14: (a) When all inputs are tense, the signal cannot pass. (b) Triggering the correct inputs, here A, C, and D, moves the blockers so that the signal can pass.

If we rename the inputs of the logic function that is shown in Figure 5.14 from A–E to 0–4, it implements a 5-digit code evaluation. To implement the combination lock with 10 digits, we add a second row of inputs. Now we have two logic functions (one in each row), which both need to be correct, thus we add an AND gate. Figure 5.15 illustrates that the key code is '0 2 3 8'.

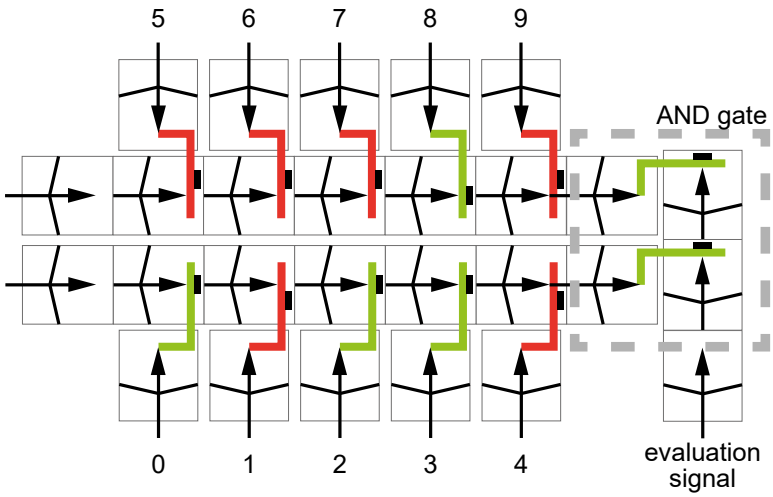

Figure 5.15: We add an AND gate to validate the two rows that yield the 10-digit.

Again, we use our *gate cells* to employ the AND gate. Each code evaluation row has a gate cell at the end. Only if all the inputs of the corresponding row were correct, the blocker is moved out of the way for a third signal to pass, the *evaluation signal.*

Combinational logic using an evaluation signal

Our implementation of the AND gate has three inputs, namely two values and one additional *evaluation signal.* We add this additional signal because our mechanical computation is fundamentally different from electronic circuits yet adding only one single signal allows us to implement any Boolean predicate without any electronics.

A 'signal' within our system is not an applied voltage, but an *impulse,* i.e., a mechanical force within the object. This impulse *changes the system state* by changing physical properties of the material, such as the position of the blockers. Since we block invalid signals, the output of gate cells is no signal instead of a 0-signal (logical low). However, not receiving a signal is indistinguishable from a dormant system. This means that we cannot provide an 0-signal that can serve as an input to the next gate, as in classical electronic circuits.

Despite this, we are still able to employ combinational logic within our materials. The most space efficient way is to integrate the inversion into logic functions, e.g., by using a NOR instead of an OR. Figure 5.16 illustrates a selection of logic gates implemented with our digital cells. Note that we show a different OR gate compared to the one shown in Figure 5.11. The one shown here uses the general-purpose assembly that is also used in the NOT and NAND gate.

We add the additional evaluation signal as a second computation step. After the inputs are provided to the system, we send a signal off that evaluates the inputs in order to produce output. Independent of the complexity of

the logic, only one single evaluation signal is necessary, since it can be furcated, merged, and routed through the material. Race conditions within operations can be resolved by adapting the length of signal paths.

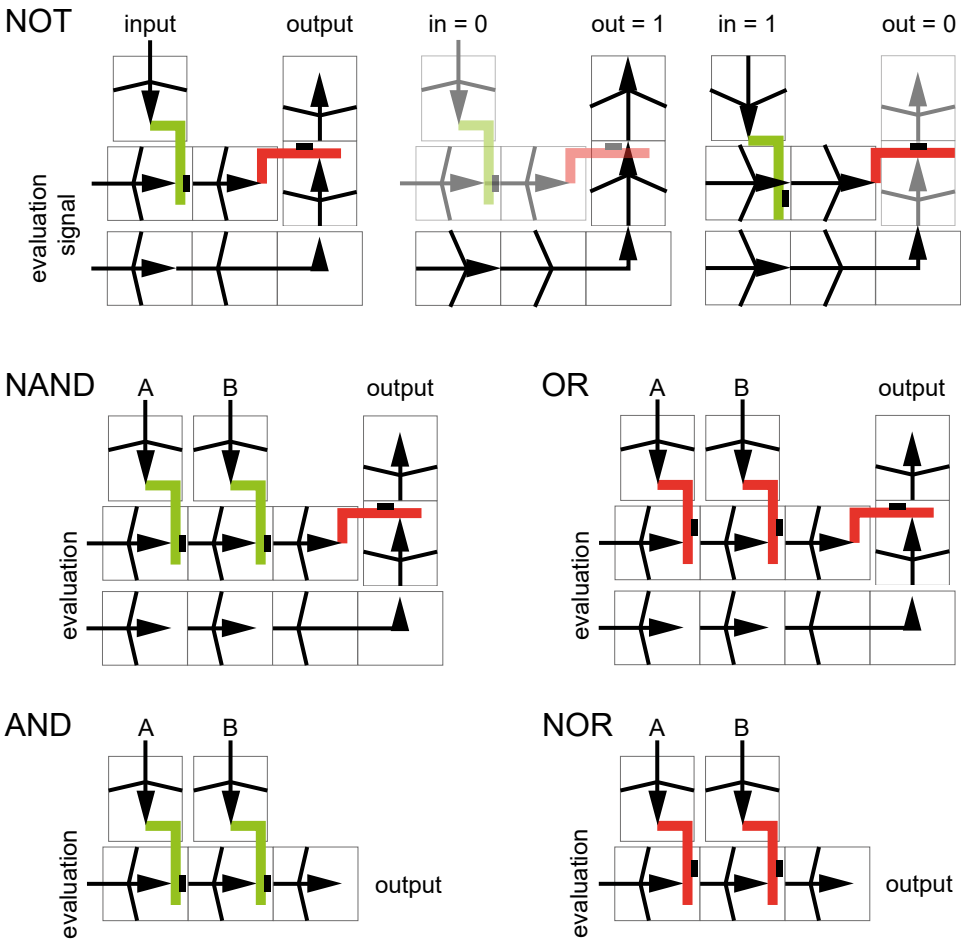

Figure 5.16: Only one additional signal allows us to implement combinational logic, despite not having a traditional 0-signal.

5.3.3 Amplifying the output

While the cells that implement the signal transmission can be arbitrarily small, the *output* cells that move material to change the material properties might need to produce a certain amount of movement or force.

In the example of the door lock, we need to move the bolts sufficiently far into the door latch's structure to stiffen it. We use what we call an *amplifier cell,* which is inspired by the metaphor of operational amplifiers in the electronics domain. Such an amplifier cell, as shown in Figure 5.17, is a cell that is doubled in size. This allows us to add a larger spring to produce more stroke length.

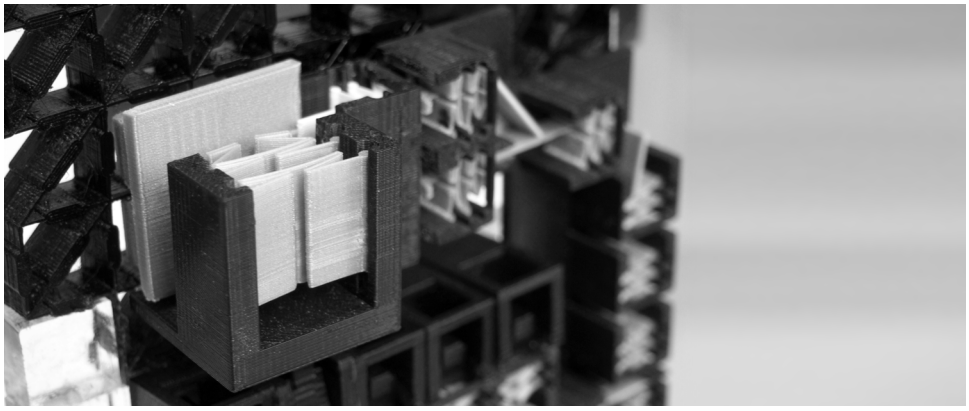

Figure 5.17: We amplify the stroke length of our output by going from small cells to a double-sized cell.

To transition from small cells to bigger cells, we bifurcate the signal. This gives us the energy of two cells, which together trigger the spring within the amplifier cell. In our door lock example, our 30 mm amplifier cell moves the bolts by 6 mm as compared to the stroke length of 3 mm of the 15 mm bit cells.

5.3.4 Recharging

After the springs were triggered and they are in their relaxed state, they need to be reset to their tense state before the computation can be run again. To do so, we designed a small lid on top of each cell, which uses the cell's third dimension to recharge the spring. Figure 5.18 shows how as the lid is pushed down, the attached wedges move the spring backward to its tense position. We use an additional plate to push multiple recharge

lids at the same time. This design enables one recharge action for every plane of computation. We added a small bump on the underside of the lid, which causes the lid to spring back upward in order to not hinder the signal transmission between the cells.

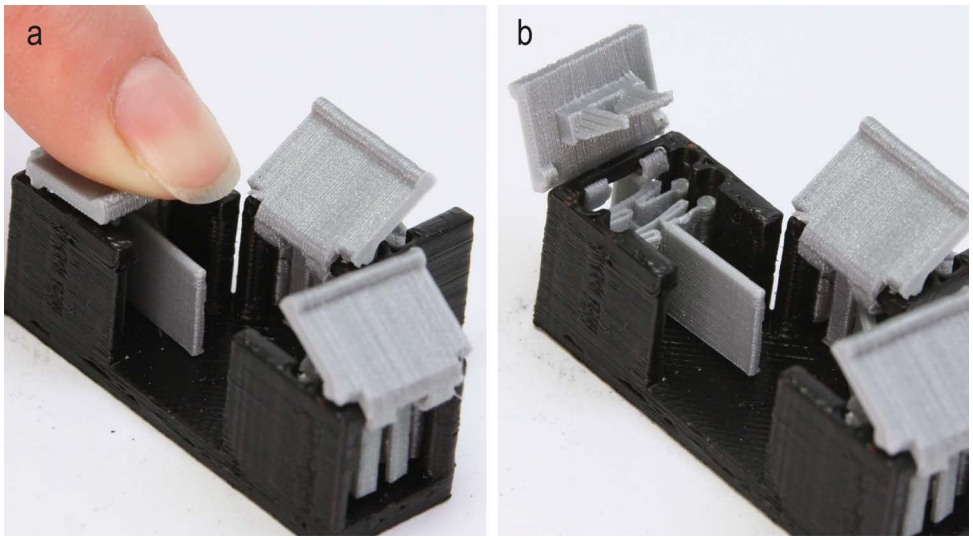

Figure 5.18: (a) Each cell features a lid with wedges, pushing it (b) recharges the spring underneath.

5.3.5 Scope of Applications

We see digital mechanical metamaterials being particularly useful for objects that have (1) many mechanical inputs (e.g., the code lock), and/or (2) many mechanical outputs (e.g., the following example of a plant pot), and (3) which are not frequently reconfigured. For example, the density plant pot might be reconfigured when seasons change, or the door might be locked once a day. In contrast, for objects that require frequent updates (e.g., displays) or more complex programming involving loops, etc., we recommend traditional electronics.

Additional example: plant pot

Figure 20 shows an example of a plant pot, where (a) users input the plant's size (small–large) and its water demands (little–much) using sliders. This triggers the computation in the bottom layer of the pot, which determines (b) how many *density cells* will be closed. After users configured the pot's density, (c) they place it into a cachepot with water. The density of the plant pot now determines how fast water can pass through to the plant.

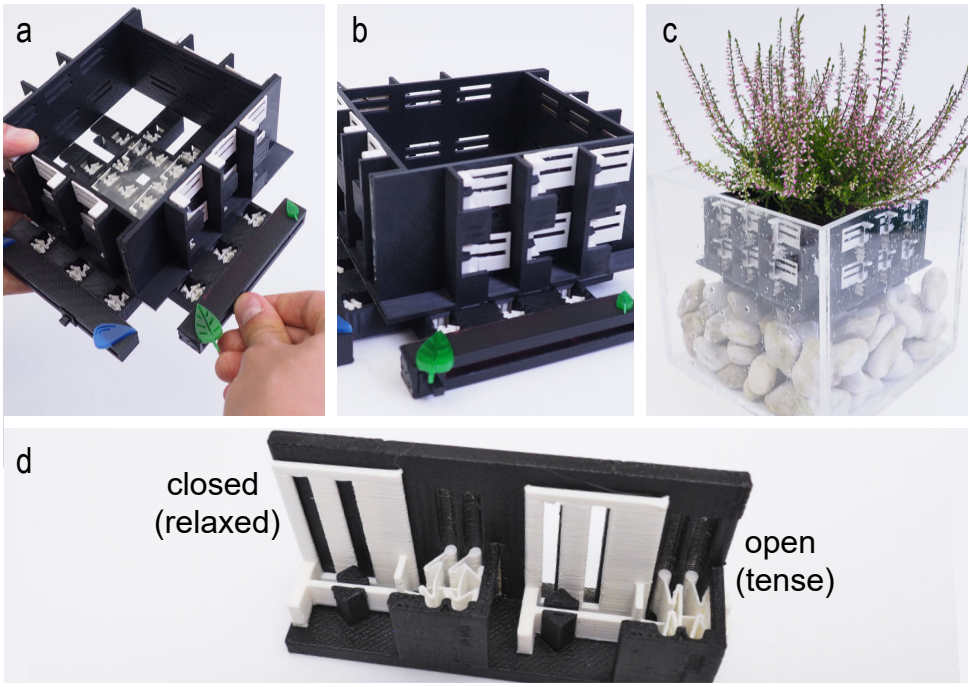

Figure 5.19: We implement a plant pot that changes its density based on user input of the plant's size and water demands.

We use *gate cells* to change the weight of the parameters of the plant pot example. Figure 5.20 shows that by simply placing gate cells along the diagonal, we give more weight to the low values of the parameters. Since the gate cells prevent the signal from passing through, they prevent all

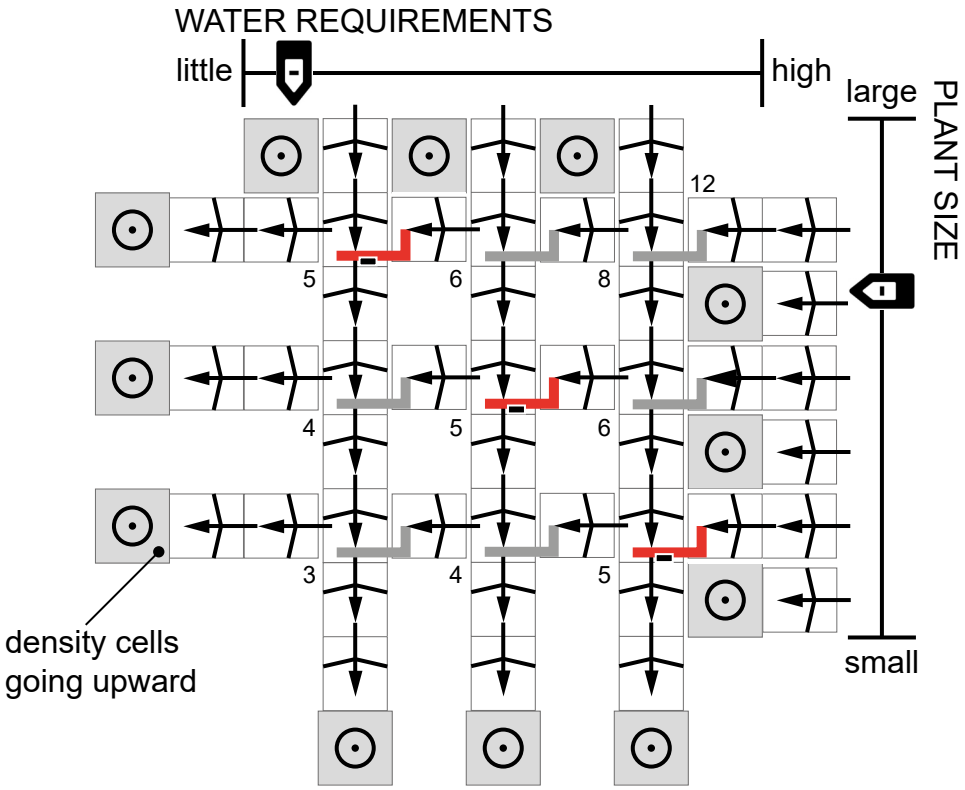

Figure 5.20: The weighted computation of the plant pot's density ensures that small plants get enough water by preventing some density cells from closing. The numbers indicate how many density cells are open for each parameter combination.

density cells from closing, so that even for a small plant with little water demand the plant pot's permeability is 25% (3 out of 12 density cell rows remain open).

5.4 TECHNICAL DETAIL ON THE CELLS

5.4.1 Fabrication of cells

We print our prototypes from the commonly available filaments ABS and PLA. While our cells are designed to be printed in an assembled state, we tend to print the parts of our prototypes separately. This allows us to print all elements without support material, which tends to be faster than the

single-part design that requires dissolving the support material. We printed the springs from PLA using the *Ultimaker 2+* 3D printer, and the frames that hold the springs from ABS on our *Stratasys Dimension SST 1200es.* Our cell size is 15 mm for all our prototypes with a printed spring thickness of 0.4 mm.

The cell shown in Figure 5.21 is printed in an assembled state. We had it made at *shapeways* using their "frosted detail plastic" material, which is a UV cured acrylic polymer that is printed using the MultiJet Modeling process.

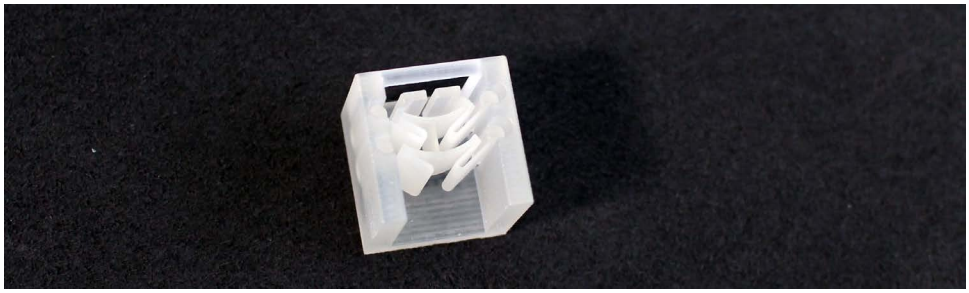

Figure 5.21: A cell printed fully assembled using shapeways' "frosted detail plastic" material.

We empirically tested how the cells miniaturize while retaining the same stress values using *Autodesk Fusion's* simulation. The results showed that reducing the spring thickness to $\frac{1}{2}$ allows it to be shortened to $\frac{1}{4}$ of its length, i.e., to $\frac{1}{64}$ of the cell volume. For example, a 0.2 mm thick spring allows for a cell size of 3.75 mm, which is a matter of printer resolution.

5.4.2 Bistable spring design

The bistable spring in our cells differs from a typical bistable spring that is shown in Figure 5.22b, which is a simple pre-bent beam that is fixed within rigid walls [124]. However, such designs have very high width-to-length ratios, which do not utilize the space within a rotation-invariant cubic cell well.

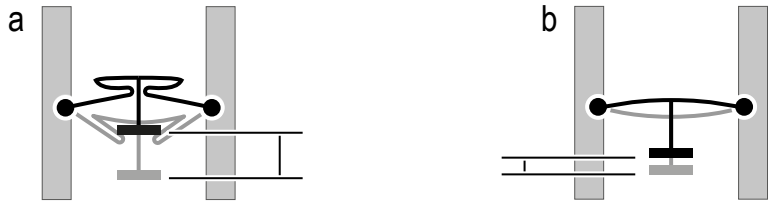

Figure 5.22: (a) The spring we use in our bit cells is longer and thus weaker than (b) conventional bistable springs. Or, for the same force, our cells produce more stroke length.

Figure 5.22 illustrates that our spring design includes an additional 'loop', which prolongs the beam and therefore makes it weaker, i.e., it requires less force to be triggered and charged. We measured 45% less force required to charge our type of spring compared to the conventional spring. Another way to view it is that our longer springs produce more output length (by 23% according our measurements) while requiring similar force.

Note that our cells incorporate two connected springs. This is a common technique [124] for increasing the stability of bistable springs during the so-called 'snap-through', i.e., the point where the spring is compressed the most as it is forced to its other second position.

5.4.3 Technical evaluation

The geometry of our spring allows us to make limited changes in stroke length and force by varying the spring parameters. For example, we used slightly stronger springs in the plant pot example to compensate for the higher density of water. While the output of bit cells usually needs to be only strong and far enough to trigger the neighbor cell, the output cells may have to meet specific requirements in terms of amount of force or stroke length.

Our evaluation informs the geometrical spring parameters for achieving bistability and the maximum possible fan-out of a cell, i.e., how many cells can be triggered by one single cell.

Independent variables. We compared a total of 75 springs of our design where we varied three parameters independently illustrated by Figure 5.23: (1) the arm angle, (2) the length of the bent bridge in the middle, and (3) the strength of the bridge, varied trough changing its buckling magnitude and its thickness concurrently. We varied the values for the bridge strength from $1.05\times$ the normal spring thickness to $1.85\times$, and a buckle distance from 8% of the bridge length to 40%. The spring thickness is limited by the 3D printer's resolution; we use 0.18 mm. Bridge length values ranged from 45% to 85% the total distance between the walls. We tested these values for 20°, 30° and 40° arm angles. This yields 25 springs for each arm angle.

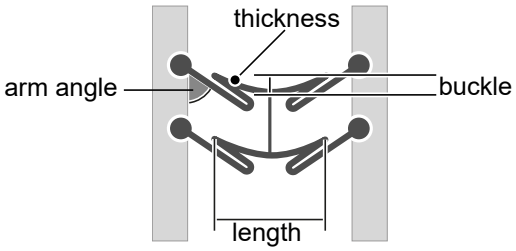

Figure 5.23: We vary the parameters of bridge length and bridge strength for three different arm angles each. We measure the stroke length and the forces for charging and triggering the springs, as well as their output energy.

Dependent variables. We measured (1) the force it takes to push a spring to its tense position, (2) the force necessary to trigger the spring, (3) its stroke length, and (4) the force it outputs when triggered.

Test setup. Figure 5.24 shows our test setup. We placed a ruler (error 0.5 mm) under the spring to measure the stroke length. We used a force gauge with an error of 0.05 N, which was constrained to linear movement centered to the spring and moved by a threaded rod. We pushed the force gauge against the spring to measure the charge energy, we released the pressure while slowly moving the force gauge backward to measure the output energy. We measured the trigger energy by pushing the spring.

Figure 5.24: We measure the forces using a force gauge (error $0.05\,\mathrm{N}$), and the stroke length using a ruler (error $0.5\,\mathrm{mm}$).

5.4.4 Results

Figure 5.25 shows charge, output, and trigger energy and stroke lengths for 50 springs. Empty fields denote springs that were not bistable. The results for springs with a 20° arm angle were omitted since only 2 of them were bistable.

All four measured values increase when increasing the arm angle or the bridge strength, or when decreasing the length of the bridge. The *output energy* was on average 73% of the charge energy for 30° springs and 66% for 40° springs.

The difference between output energy and *trigger energy* is greatest right when the springs start becoming bistable. The ratio between the two decides the maximum possible fan-out of the springs, thus a 2:1 ratio is necessary for bifurcation. Choosing a higher trigger energy however increases the fault-tolerance of the system with regards to unwanted activation, e.g., by dropping the object. *Stroke length* is affected most by the arm angle, i.e., the stroke length increases with the arm angle. Stroke is least affected by the strength of the bridge. In contrast, *charge energy* of the spring is affected most by the strength of the bridge and least by the arm angle, which can also be seen from Figure 5.25 in the rapid changes along the y-axis.

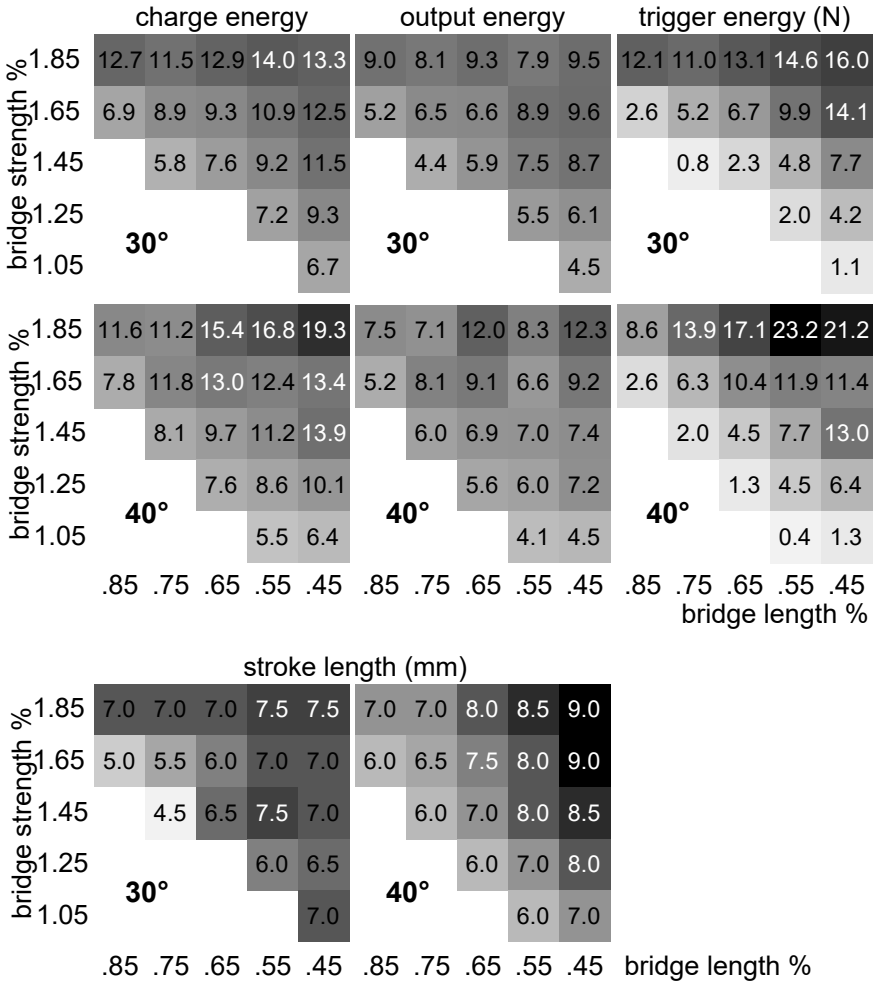

Figure 5.25: Raw results of our technical evaluation for charge, output and trigger energy in N and stroke length in mm. Missing values indicate non-bistable springs.

Choosing appropriate values for each can tune the spring toward a longer stroke or a higher output energy without changes to its bistability. Note that these values apply to the springs we tested with and that due to differences in manufacturing they might vary slightly.

5.5 EDITING DIGITAL MECHANICAL METAMATERIALS

To allow expert users to create and fabricate objects from digital metamaterials, we implemented a specialized 3D voxel-style editor, which is based on the editor for metamaterial mechanisms shown in Section 3.6.5. The main intent is to allow users to draw signal paths and verify them within the editor (Figure 5.26). We support users by allowing them to enter simple logic functions, which our editor converts to cell arrangements that implement that function. While the editor is built to help users design digital metamaterials efficiently, knowledge about signals and logic remains necessary, i.e., this editor is for expert users.

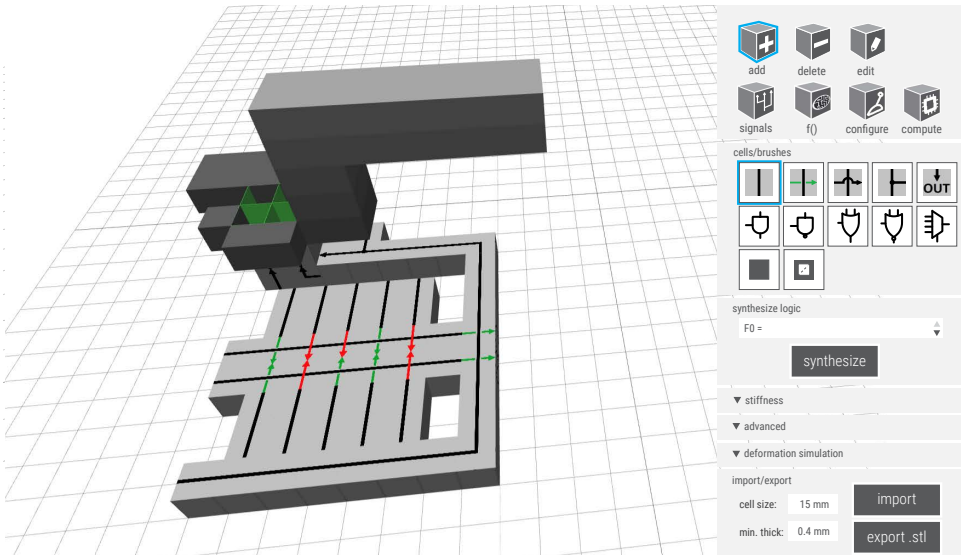

Figure 5.26: Our editor helps users create digital metamaterials.

5.5.1 User interaction

Figure 5.27 illustrates how users create the door lock example from Figure 5.1. (a) They first draw the signal line that evaluates the upper 5 digits by dragging over the ground plane using our "draw signals" tool. (b) Then, using the same tool, they draw signals perpendicular to the first signal line. (c) When the two signals cross, the editor automatically draws a gate cell.

(d) They do the same for the lower row of digits. (e) In this example, users manually configure the gate cells using the "configure", i.e., they change the initial state of 5 gate cells from initially 'pass' to 'block' by clicking on the respective gate cell. (f) The configured gate cells implement the key code for the lock.

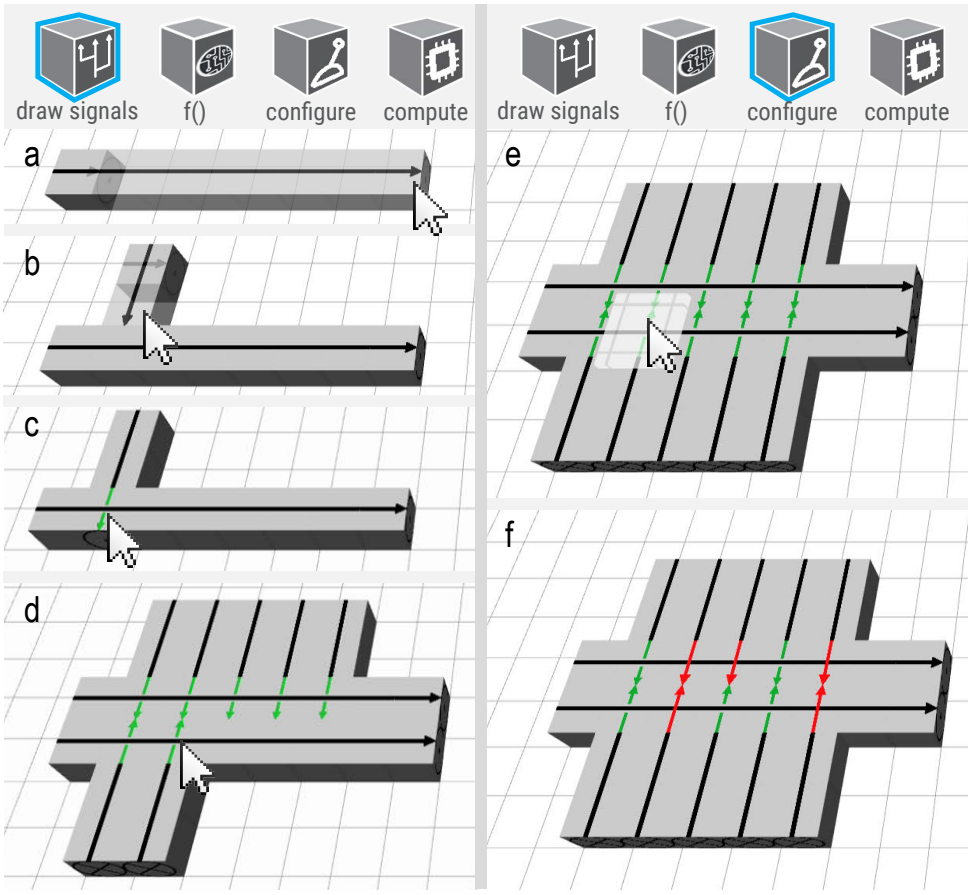

Figure 5.27: (a) Users draw the signal routing using the "draw signals" tool. (b) Once they cross an existing signal route, (c) the editor automatically draws a gate cell. (d) After creating all cells for the digit evaluation, (e) users set the initial states of the gate cells using the "configure" tool (f) to define the key code.

Users continue by adding the evaluation line, the AND gate and the output cells, which will move the bolts. Finally, they model the analog door latch mechanism on top of the digital metamaterial.

Figure 5.28 shows how users verify the signal transmission in our custom editor. They first charge the cells by selecting the “compute” tool. The editor visualizes charged cells by turning the signal lines blue. Clicking on a cell, as shown in Figure 5.28a, sets a signal off. The impulse runs through the cells, being visualized in yellow at the currently active cell. After the impulse has passed a cell, the signal path is shown in black again, because the cell is back in its relaxed state (Figure 5.28b). To verify the whole computational assembly, users trigger the inputs first and then the evaluation signal, as they do on the 3D printed object. They subsequently watch if the signal runs all the way through to the door. If not, they see where the signal stopped and can correct the error.

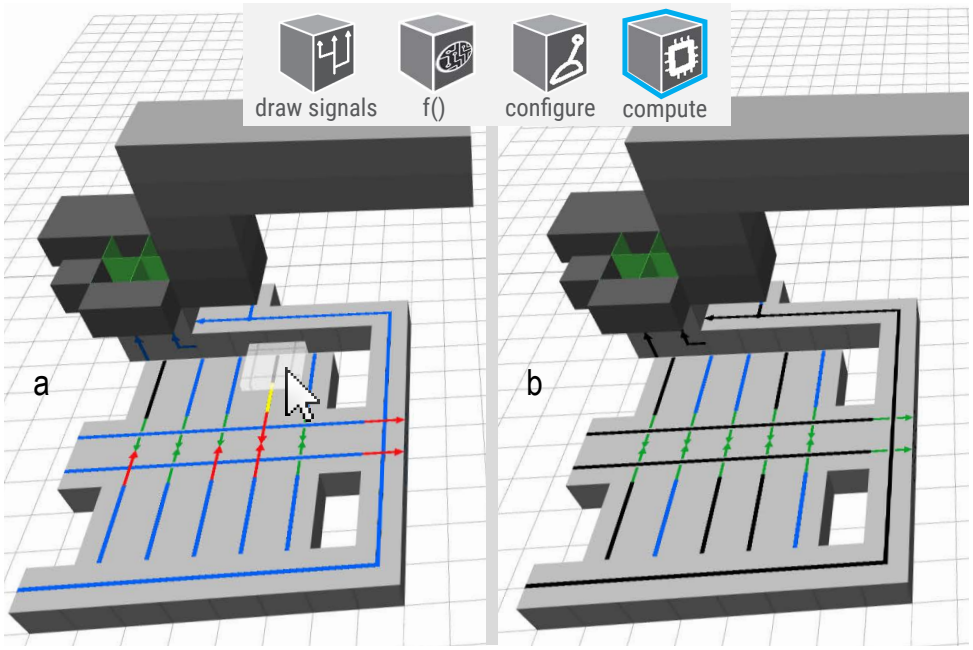

Figure 5.28: Users can verify their logic and signal routing. They first charge all springs, then (a) they click the inputs to trigger the signal there, and lastly (b) they trigger the evaluation line and find that the signal passes all the way through to the latch.

To help users create logic functions efficiently, e.g., by avoiding the need for manual configuration of cells, we allow users to input logic functions. Figure 5.29 shows an example, where users enter the function 'A & ~B & C & D & ~E' and click 'synthesize'. Then, they indicate where the synthesized cell arrangement shall be positioned by simply clicking on the grid. Our editor automatically synthesizes the cells that implement the entered logic functions.

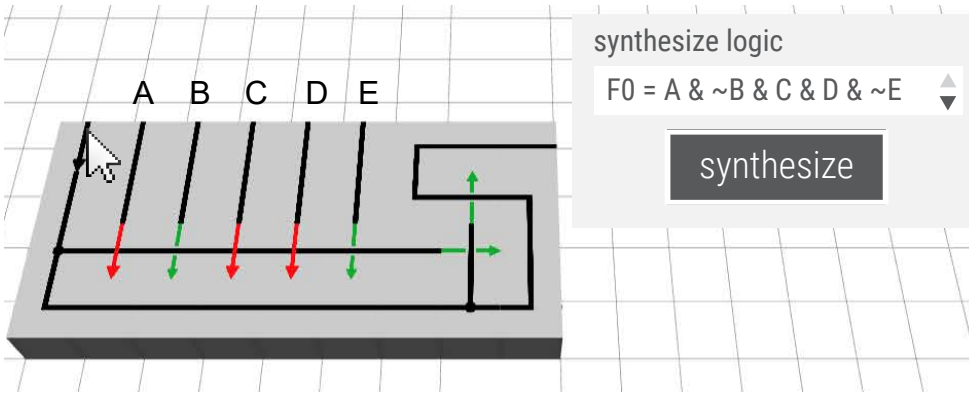

Figure 5.29: Users enter the function 'A & ~B & C & D & ~E', and indicates the location by clicking on the grid. Our editor responds by automatically inserting the corresponding cells.

5.5.2 Implementation

We build on the metamaterial mechanisms voxel-style editor and extend it to allow users to draw signal routes and to input logic functions. Our extension of the editor is based on a node.js javascript framework, using the three.js graphics framework and WebGL for rendering the basic geometries.

Rendering of 3D-printable .stl files was done in modular hierarchical OpenSCAD script files. The editor simply exports the cell geometries as OpenSCAD script commands to render a cell with the specific parameters.

Drawing signal paths

Drawing signal paths is realized through a freeform line tool that places appropriately connected cells following the cursor. We use a pathfinding library for 3D[1] that provides the A pathfinding algorithm to simplify connecting cells via a signal line by finding the shortest available path while crossing existing signals where necessary.

5.5.3 Synthesizing cell arrangements from boolean expressions

To generate cell arrangements of minimal size that implement a user-defined logic function, we use a version of the Espresso heuristic logic minimizer[2].

The logic minimizer parses the user text input, minimizes the described input function and returns it in its disjunctive normal form. This minimized DNF is parsed a second time by our editor to identify its terms and literals, which are used to create a very compact cell arrangement that forms a disjunction of minterm conjunctions. A minterm is a minimal conjunction of the input literals that returns true. This means that an array of signals representing the disjunctions of the function is run in parallel. All input variables of the function intersect and potentially block these lines, forming conjunctions along each of the parallel lines. The combined arrangement implements the function as a whole. To choose the most succinct cell representation of the function, we also minimize the negated input function and negate its result again directly on the cell level. The cell arrangement variant that requires the least cells to implement the input function is constructed and placed in the editor at the last user-selected cell location.

[1] https://github.com/schteppe/PathFinding3D.js
[2] https://embedded.eecs.berkeley.edu/pubs/downloads/espresso/index.htm

The logic minimizer runs in a separate python virtual environment using the PyEDA[3] library for electronic design automation. This python server is queried via HTTP requests to a REST architecture and replies with minimized functions to logic functions encoded in the request-URL.

5.6 CONCLUSIONS

We presented digital mechanical metamaterials. While (analog) metamaterial mechanisms suffer from signal decay, digital metamaterials are not subject to such decay, allowing us to create larger or more complex objects. Unlike solutions based on sensors, actuators, and microcontrollers, our approach still is entirely mechanical.

We showed the design of bit cells, which are the key element for the signal transmission. They contain bistable springs, which have two states, the relaxed and tense state. We also presented cells that allow for routing signals in 2D and 3D. To employ simple computation, we showed gate cells. We demonstrated our approach at the example of two prototypes: a digital door lock without electronics and a configurable plant pot. To help expert users create such digital metamaterials, we contribute an editor that allows for routing signals, verifying them, and synthesizing cell arrangements from user-defined logic functions.

[3] http://pyeda.readthedocs.io/en/latest/index.html

6

Metamaterial textures

We demonstrated in the previous sections that metamaterials can be appropriated to process analog signals in the form of mechanisms and even digital information. The previous concepts were demonstrated at examples of complete devices featuring input handles, the processing part and an output (e.g., the retracting bolt in the door latch example). However, the focus of our previous works was clearly on processing information and forces.

In this work, we apply the main idea behind metamaterials, i.e., subdivision into a large number of cells and customization on a per-cell basis, to the *outsides* of 3D printed objects. The resulting *metamaterial textures* allow designers to shape how the object interacts with the environment and with the tactile sense of the user.

6.1 OVERVIEW OF METAMATERIAL TEXTURES

Metamaterial textures are 3D printed surface geometries that can perform a controlled transition between two or more textures. Haptic properties, such as compliance [145], weight, and static texture can enhance 3D objects and are easy to fabricate. More complex fabrication machines, such as multi-material 3D printers, also allow for continuously controllable textures [43]. However, so far, they do not apply to *objects* and are limited by one texture.

In this work, we introduce metamaterials that undergo a controlled transformation when an external force is applied, resulting in *multiple* dynamic textures. Figure 6.1 shows an example.

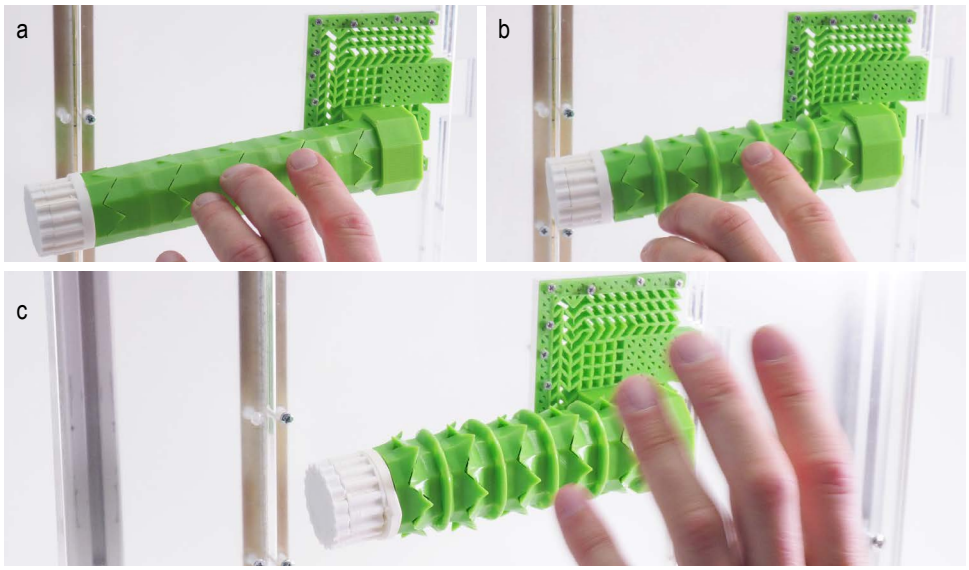

Figure 6.1: When an external force is applied, metamaterial textures undergo a controlled transformation. This door handle, for example, transforms (a) from flat (b) to rippled (c) to spiky, allowing the person behind the door to set a tactile message with three levels of *enter/busy/do not enter* messages for visually impaired or sighted users trying to enter.

This door handle transforms from flat to rippled to spiky, allowing the person behind the door to set three levels of *enter/busy/do not enter* tactile messages for everyone trying to enter. For completeness, we integrated our textured handle with our metamaterial door latch (Figure 3.1).

As shown in Figure 6.2, the inside of the door handle consists of a grid of cells, which controls how the texture on the object's surface will be formed.

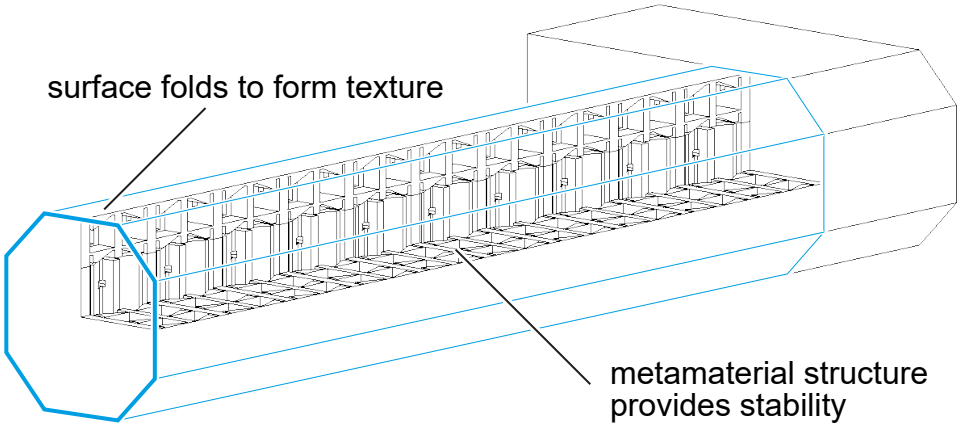

Figure 6.2: Metamaterial textures are made from cells that can fold upwards, creating a tactile bump. The metamaterial allows for this behavior while simultaneously providing stability.

Figure 6.3 illustrates the design of the underlying cells, which we call *fold cells.* Each cell implements a simple mechanism that transforms horizontal compression into vertical deformation, i.e., it *folds* upwards when compressed. The cell consists of two four-bars, which is a basic linkage the rigid members of which move in parallel. When the cell is compressed horizontally it causes the four-bars to shear and the cell to fold upwards, creating a tactile bump. Hence, chaining multiple of these cells allows *popping out a texture* on the surface of the object. These four-bars can be repeated inside the object until it is filled.

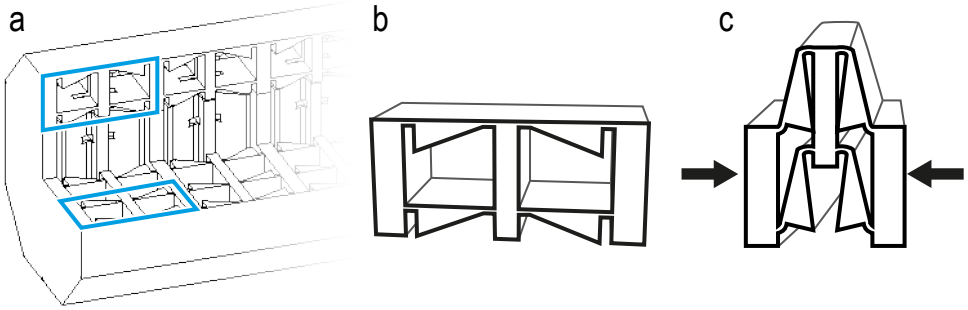

Figure 6.3: (a) Textured objects consist of many (b) unit cells, which (c) pop out of the object's surface, when compressed.

Metamaterial textures are generally actuated to transition between textures by a global compression. To ease user interaction, we deploy them with a mechanism that allows producing the force required to deform the metamaterial texture. Figure 6.4 shows the mechanism we use to actuate the door handle texture in Figure 6.1. The mechanism runs strings through the door handle. As the user turns the knob, the strings are wound up and cause all fold cells to compress and to fold outward, forming the texture. During actuation metamaterial textures compress by a certain amount—30% in the case of the door handle example in Figure 6.1. This means our approach is limited to objects for which length is not a critical property.

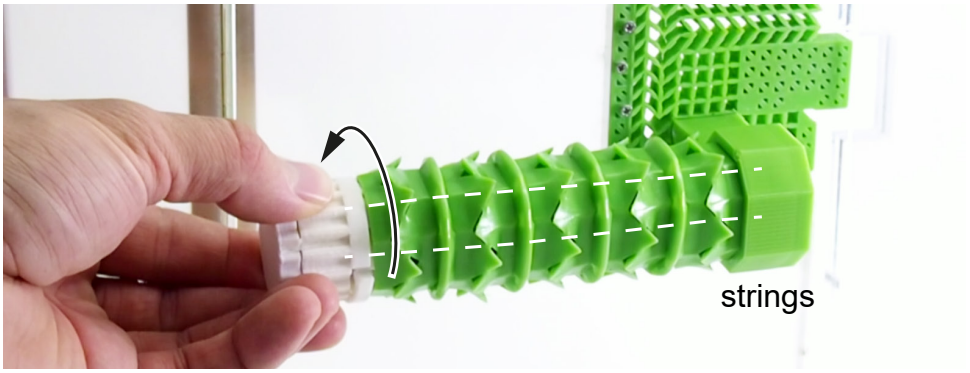

Figure 6.4: Users transition through the door handle's embedded textures by turning the knob. That winds up the strings on the inside, which compresses all cells and forms the textures.

Application examples

Metamaterial textures are suitable for conveying tactile messages or providing tactile feedback, for rapid prototyping of textured objects, or for adapting objects on demand.

Conveying tactile messages. Figure 6.1 showed an example for this category as a door handle that utilizes its surface to inform users (visually impaired or sighted) about one's availability for interruptions. We achieve three textures within the one object by (1) alternating the two textures row-wise and (2) defining the sequence in which they pop out based on the amount of compression, i.e., compressing the door handle halfway only activates the smoother ridges, and compressing it all the way adds the spiky texture to the previous bumpy texture.

Adapting the functionality of an object to the context of use. Figure 6.5 shows a shoe sole with a metamaterial texture. It can be transformed from a flat sole to a corrugated one for more traction on snow or mud. This example illustrates one benefit of these metamaterial structures: they add enough *stability* to the origami-like surface to hold the weight of an adult (here 55 kg). To achieve this, we designed our cell to fold tightly, which prevents the beams from buckling in order to maximize strength (Figure 6.3c).

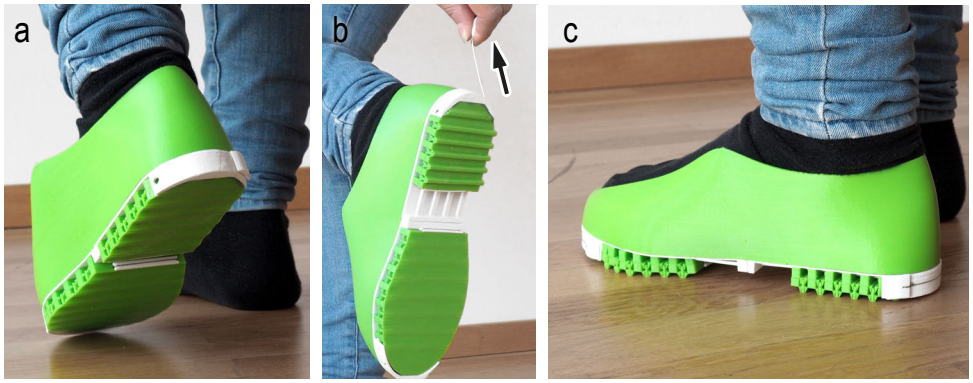

Figure 6.5: (a) This shoe sole is flat by default. (b) The user transforms it into a treaded sole it by pulling a string, e.g., when it starts snowing. (c) Note that the sole is functional and robust enough to walk on.

Exploring texture designs quickly. One of the qualities of metamaterial textures is that they can be continuously actuated to different levels (Figure 6.6a-c). This is useful when trying to explore how "strong" a texture should be during rapid prototyping. To test for the ergonomics of this bicycle grip, designers can prevent cells from folding locally. Figure 6.6d shows that sliding spacers into the material causes those cells to resist the compression. Designers can then again explore if the design feels right (Figure 6.6e).

Figure 6.6: (a) Designers fabricate one single bicycle grip that they (b) actuate continuously to different levels, (c) to feel the tactile qualities during rapid prototyping. (d) By inserting spacers after fabrication that (e) deactivates selected rows allows them to further investigate the grip's ergonomics.

Metamaterial textures allow product designers to quickly iterate through multiple textures in one 3D print only, instead of fabricating many prototypes, which is slow. In fact, designers and researchers agree that tactile designs "need to be felt early and often" [144]. Similar approaches exist for quickly iterating over rigid 3D shapes [98], we extend this idea to texture prototyping.

6.2 CONTRIBUTIONS

Our main contribution is the concept of embedding *multiple* dynamic textures into one 3D printed object using metamaterials. Such textured objects allow for continuously transitioning between magnitudes of the texture. Furthermore, they allow users to define the sequence in which the cells fold, which enables transitioning between multiple integrated textures after fabrication.

Our textures are integrated in the object at printing time. We contribute parameterized metamaterial cells that function using a single material and that enable a range of textures by only varying the cell geometry.

To assist users and researchers in designing new textures using metamaterials, we contribute an interactive editor that features a fast preview of the texture transformation.

6.3 DESIGN SPACE OF METAMATERIAL TEXTURES

To summarize the capabilities of metamaterial textures, we characterize their design space, as illustrated in Figure 6.7. The resulting design space consists of six dimensions:

Single cell primitives. We identified three geometry classes that can be parameterized to create a range of shapes for the tactile bump on a single cell: simple straight ridges (which can create box-like cells or rounded cells), diagonal (zigzag, spiky, etc.) or diamond shaped.

Composition. Single cell primitives can be composed to form a texture by (1) *uniformly tiling* the same cells one next to the other. A more expressive composition can be achieved by (2) chaining cells of the same type *row-wise* (such as in our door handle example). Lastly, it is also possible to

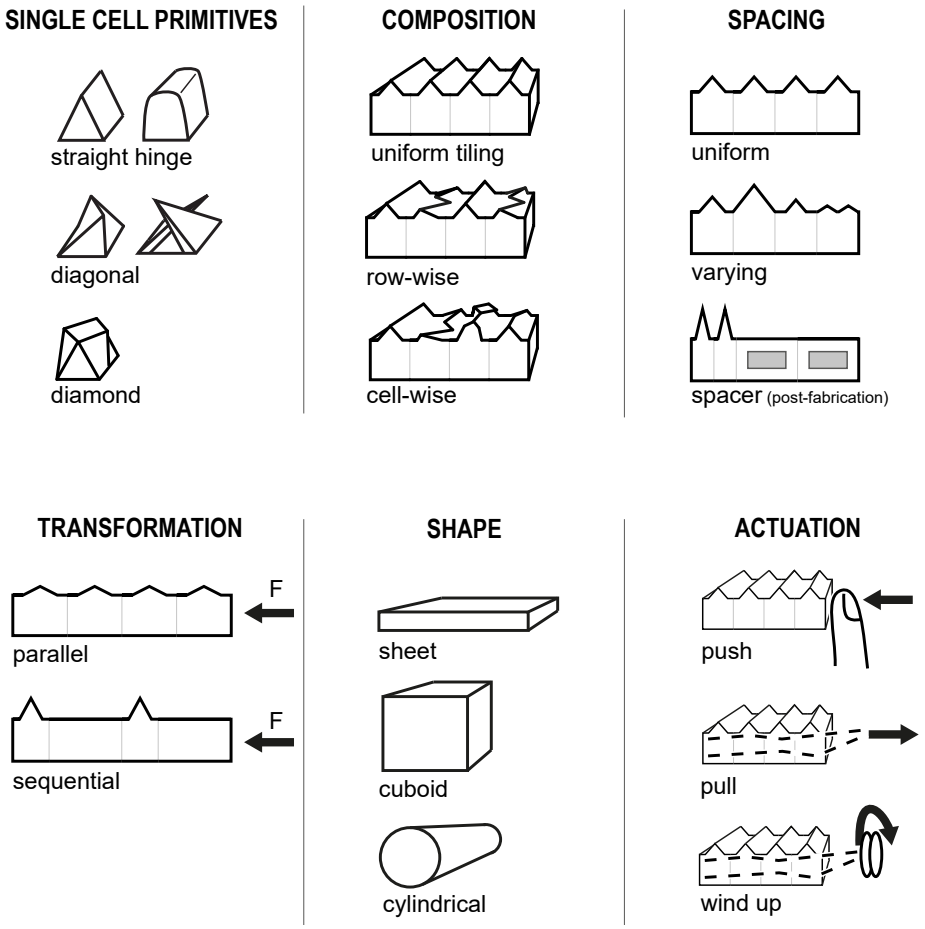

Figure 6.7: Our six-dimensional design space describing metamaterial textures.

(3) compose the texture from a *cell-wise* arrangement, the only restriction being that the positions of the folding hinge must join continuously from one cell to the next in the row.

Spacing. We identified three possible variations for how to define the space between tactile bumps. The simplest configuration available to designers is to spread out their bumps at equally distant points by *uniformly* spacing cells. To *vary* the spacing between tactile bumps, designers can choose cell parameters which increase or decrease the spacing and chain these cells. A complete explanation of the cell parameters is given in Sec-

tion 6.4.2. Lastly, the distance between bumps can be increased by inserting *spacers* into the material after it was fabricated, which prevents the selected cells from folding.

Transformation. Designers can define how the transformation between textures will be performed: the texture can fold in parallel, sequentially, or a combination of these. A parallel transformation implies that all cells fold at once (e.g., the bike grip example). A sequential transformation causes cells to fold subsequently as the force increases, as illustrated by the door handle in Figure 6.1. This is detailed in Section 6.4.3.

Shape. We identified three shape classes that our metamaterial textures can cover. The simplest shape that metamaterial textures can be applied to is a planar sheet. It can also be applied to cylindrical shapes (e.g., bike grip) or on the outside of cuboid shapes, e.g., the door handle example.

Actuation. Since our textures are actuated by global compression, we see three actuation possibilities for our resulting textures. The simplest form of actuating metamaterial textures is by pushing them manually to compress. Alternatively, designers can run strings through the material that are either pulled or wound up. While in this work we explore the idea of having no electronics in our materials and use human actuation for all three actuation classes, using motors or other automated actuators is certainly possible.

6.4 IMPLEMENTING TEXTURES BASED ON CELLS

Our materials consist of cells on a regular grid. In the following, we describe the design of the cells. In order to create different textures, we describe the cell parameters that can be varied by designers and their effects. For simplicity, we first describe the mechanics of the fold cell at the example of creating a simple bumpy structure. Then, we focus on the top of the cell and how to create more complex texture geometries.

Our prototypes were printed using *NinjaFlex*, a rubber-like filament, on an *Ultimaker 2+*, which is a consumer-level fused deposition modeling (FDM) 3D printer. The fold cells have a width of 15 mm and a height and depth of 7.5 mm.

6.4.1 Geometry of the fold cell

Figure 6.8 illustrates the structure of our unitary cell, the fold cell. It consists of two walls that are connected to four members by living hinges, i.e., thin parts that, due to their reduced stiffness, can flex.

Compressing the cell causes the internal four-bars to shear in a pre-defined direction, here it shears upwards. The shearing direction is encoded into the triangular shape of the members (Figure 6.8c): having the thicker part of the members towards the walls prevents them from tilting down as they would collide with the walls; reversing the triangular members would result in a downwards fold.

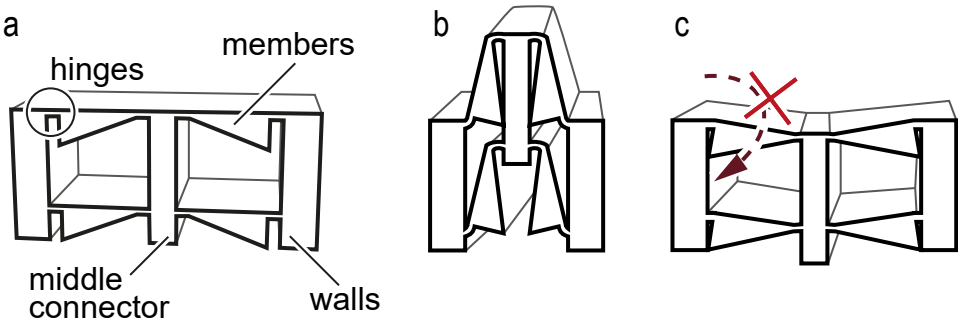

Figure 6.8: The fold cell consists of parametrizable walls, hinges and members that enable a stable fold to transform the material from straight to textured (here corrugated).

6.4.2 Amplitude and frequency of a texture

The height of the protruding texture is defined by the length of the fold cell's members. As shown in Figure 6.9a, longer members allow the resulting texture to pop out more from the 3D object. We call this the amplitude of the texture.

We found the relationship of maximum amplitude and member length to be described as:

$$\text{amplitude}_{max} = \sqrt{\left(\frac{c}{2} - w - m\right)^2 - t^2}$$

where c denotes the cell width, w the wall thickness, m the middle connector thickness, and t the member thickness.

While an increased wall thickness decreases the amplitude, it simultaneously decreases the spatial *frequency* of the texture. Figure 6.9b illustrates how increasing the wall thickness of the cells separates neighboring cells further apart and thus reduces the resulting texture's frequency. To achieve a high frequency with low amplitude, we can split the cell in two, as shown in Figure 6.9c.

The same parameters that influence the amplitude also define the maximum compression ratio of the texture. For example, cells with thick walls can be compressed by a smaller extend than cells with thin walls. The compression ratio per cell is therefore modeled as

$$\text{compression}_{cell} = \frac{2w + 2t + m}{c}.$$

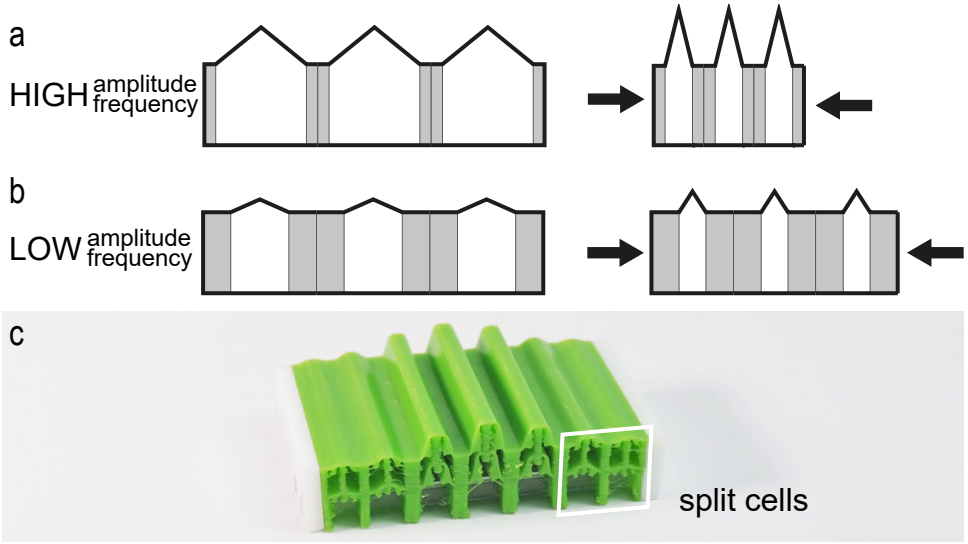

Figure 6.9: (a-b) The length of the fold cell's members defines the texture's amplitude, i.e., the height of each bump. (c) The amplitude can be varied across the material and cells can be split to create a higher frequency.

6.4.3 Force-dependent textures

The thickness of the hinges defines how much force is required to fold them outwards. Figure 6.10 illustrates that because thicker hinges require more force to be deformed than thin hinges (in fact, the required force is to the power of three [41]), they also *fold later* as they are subjected to constant compression.

To embed multiple textures in one object, as demonstrated in our door handle example shown in Figure 6.1, designers specify a smaller hinge thickness for the cells that will pop up first (cf. the ridges in the door handle) and a larger hinge thickness for the cells that will fold later (cf. the spiky cells). This allows designers to potentially make every row dependent on different amounts of force to create animated textures on 3D objects.

While a more exhaustive technical evaluation is planned for future work, we report a simple experiment to evaluate an example structure, featuring a row with four different fold cells. Each cell in Figure 6.11 has a unique

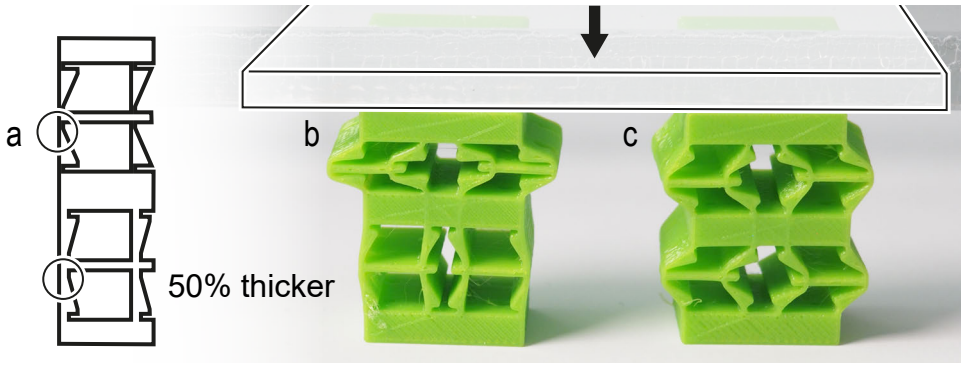

Figure 6.10: (a) Here, we demonstrate how our approach embeds multiple textures in one surface by varying the thickness of the hinges. (b) The upper cell's hinges are thinner and thus fold before (i.e., under less force) than the lower cell's hinges, which are 50% thicker. (c) If the hinge thickness is uniform across cells, they fold simultaneously under the same load.

hinge thickness; from left to right $0.6\,\text{mm}$, $0.8\,\text{mm}$, $0.4\,\text{mm}$, $1.0\,\text{mm}$. Figure 6.11a shows the third cell folding upwards as the whole row is compressed while connected to a force gauge. As expected, we confirmed that the folding order of our printed cells is indeed dictated by their increasing hinge thickness: $0.4\,\text{mm}$ folds at $3.0\,\text{N}$, $0.6\,\text{mm}$ folds at $3.65\,\text{N}$, $0.8\,\text{mm}$ folds at $6.4\,\text{N}$ and $1.0\,\text{mm}$ folds at $8.15\,\text{N}$.

Figure 6.11: Our force test confirms that by varying the hinge thicknesses, we can control when the cell folds up.

6.4.4 Single cell primitives

We now describe how to create bumps with more expressive shapes. So far, we have seen only textures created by bumps with a triangular protrusion. Now, we demonstrate that by simply altering the hinges on the top of the single cells, we can achieve more interesting results.

Triangular, squared and rounded texture bumps

The simplest bump is just a straight fold, which resembles a small triangle (Figure 6.12a). Next, we design a box-like texture bump by specifying equal widths for all members (Figure 6.12b). We can also create round bumps (Figure 6.12c) by making the hinge in the middle long and double its thickness (here to 0.8 mm).

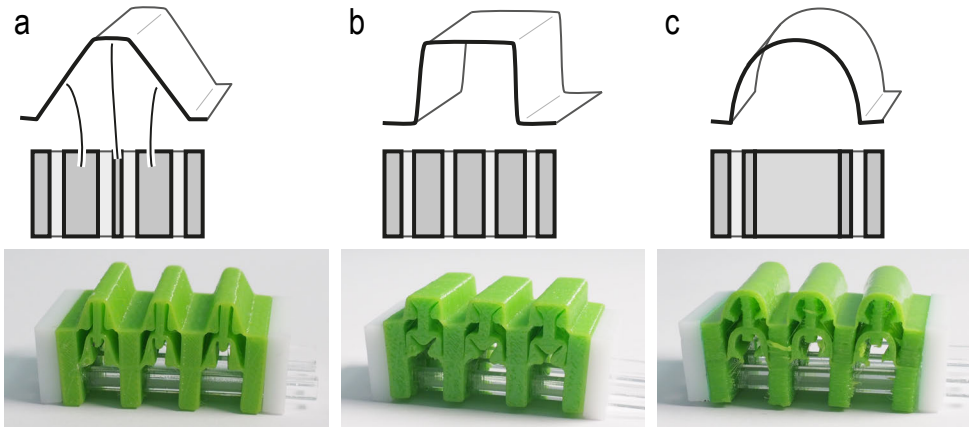

Figure 6.12: Variations of textures using a straight fold: (a) triangular folds, (b) box-like or (c) round bumps.

Zigzag texture bumps

The simple straight bend that we demonstrated can be transformed to create more elaborate textures. We do so by offsetting the hinge positions on both edges of the cell, as illustrated by Figure 6.13a. This results in a diagonal folding up. The connection to the cell's middle connector is only possible at a small part in the middle of the hinge, which requires the

connector to be tapered in the z-axis (Figure 6.13b). Despite the connection being very small, the structure works as before because the lower members push the cell up.

Note that the diagonal hinge also reduces the maximum amplitude and the compression ratio. This is because the offset of the hinge created members of different lengths, i.e., a shorter and a longer member. The cell can now only fold up to the extent of the shorter member.

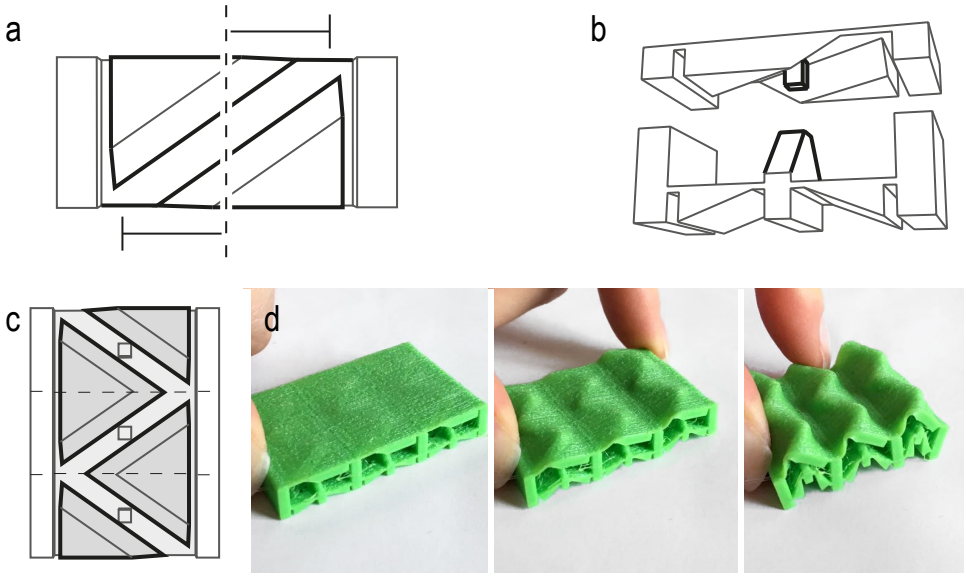

Figure 6.13: (a) Adding horizontal offsets to the hinge creates diagonal folds. (b) The connector width to the lower cell structure needs to be decreased gradually to connect to the cell top. (c) Composing multiple diagonal fold cells creates zigzag patterns that (d) can be varied in magnitude.

Spiky texture bumps

To create the spiky texture from our door handle example, we take the zigzag pattern from Figure 6.13a and remove the material from the hinge, only leaving a thin connection in the middle as shown in Figure 6.14a. This results a malleable spiky texture as the triangles fold upwards but are disconnected from their walls.

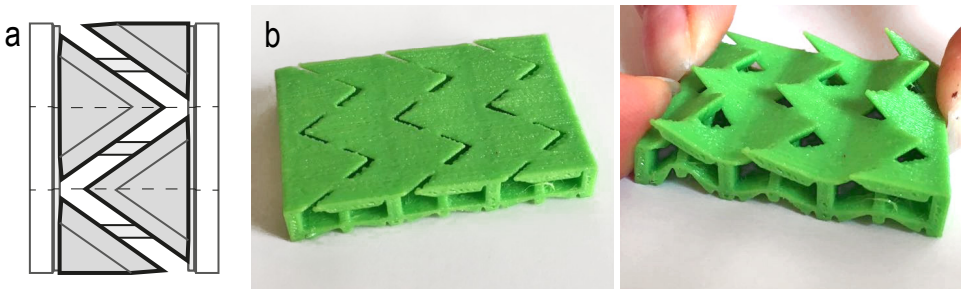

Figure 6.14: (a) Leaving gaps between the members of the cell top (b) allows for creating spiky textures.

Diamond texture bumps

The diamond bump can be created by changing the location where the horizontal offset is effective in the vertical axis. Figure 6.15 shows how this allows designers to create a Y-shaped pattern, which has the effect of flattening out in the middle. By mirroring this pattern to the neighbor cell, it creates a diamond-shaped texture. It is also possible to achieve the diamond pattern on a single cell, by varying the vertical offsets on both sides (Figure 6.15b).

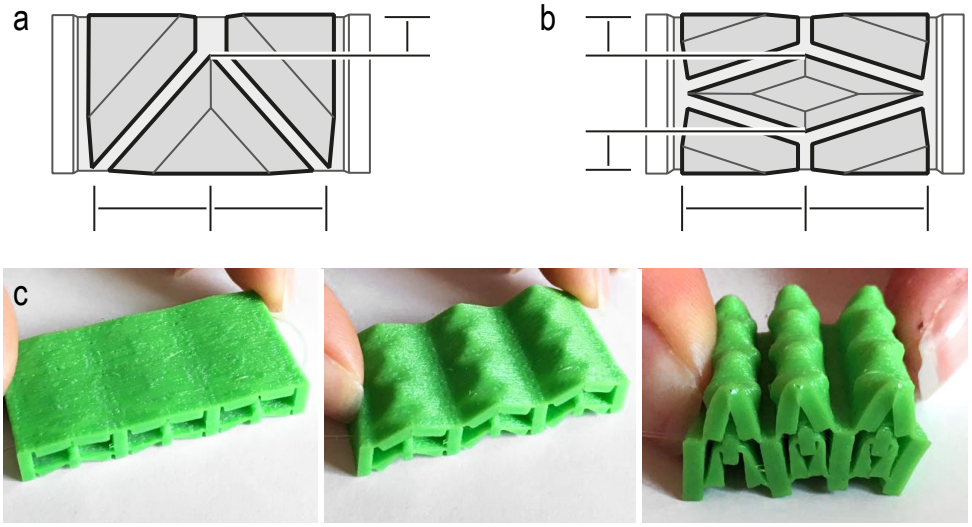

Figure 6.15: (a-b) Offsetting the hinge vertically and horizontally creates (c) diamond shaped textures, which flatten in the middle and thus create a different tactile feel.

6.5 EDITING METAMATERIAL TEXTURES

Figure 6.16 shows the interactive editor we built to assist designers in creating textures based on metamaterials. Here, we see a user creating a block that when pushed displays a zigzag texture on the top.

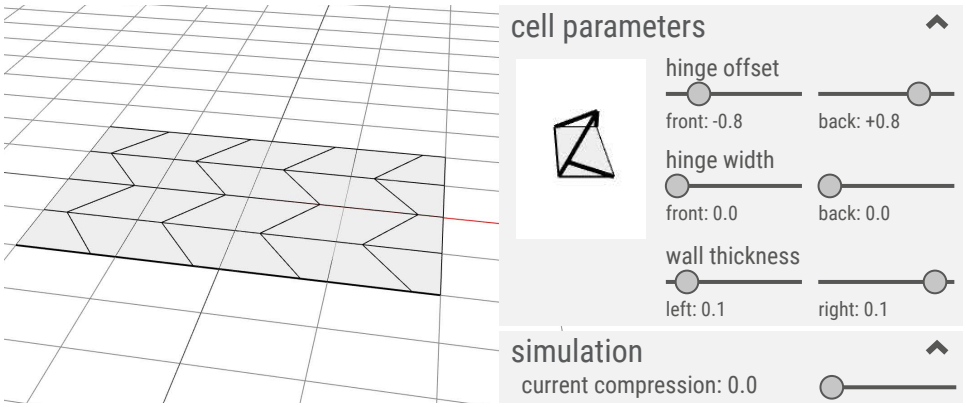

Figure 6.16: Our editor assists users in creating metamaterial textures. Users adjust the cell geometry using sliders and lay the cells out on the grid.

6.5.1 User interaction

In the interactive editor, we exploit the fact that the fold cell is fully parameterized. Users can set all parameters (Figure 6.16, *right*), for example hinge offset, width and wall thickness, simply by dragging the individual sliders. An interactive preview of a cell in its actuated state (Figure 6.16, *left from the parameters*) allows users to see how the cell will look like once placed and actuated. After setting all parameters, users can arrange the cells on a regular grid to create textures from metamaterials.

Our editor runs in a browser. It is based on the editor for metamaterial mechanisms, which is built using node.js (a Javascript runtime framework) and utilizes the three.js library for rendering.

6.5.2 Previewing textures by means of simulation

The editor offers a preview of the resulting texture based on the user's current metamaterial design. They can interactively preview the different textures, which result from different compression rates by simply dragging the slider (Figure 6.17) that specifies the current compression. The deformation of the textures is rendered in real-time.

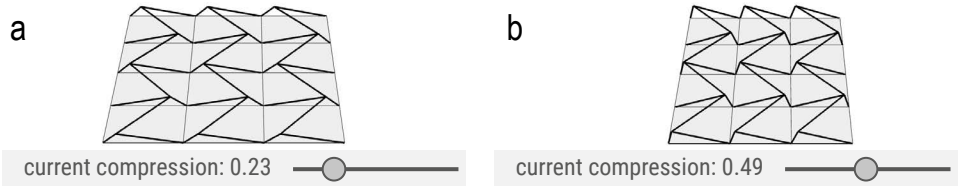

Figure 6.17: Users interactively preview their textures by dragging the slider that sets the simulated compression.

The simple kinematic simulation that we implemented in our editor allows for previewing, in real-time, how the designed textures fold up when the material is compressed via a GUI slider. Our simulation calculates only geometric transformations by simple propagation, i.e., as a cell compresses, its members move, in turn, these members move the neighboring cell's members, and so forth.

Note that at this stage our simulation does not take material properties into account. We opted for this approach as it allows for an interactive simulation (at 30 fps) compared to, e.g., finite element analysis, which is computationally more expensive and therefore slower, but more accurate.

6.5.3 Generating a printable file

To simplify the process of creating a texture based on metamaterials, our editor only requires users to choose how the surface of the cell should fold to form their texture. In fact, the remainder of the 3D object, e.g., its internal structure beyond the top layer, is generated by our editor when users export the final texture.

We derive the parameters for the remainder cell geometry from the user defined hinge positions. The wall thickness is derived from the hinge offsets. For a diagonal hinge, we gradually decrease the thickness of the middle connector towards the top so that it connects only to the center of the texture geometry with minimum widths of $0.3\,\text{mm}$. After having created the cell body for the fold cell, we repeat this structure until the user-defined volume is filled. Finally, this geometry is exported into a 3D printable .stl file.

6.5.4 Limitations of the interactive editor

Our editor only allows creating objects with simple geometries, i.e., planar, curved, and cylindrical. Our current kinematic simulation enables real-time interaction at the expense of offering more simplistic results. These, however, preview all the textures we demonstrated correctly.

6.6 DISCUSSION

In the following we present a discussion of our prototype centered on its limitations and potential implications (illustrated by Figure 6.18).

6.6.1 Limitations

While we see this work as the first step towards creating textured objects using metamaterials, it certainly has a number of limitations. The most evident one is that our approach works only for objects in which exact dimensions are not critical. The exact length of a door handle, for example, is not critical for its functionality. In fact, our approach always generates a change in the object's overall shape. Secondly, our current approach is limited to actuation using a global force pulled along one dimension (e.g., when the user pulls the wires to configure the shoe sole). In the future, we want to investigate dynamic textures that can be actuated in two and three dimensions. Furthermore, we currently use external materials, such

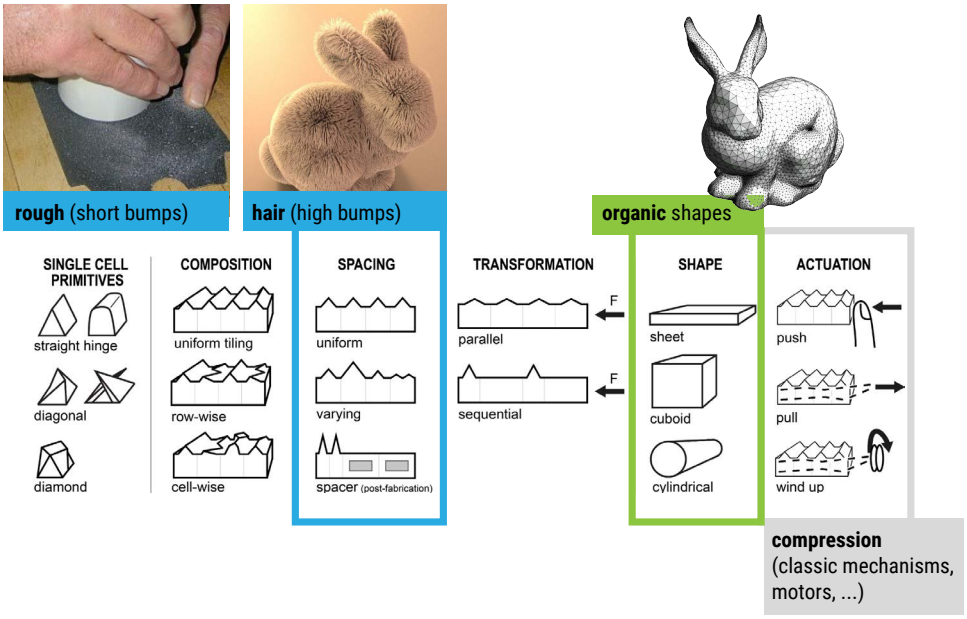

Figure 6.18: Metamaterial textures can be actuated automatically (gray). In the future, our proposed cells should be extended to organic shapes (green). Using nano-scale printing could allow sophisticated textures such as fur or adaptable sandpaper (blue).

as the strings that are pulled, to actuate the metamaterial textures. In the future, we want to integrate the actuation with the metamaterial itself so that it can be fabricated in one piece. Lastly, our demonstration objects are limited in that the textures pop out on planes or cylindrical shapes only. Ideally, textures would be integrated in arbitrary organic shapes.

6.6.2 Alternative forms of actuation

Since in this work, we explore the idea of materials that exhibit textures by means of metamaterials, we opted to actuate our textures in the simplest way (e.g., pushing, strings, etc.), so that no electronic components are required (such as motors, batteries, microcontrollers, etc.). However, alternative actuation mechanisms can certainly be used, e.g., motors, that will allow programmatic real-time behavior without users' actions. For instance, one can envision how the door handle changes texture automatically by being digitally connected to the user's calendar.

6.6.3 Outlook for nano-scale metamaterial textures

The scale of our current textures was dictated by the resolution of our consumer-grade desktop printer. However, using state of the art high-resolution printers (e.g., nanoscribe[1]) it might be possible to uncover new opportunities using our approach. Firstly, by scaling down a design similar to that of Figure 6.9a, i.e., with long members that protrude a lot outside the object, it might be possible to use our approach to generate furry textures [70, 104] that are transformable.

Conversely, using high-resolution printers to fabricate cells with short members would yield a texture made from micro bumps that would feel "rough" to the user. This could also be employed as a prototyping tool to alter an object's friction when sliding on a surface or create adaptable sandpaper (Figure 6.18, marked in blue).

6.7 CONCLUSIONS

We proposed an approach that leverages metamaterials to create transformable textures on 3D printed objects. We demonstrated the benefits of our approach in three objects and provided an interactive editor to allow researchers and users to create novel textures.

We see metamaterial textures as a first step to integrate transforming textures into 3D printed objects. In the long run, we think such an approach might be relevant to disseminate more expressive haptics in everyday objects. We hope this opens new dialogs between UX and product designers and results in novel everyday objects with multiple pre-integrated textures that can be activated by the end user.

[1] http://www.nanoscribe.de/en/

7

Conclusion

In this chapter, we expand the insights of the individual projects and draw conclusions on the bigger picture—the concept of metamaterial devices. We discuss how this work might impact technology and summarize our main contribution before we close by discussing short-term future challenges and outlining long-term future directions of metamaterial devices.

7.1 CONTRIBUTION

With this thesis, we extend the research fields of metamaterials, digital fabrication, and 3D modelling tools by contributing a novel approach to creating interactive devices that do not rely on electronics, but their interactivity is defined by their material structure. Such metamaterial structures

are a new genre that is of higher complexity and that exploits more degrees of freedom than previous work in this field, and that allow metamaterials to tackle problems they have traditionally not been able to address. Since the entire device is defined by the architected cell structure of the material it consists of, we like to think that we blur the boundaries between materials and devices. Therefore, we see the main promise of this work in that it allows us to achieve a deeper integration between the structural and the mechanical functions of materials.

On a more tangible level, we contribute different cell designs that play together to transform forces, perform simple computation, or change their outside structure. Moreover, we presented software tools that assist users in creating such devices. We also dug deeper into the structural constraints that cells exert when connected on a grid. This understanding enabled a computational design tool that generates metamaterial mechanisms for the user—and is a promising approach for the future.

7.2 POTENTIAL IMPACT: TWO VIEWS

Here, we would like to discuss two aspects that we find particularly interesting about our work. One is that metamaterial devices do not simply mimic traditional devices, but leverage the benefits of 3D printing. Secondly, we believe that software tools and assembly-free designs combined with the availability of consumer-grade 3D printers enables many more (lay) people to help invent future technology.

Not mimicking traditional devices

While 3D printing is still on the rise [147], for a technology that can arrange matter freely in space it is still not very wide spread. We think that one of the reasons is that users at home as well as in industry often just mimic traditional devices. 3D printing traditionally shaped mechanical parts (e.g,

axles, sockets, etc.) is of course viable. Multiple parts can even be printed in one process [21]. We argue, however, that this does not make good use of the technological advancement that 3D printers allow for.

With new technology, we should rethink how we design items to really leverage its benefits. The work we presented in this thesis does rethink the design of devices. While the application examples that illustrated the potential use of such metamaterial devices were rather traditional, the approach of using cell structures to achieve them is certainly not. We argue that our metamaterial-based approach leverages the benefits of 3D printing well, because the cell structures act like a framework, which in many cases eliminates the need for support structures[1].

Considering the fabrication technology in the design process allows expanding the devices from, e.g., only the door latch mechanism to fabricating the complete door including the mechanism. Going even further, we can imagine printing a house where the door with the latch mechanism is printed with the walls, connecting them with metamaterial hinges. The walls can also consist of structures acting as acoustic dampers from outside noise or thermal insulators. We believe that such a future is possible if the technology and the design fit together.

Everyone!

While creating devices and mechanisms has so far remained a privilege of experts only, in this thesis, we provide concepts, mechanical cell designs and interactive editing tools that enable everyone to design and fabricate new devices.

Creating new devices typically consists of two phases, (1) the design phase and (2) the craft phase. The design phase involves the design of individual parts and their positioning with respect to each other in order to fulfil

[1] Recently, researchers even started focusing specifically on 3D-printable structures with directional stiffness control [76].

a function. The craft phase is concerned with the manufacturing and assembly of those parts. However, both phases usually require substantial knowledge and expertise with regards to fit, play, friction, etc.

The works presented in this thesis ease both phases in order to enable experts and novices to create new devices. Our metamaterial devices require no assembly and because the manufacturing is offloaded to the 3D printer, users are being shielded from the craft phase. We support users in the design phase by providing interactive computational tools that help creating the geometry of the cell structures, which define the function of the device. On a more abstract level, we believe that allowing even novice users to participate in the innovation of new devices will accelerate technological advancement. Potentially billions of people could invent devices and help push the boundaries of technology as compared to only a few experts and researchers.

7.3 OPEN CHALLENGES

To expand the capabilities of metamaterial devices further, we see three main challenges that need to be solved. In this section, we refer to challenges that can be solved in the near future (~2 years).

3D printing technology & materials

Intricate microstructures are still tricky to print. While lithography-based technologies, which fuse liquid or powder-based materials with light (SLA or SLS), work well for such structures as there is mostly no need for support structures, the material choice is very limited. Many materials for these 3D printers are soft, but not elastic, or tear too easily. We see, however, first sparks of interest in industry, which promises for a push in 3D printing technology for faster and cheaper manufacturing possibilities. For example, Adidas just released mass-customized sneakers with microstructured

soles. They partnered with the 3D printing company Carbon[2] for fabrication. Carbon is actually the first resin curing printing technology to offer a truly elastic material, which is very promising.

Sensing

While in this work the input for metamaterial devices was only mechanical motion input by humans, we think that sensing capabilities are an interesting challenge. An interesting research perspective would be to investigate sensor cells to detect, e.g., light, motion, magnetic fields, moisture, gas, etc. The sensed value needs to be converted into a mechanical input to the metamaterial device. Moreover, sensor cells that can detect their state would enable closed-loop metamaterial devices.

Computer-controlled actuation

We want to extend this view to automatic actuation. Metamaterial cells, for example, could be fabricated from shape memory polymers. Another idea would be to coat them with conductive material such that they can be actuated by an electromagnetic field. Such approaches could be suitable to reset the bistable springs in our digital metamaterials or automatically change the state of metamaterial textures.

7.4 OUTLOOK

We think that the classic concept of metamaterials is an extremely interesting research perspective with much potential to advance technology. Being able to save material, create better insulators, reduce weight, redirect light, etc. will enable unforseen technological advances. As a vision, which is 10+ years further out, we think that pushing such high-functioning metamaterials further by combining them with our concept of letting them perform mechanical functions will push these boundaries even more. A future

[2] https://www.carbon3d.com/stories/rethinking-foam-carbons-lattice-innovation/

metamaterial might not only be lightweight, but it can be lightweight *and* mechanically actuate a part. Future metamaterials might not only perform thermal control on space crafts, but steer it by changing their resistance at the same time—the rate of which is computed within the material itself.

So far, most metamaterial structures are tiling the same cell. Combining a cell structure with different parameters within a material is only a recent development. We think that combining topologically different cells is an important step to make in order to create such advanced metamaterials, as envisioned above. Ultimately, such *heterogeneous mechanical metamaterials*—as we would like to call them—enabled our metamaterial devices.

A missing piece for making such advanced metamaterial devices a reality are efficient ways for their exploration. We believe that to explore such devices, we need to build computational tools that allow researchers to explore their novel cell designs quickly. A future software tool should allow users to upload novel structures and combine them on a grid. The software needs to automatically deduce the constraints of the structures to adapt them or generate transition structures. Such tools would foster research of metamaterials with all their potential drastically and accelerate technological advancement.

References

[1] Robert F. Almgren. An isotropic three-dimensional structure with Poisson's ratio = -1. *Journal of Elasticity*, 15(4):427–430, 1985. ISSN 0374-3535. doi:10.1007/BF00042531. URL http://link.springer.com/10.1007/BF00042531.

[2] Juan Carlos Álvarez Elipe and Andrés Díaz Lantada. Comparative study of auxetic geometries by means of computer-aided design and engineering. *Smart Materials and Structures*, 21(10), oct 2012. ISSN 0964-1726. doi:10.1088/0964-1726/21/10/105004. URL http://stacks.iop.org/0964-1726/21/i=10/a=105004?key=crossref.c6f7c02eff89e04c196afb61933a2d7e.

[3] Byoungkwon An, Shuhei Miyashita, Michael T. Tolley, Daniel M. Aukes, Laura Meeker, Erik D. Demaine, Martin L. Demaine, Robert J. Wood, and Daniela Rus. An end-to-end approach to making self-folded 3D surface shapes by uniform heating. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1466–1473, 2014. ISBN 9781479936847. doi:10.1109/ICRA.2014.6907045.

[4] Byoungkwon An, Hsiang-yun Wu, Teng Zhang, Lining Yao, Ye Tao, Jianzhe Gu, Tingyu Cheng, Xiang Anthony Chen, Xiaoxiao Zhang, Wei Zhao, Youngwook Do, and Shigeo Takahashi. Thermorph: Democratizing 4D Printing of Self-Folding Materials and Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, New York, New York, USA, 2018. ACM Press. ISBN 9781450356206. doi:10.1145/3173574.3173834. URL http://dl.acm.org/citation.cfm?doid=3173574.3173834.

[5] Autodesk. Autodesk Netfabb. last accessed on 2018-06-19, . URL https://www.autodesk.com/products/netfabb/overview.

[6] Autodesk. Autodesk Within Medical. last accessed on 2018-06-19, . URL https://www.autodesk.com/products/within-medical/overview.

[7] Sahab Babaee, Jongmin Shim, James C. Weaver, Elizabeth R. Chen, Nikita Patel, and Katia Bertoldi. 3D soft metamaterials with negative poisson's ratio. *Advanced Materials*, 25(36):5044–5049, 2013. ISSN 09359648. doi:10.1002/adma.201301986.

[8] Sahab Babaee, J. T. B. Overvelde, Elizabeth R. Chen, Vincent Tournat, and Katia Bertoldi. Reconfigurable origami-inspired acoustic waveguides. *Science Advances*, 2(11), nov 2016. ISSN 2375-2548. doi:10.1126/sciadv.1601019. URL http://advances.sciencemag.org/cgi/doi/10.1126/sciadv.1601019.

[9] Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. Spin-it: Optimizing Moment of Inertia for Spinnable Objects. *ACM Trans. Graph.*, 33(4):96:1–96:10, 2014. ISSN 0730-0301. doi:10.1145/2601097.2601157. URL http://doi.acm.org/10.1145/2601097.2601157.

[10] Moritz Bächer, Stelian Coros, and Bernhard Thomaszewski. LinkEdit: interactive linkage editing using symbolic kinematics. *ACM Transactions on Graphics*, 34(4):99:1–99:8, 2015. ISSN 07300301. doi:10.1145/2766985. URL http://dl.acm.org/citation.cfm?doid=2809654.2766985.

[11] Seok-hyung Bae, Ravin Balakrishnan, and Karan Singh. ILoveSketchh: As-Natural-As-Possible Sketching System for Creating 3D Curve Models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 151–160, 2008. ISBN 9781595939753. doi:10.1145/1449715.1449740. URL http://dl.acm.org/citation.cfm?id=1449715.1449740.

[12] Seok-hyung Bae, Ravin Balakrishnan, and Karan Singh. EverybodyLovesSketch : 3D Sketching for a Broader Audience. In *Proceedings of ACM Symposium on User Interface Software and Technology*, pages 59–68, 2009. ISBN 9781605587455. doi:http://doi.acm.org/10.1145/1622176.1622189. URL http://www.dgp.toronto.edu/{~}shbae/everybodylovessketch.htm.

[13] Patrick Baudisch and Stefanie Mueller. Personal Fabrication. *Foundations and Trends in Human-Computer Interaction*, 10(3-4):165–293, 2017. doi:10.1561/1100000055.

[14] Jasmin Beharic, Thomas Matthew Lucas, and Cindy K. Harnett. Analysis of a Compressed Bistable Buckled Beam on a Flexible Support. *Journal of Applied Mechanics*, 81(8):081011, jun 2014. ISSN 0021-8936. doi:10.1115/1.4027463. URL http://appliedmechanics.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4027463.

[15] Katia Bertoldi, Pedro M. Reis, Stephen Willshaw, and Tom Mullin. Negative poisson's ratio behavior induced by an elastic instability. *Advanced Materials*, 22(3):361–366, 2010. ISSN 09359648. doi:10.1002/adma.200901956.

[16] Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin van Hecke. Flexible mechanical metamaterials. *Nature Reviews Materials*, 2(11):17066, oct 2017. ISSN 2058-8437. doi:10.1038/natrevmats.2017.66. URL http://www.nature.com/articles/natrevmats201766.

[17] Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. Design and fabrication of materials with desired deformation behavior. *ACM Transactions on Graphics*, 29(4):1, 2010. ISSN 07300301. doi:10.1145/1778765.1778800. URL http://portal.acm.org/citation.cfm?doid=1778765.1778800.

[18] Tiemo Bückmann, Michael Thiel, Muamer Kadic, Robert Schittny, and Martin Wegener. An elasto-mechanical unfeelability cloak made of pentamode metamaterials. *Nature Communications*, 5(May):1–6, 2014. ISSN 20411723. doi:10.1038/ncomms5130.

[19] Tiemo Bückmann, Muamer Kadic, Robert Schittny, and Martin Wegener. Mechanical cloak design by direct lattice transformation. *Proceedings of the National Academy of Sciences*, 112(16):4930–4934, apr 2015. ISSN 0027-8424. doi:10.1073/pnas.1501240112. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1501240112.

[20] Jesse Burstyn, Nicholas Fellion, Paul Strohmeier, and Roel Vertegaal. PrintPut : Resistive and Capacitive Input Widgets for Interactive 3D Prints. In *Proceedings of Interact 2015*, pages 332–339. IFIP International Federation for Information Processing, 2015. ISBN 9783319227016. doi:10.1007/978-3-319-22701-6.

[21] Jacques Calì, Dan A. Calian, Cristina Amati, Rebecca Kleinberger, Anthony Steed, Jan Kautz, and Tim Weyrich. 3D-printing of non-assembly, articulated models. *ACM Transactions on Graphics*, 31(6): 1–8, 2012. ISSN 07300301. doi:10.1145/2366145.2366149. URL http://dl.acm.org/citation.cfm?doid=2366145.2366149.

[22] Paul Cazottes and Moustapha Hafez. Bistable Buckled Beam: Modeling of Actuating Force and Experimental Validations. *Journal of Mechanical Design*, 131(10):101001, 2009. ISSN 10500472. doi:10.1115/1.3179003. URL http://dx.doi.org/10.1115/1.3179003.

[23] Desai Chen, David I. W. Levin, Piotr Didyk, Pitchaya Sitthi-Amorn, and Wojciech Matusik. Spec2Fab: A Reducer-Tuner Model for Translating Specifications to 3D Prints. *ACM Transactions on Graphics*, 32:1, 2013. ISSN 07300301. doi:10.1145/2461912.2461994. URL http://dl.acm.org/citation.cfm?id=2461994{%}5Cnhttp://dl.acm.org/citation.cfm?doid=2461912.2461994.

[24] Desai Chen, Mélina Skouras, Bo Zhu, and Wojciech Matusik. Computational discovery of extremal microstructure families. *Science Advances*, 4(1), jan 2018. ISSN 2375-2548. doi:10.1126/sciadv.aao7005. URL http://advances.sciencemag.org/lookup/doi/10.1126/sciadv.aao7005.

[25] Tian Chen, Jochen Mueller, and Kristina Shea. Integrated Design and Simulation of Tunable, Multi-State Structures Fabricated Monolithically with Multi-Material 3D Printing. *Scientific Reports*, 7:1–8, 2017. ISSN 20452322. doi:10.1038/srep45671. URL http://dx.doi.org/10.1038/srep45671.

[26] Xiang Anthony Chen, Y. Tao, G. Wang, R. Kang, Tovi Grossman, Stelian Coros, and Scott E. Hudson. Forte: User-driven generative design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, volume 2018-April, 2018. ISBN 9781450356206. doi:10.1145/3173574.3174070.

[27] Johan Christensen, Muamer Kadic, Oliver Kraft, and Martin Wegener. Vibrant times for mechanical metamaterials. *MRS Communications*, 5(03):453–462, 2015. ISSN 2159-6859. doi:10.1557/mrc.2015.51. URL http://www.journals.cambridge.org/abstract{_}S2159685915000518.

[28] Chen Chu, Greg Graf, and David W. Rosen. Design for additive manufacturing of cellular structures. *Computer-Aided Design and Applications*, 5(5):686–696, 2008. ISSN 16864360. doi:10.3722/cadaps.2008.686-696.

[29] Dixon M. Correa, Timothy Klatt, Sergio Cortes, Michael Haberman, Desiderio Kovar, and Carolyn Seepersad. Negative stiffness honeycombs for recoverable shock isolation. *Rapid Prototyping Journal*, 21(2):193–200, 2015. ISSN 1355-2546. doi:10.1108/RPJ-12-2014-0182. URL http://www.emeraldinsight.com/doi/abs/10.1108/RPJ-12-2014-0182.

[30] Dixon M. Correa, Carolyn Seepersad, and Michael Haberman. Mechanical design of negative stiffness honeycomb materials. *Integrating Materials and Manufacturing Innovation*, 4(10), 2015. ISSN 2193-9764. doi:10.1186/s40192-015-0038-8. URL http://dx.doi.org/10.1186/s40192-015-0038-8.

[31] Corentin Coulais, Eial Teomy, Koen de Reus, Yair Shokef, and Martin van Hecke. Combinatorial design of textured mechanical metamaterials. *Nature*, 535:529–532, 2016. ISSN 0028-0836. doi:10.1038/nature18960. URL http://dx.doi.org/10.1038/nature18960.

[32] E. L. Doubrovski, E.Y. Tsai, Daniel Dikovsky, Jo Geraedts, H. Herr, and Neri Oxman. Voxel-based fabrication through material property mapping: A design method for bitmap printing. *Computer Aided Design*, 60:3–13, 2015. ISSN 00104485. doi:10.1016/j.cad.2014.05.010. URL http://dx.doi.org/10.1016/j.cad.2014.05.010.

[33] Xi-qiao Feng. Wrinkling of a stiff film resting on a fiber-filled soft substrate and its potential application as tunable metamaterials. *Extreme Mechanics Letters*, 11:121–127, 2017. ISSN 2352-4316. doi:10.1016/j.eml.2016.12.002. URL http://dx.doi.org/10.1016/j.eml.2016.12.002.

[34] Evgueni T. Filipov, Tomohiro Tachi, and Glaucio H. Paulino. Origami tubes assembled into stiff , yet reconfigurable structures and metamaterials. *Proceedings of the National Academy of Sciences (PNAS)*, 112(40):12321–12326, 2015. doi:10.1073/pnas.1509465112.

[35] Tobias Frenzel, Claudio Findeisen, Muamer Kadic, Peter Gumbsch, and Martin Wegener. Tailored Buckling Microlattices as Reusable Light-Weight Shock Absorbers. *Advanced Materials*, 28(28):5865–5870, jul 2016. ISSN 09359648. doi:10.1002/adma.201600610. URL http://doi.wiley.com/10.1002/adma.201600610.

[36] Joshua B. Gafford, Samuel B. Kesner, Alperen Degirmenci, Robert J. Wood, Robert D. Howe, and Conor J. Walsh. A monolithic approach to fabricating low-cost, millimeter-scale multi-axis force sensors for minimally-invasive surgery. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1419–1425, 2014. ISBN 978-1-4799-3685-4. doi:10.1109/ICRA.2014.6907038.

[37] Madeline Gannon, Tovi Grossman, and George Fitzmaurice. Tactum : A Skin-Centric Approach to Digital Design and Fabrication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10, 2015. doi:10.1145/2702123.2702581.

[38] Qi Ge, Conner K Dunn, H. Jerry Qi, and Martin L. Dunn. Active origami by 4D printing. *Smart Materials and Structures*, 23(9), sep 2014. ISSN 0964-1726. doi:10.1088/0964-1726/23/9/094007. URL http://stacks.iop.org/0964-1726/23/i=9/a=094007?key=crossref.b77e5da2653ac838509a57efc6aad683.

[39] Ren Geryak and Vladimir V Tsukruk. Reconfigurable and actuating structures from soft materials. *Soft Matter*, 10:1246–1263, 2014. ISSN 17446848. doi:10.1039/c3sm51768c.

[40] J. N. Grima and K. E. Evans. Auxetic behavior from rotating squares. *Journal of Materials Science Letters*, 19(17):1563–1565, 2000. ISSN 02618028. doi:10.1023/A:1006781224002. URL https://link.springer.com/article/10.1023{%}2FA{%}3A1006781224002.

[41] Dietmar Gross, Werner Hauger, Jörg Schröder, and Wolfgang A. Wall. *Technische Mechanik 2*. Springer-Lehrbuch. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-40965-3. doi:10.1007/978-3-642-40966-0. URL http://link.springer.com/10.1007/978-3-642-40966-0.

[42] Ruslan Guseinov, Eder Miguel, and Bernd Bickel. Curve-Ups. *ACM Transactions on Graphics*, 36(4), jul 2017. ISSN 07300301. doi:10.1145/3072959.3073709. URL http://dl.acm.org/citation.cfm?doid=3072959.3073709.

[43] Mark Guttag and Mary C. Boyce. Locally and dynamically controllable surface topography through the use of particle-enhanced soft composites. *Advanced Functional Materials*, 25:3641–3647, 2015. ISSN 16163028. doi:10.1002/adfm.201501035.

[44] Babak Haghpanah, Hamid Ebrahimi, Davood Mousanezhad, Jonathan Hopkins, and Ashkan Vaziri. Programmable Elastic Metamaterials. *Advanced Engineering Materials*, 18(4):643–649, apr 2016. ISSN 14381656. doi:10.1002/adem.201500295. URL http://doi.wiley.com/10.1002/adem.201500295.

[45] Babak Haghpanah, Ladan Salari-Sharif, Peyman Pourrajab, Jonathan Hopkins, and Lorenzo Valdevit. Multistable Shape-Reconfigurable Architected Materials. *Advanced Materials*, 28:7915–7920, 2016. ISSN 15214095. doi:10.1002/adma.201670255.

[46] Ryan L. Harne and K. W. Wang. A review of the recent research on vibration energy harvesting via bistable systems. *Smart Materials and Structures*, 22(2), 2013. ISSN 09641726. doi:10.1088/0964-1726/22/2/023001.

[47] E. Hawkes, Byoungkwon An, N. M. Benbernou, H. Tanaka, S. Kim, Erik D. Demaine, Daniela Rus, and Robert J. Wood. Programmable matter by folding. *Proceedings of the National Academy of Sciences*, 107(28):12441–12445, 2010. ISSN 0027-8424. doi:10.1073/pnas.0914069107. URL http://www.pnas.org/lookup/doi/10.1073/pnas.0914069107.

[48] Liang He, Gierad Laput, Eric Brockmeyer, and Jon E. Froehlich. SqueezaPulse. In *Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '17*,

pages 341–350, New York, New York, USA, 2017. ACM Press. ISBN 9781450346764. doi:10.1145/3024969.3024976. URL http://dl.acm.org/citation.cfm?doid=3024969.3024976.

[49] Trishan A. M. Hewage, Kim L. Alderson, Andrew Alderson, and Fabrizio Scarpa. Double-Negative Mechanical Metamaterials Displaying Simultaneous Negative Stiffness and Negative Poisson's Ratio Properties. *Advanced Materials*, 28(46):10323–10332, dec 2016. ISSN 09359648. doi:10.1002/adma.201603959. URL http://doi.wiley.com/10.1002/adma.201603959.

[50] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. crdbrd: Shape Fabrication by Sliding Planar Slices. *Computer Graphics Forum (Eurographics)*, 31(2):583–592, 2012. ISSN 01677055. doi:10.1111/j.1467-8659.2012.03037.x. URL http://doi.wiley.com/10.1111/j.1467-8659.2012.03037.x.

[51] Jonathan Hiller and Hod Lipson. VoxCAD. last accessed on 2018-06-19. URL https://www.creativemachineslab.com/voxcad.html.

[52] Larry L. Howell, Spencer P. Magleby, and Brian M. Olsen. *Handbook of Compliant Mechanisms*. John Wiley and Sons, 2013.

[53] Scott E. Hudson. Printing Teddy Bears: A Technique for 3D Printing of Soft Interactive Objects Scott. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 459–468, New York, New York, USA, 2014. ACM Press. ISBN 9781450324731. doi:10.1145/2556288.2557338. URL http://dl.acm.org/citation.cfm?doid=2556288.2557338.

[54] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3D freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, 1999. ISBN 0201485605. doi:10.1145/311535.311602. URL http://portal.acm.org/citation.cfm?doid=311535.311602.

[55] I.materialise. Top 20: Most Popular 3D Modeling & Design Software for 3D Printing (2017 Update). last accessed on 2018-06-25, 2017. URL https://i.materialise.com/blog/en/top-25-most-popular-3d-modeling-design-software-for-3d-printing/.

[56] Alexandra Ion, Johannes Frohnhofen, Ludwig Wall, Robert Kovacs, Mirela Alistar, Jack Lindsay, Pedro Lopes, Hsiang-Ting Chen, and Patrick Baudisch. Metamaterial Mechanisms. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 529–539, New York, New York, USA, 2016. ACM Press. ISBN 9781450341899. doi:10.1145/2984511.2984540.

[57] Alexandra Ion, Ludwig Wall, Robert Kovacs, and Patrick Baudisch. Digital Mechanical Metamaterials. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 977–988, New York, New York, USA, 2017. ACM Press. ISBN 9781450346559. doi:10.1145/3025453.3025624.

[58] Alexandra Ion, Robert Kovacs, Oliver S. Schneider, Pedro Lopes, and Patrick Baudisch. Metamaterial Textures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, New York, New York, USA, 2018. ACM Press. ISBN 9781450356206. doi:10.1145/3173574.3173910.

[59] Alexandra Ion, David Lindlbauer, Philipp Herholz, Marc Alexa, and Patrick Baudisch. Understanding metamaterial mechanisms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 647:1–647:14, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-5970-2. doi:10.1145/3290605.3300877.

[60] Yoshio Ishiguro and Ivan Poupyrev. 3D printed interactive speakers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1733–1742, 2014. ISBN 9781450324731. doi:10.1145/2556288.2557046. URL http://dl.acm.org/citation.cfm?doid=2556288.2557046.

[61] Yunyao Jiang and Yaning Li. Novel 3D-Printed Hybrid Auxetic Mechanical Metamaterial with Chirality-Induced Sequential Cell Opening Mechanisms. *Advanced Engineering Materials*, 20(2):1–9, 2018. ISSN 15272648. doi:10.1002/adem.201700744.

[62] Yuichiro Katsumoto, S. Tokuhisa, and M. Inakage. Ninja track: design of electronic toy variable in shape and flexibility. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, pages 17–24, 2013. ISBN 9781450318983. doi:10.1145/2460625.2460628. URL http://dl.acm.org/citation.cfm?id=2460628.

[63] Narayanan Kidambi, Ryan L. Harne, and K. W. Wang. Energy capture and storage in asymmetrically multistable modular structures inspired by muscle. *Smart Materials and Structures*, 26, 2017. ISSN 0964-1726. doi:https://doi.org/10.1088/1361-665X/aa721a.

[64] Scott Kirkpatrick, C. D. Gelatt, and Mario P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983. ISSN 0036-8075. doi:10.1126/science.220.4598.671.

[65] Helena M.A. Kolken and Amir A. Zadpoor. Auxetic mechanical metamaterials, 2017. ISSN 20462069. URL http://dx.doi.org/10.1039/C6RA27333E.

[66] Helena M.A. Kolken, Shahram Janbaz, Sander M.A. Leeflang, Karel Lietaert, Harrie H. Weinans, and Amir A. Zadpoor. Rationally designed meta-implants: A combination of auxetic and conventional meta-biomaterials. *Materials Horizons*, 5:28–35, 2018. ISSN 20516355. doi:10.1039/c7mh00699c. URL http://dx.doi.org/10.1039/c7mh00699c.

[67] Mina Konaković, Keenan Crane, Bailin Deng, Sofien Bouaziz, Daniel Piker, and Mark Pauly. Beyond Developable: Computational Design and Fabrication with Auxetic Materials. *ACM Transactions on Graphics*, 35(4):1–11, jul 2016. ISSN 07300301. doi:10.1145/2897824.2925944. URL http://dl.acm.org/citation.cfm?doid=2897824.2925944.

[68] X.Y. Kou and S.T. Tan. Heterogeneous object modeling: A review. *Computer-Aided Design*, 39(4):284–301, 2007. ISSN 00104485. doi:10.1016/j.cad.2006.12.007.

[69] Roderic Lakes. Foam Structures with a Negative Poisson's Ratio. *Science*, 235(4792):1038–1040, feb 1987. ISSN 0036-8075. doi:10.1126/science.235.4792.1038. URL http://www.sciencemag.org/cgi/doi/10.1126/science.235.4792.1038.

[70] Gierad Laput, Xiang 'Anthony' Chen, and Chris Harrison. 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*, pages

593–597, New York, New York, USA, 2015. ACM Press. ISBN 9781450337793. doi:10.1145/2807442.2807484. URL http://dl.acm.org/citation.cfm?doid=2807442.2807484.

[71] David Ledo, Fraser Anderson, Ryan Schmidt, Lora Oehlberg, Saul Greenberg, and Tovi Grossman. Pineal: Bringing Passive Objects to Life with Embedded Mobile Devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2583–2593, New York, New York, USA, 2017. ACM Press. ISBN 9781450346559. doi:10.1145/3025453.3025652. URL http://dl.acm.org/citation.cfm?doid=3025453.3025652.

[72] Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. Build-to-Last : Strength to Weight 3D Printed Objects. *ACM Transactions on Graphics*, 33(4), 2014. ISSN 15577333. doi:10.1145/2601097.2601168.

[73] Nilesh D. Mankame and G. K. Ananthasuresh. Topology optimization for synthesis of contact-aided compliant mechanisms using regularized contact modeling. *Computers and Structures*, 82(15-16):1267–1290, 2004. ISSN 00457949. doi:10.1016/j.compstruc.2004.02.024.

[74] Nilesh D. Mankame and G. K. Ananthasuresh. Synthesis of contact-aided compliant mechanisms for non-smooth path generation. *International Journal for Numerical Methods in Engineering*, 69(12): 2564–2605, 2006. ISSN 0743-1619. doi:10.1002/nme.1861. URL http://arxiv.org/abs/1201.4903.

[75] Yiqi Mao, Kai Yu, Michael S. Isakov, Jiangtao Wu, Martin L. Dunn, and H. Jerry Qi. Sequential Self-Folding Structures by 3D Printed Digital Shape Memory Polymers. *Scientific Reports*, 5:1–12, 2015. ISSN 20452322. doi:10.1038/srep13616. URL http://dx.doi.org/10.1038/srep13616.

[76] Jonàs Martínez, Samuel Hornus, Haichuan Song, and Sylvain Lefebvre. Polyhedral voronoi diagrams for additive manufacturing. *ACM Trans. Graph.*, 37(4):129:1–129:15, July 2018. ISSN 0730-0301. doi:10.1145/3197517.3201343. URL http://doi.acm.org/10.1145/3197517.3201343.

[77] James McCrae, Nobuyuki Umetani, and Karan Singh. FlatFitFab: Interactive Modeling with Planar Sections. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*, pages 13–22, New York, New York, USA, 2014. ACM Press. ISBN 9781450330695. doi:10.1145/2642918.2647388. URL http://dl.acm.org/citation.cfm?doid=2642918.2647388.

[78] Vittorio Megaro, Bernhard Thomaszewski, Damien Gauge, Eitan Grinspun, Stelian Coros, and Markus Gross. ChaCra: an interactive design system for rapid character crafting. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 123–130. Eurographics Association, 2014. URL http://dl.acm.org/citation.cfm?id=2849517.2849538.

[79] Vittorio Megaro, Bernhard Thomaszewski, Maurizio Nitti, Otmar Hilliges, Markus Gross, and Stelian Coros. Interactive design of 3D-printable robotic creatures. *ACM Transactions on Graphics*, 34 (6), oct 2015. ISSN 07300301. doi:10.1145/2816795.2818137. URL http://dl.acm.org/citation.cfm?doid=2816795.2818137.

[80] Vittorio Megaro, Jonas Zehnder, Moritz Bächer, Stelian Coros, Markus Gross, and Bernhard Thomaszewski. A computational design tool for compliant mechanisms. *ACM Transactions on Graphics*, 36(4):1–12, 2017. ISSN 07300301. doi:10.1145/3072959.3073636. URL http://dl.acm.org/citation.cfm?doid=3072959.3073636.

[81] Gianluca Memoli, Mihai Caleap, Michihiro Asakawa, Deepak R. Sahoo, Bruce W. Drinkwater, and Sriram Subramanian. Metamaterial bricks and quantization of meta-surfaces. *Nature Communications*, 8:14608, feb 2017. ISSN 2041-1723. doi:10.1038/ncomms14608. URL http://www.nature.com/doifinder/10.1038/ncomms14608.

[82] Ralph C. Merkle. Two types of mechanical reversible logic. *Nanotechnology*, 4(2):114–131, apr 1993. ISSN 0957-4484. doi:10.1088/0957-4484/4/2/007. URL http://stacks.iop.org/0957-4484/4/i=2/a=007?key=crossref.fe14986a5d829dd81f6598fa6b3397f8.

[83] Lucas R. Meza and Julia R. Greer. Mechanical characterization of hollow ceramic nanolattices. *Journal of Materials Science*, 49(6):2496–2508, 2014. ISSN 00222461. doi:10.1007/s10853-013-7945-x.

[84] Lucas R. Meza, Satyajit Das, and Julia R. Greer. Strong, lightweight, and recoverable three-dimensional ceramic nanolattices. *Science*, 345(6202):1322–1326, 2014. ISSN 0036-8075. doi:10.1126/science.1255908. URL http://www.sciencemag.org/cgi/doi/10.1126/science.1255908.

[85] Lucas R. Meza, Alex J. Zelhofer, Nigel Clarke, Arturo J. Mateos, Dennis M. Kochmann, and Julia R. Greer. Resilient 3D hierarchical architected metamaterials. *Proceedings of the National Academy of Sciences*, 112(37):11502–11507, 2015. ISSN 0027-8424. doi:10.1073/pnas.1509120112. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1509120112.

[86] Panagiotis Michalatos and Andrew O. Payne. Monolith. last accessed on 2018-06-19. URL http://www.monolith.zone/{#}introduction.

[87] Graeme W. Milton and Andrej V. Cherkaev. Which Elasticity Tensors are Realizable? *Journal of Engineering Materials and Technology*, 117 (4):483, 1995. ISSN 00944289. doi:10.1115/1.2804743.

[88] Mariam Mir, Murtaza Najabat Ali, Javaria Sami, and Umar Ansari. Review of Mechanics and Applications of Auxetic Structures. *Advances in Materials Science and Engineering*, 2014:1–17, 2014. ISSN 1687-8434. doi:10.1155/2014/753496. URL http://www.hindawi.com/journals/amse/2014/753496/.

[89] M. J. Mirzaali, Shahram Janbaz, M. Strano, L. Vergani, and Amir A. Zadpoor. Shape-matching soft mechanical metamaterials. *Scientific Reports*, 8(965):1–7, 2018. ISSN 20452322. doi:10.1038/s41598-018-19381-3. URL http://dx.doi.org/10.1038/s41598-018-19381-3.

[90] Koryo Miura. Method of packaging and deployment of large membranes in space. Technical report, 1985. URL https://repository.exst.jaxa.jp/dspace/handle/a-is/7293.

[91] Shuhei Miyashita, Steven Guitron, Marvin Ludersdorfer, Cynthia Sung, and Daniela Rus. An untethered miniature origami robot that self-folds, walks, swims, and degrades. In *Proceedings of*

IEEE International Conference on Robotics and Automation, volume 2015-June, pages 1490–1496, 2015. ISBN 978-1-4799-6923-4. doi:10.1109/ICRA.2015.7139386.

[92] Shuhei Miyashita, Steven Guitron, Kazuhiro Yoshida, Shuguang Li, Dana D Damian, and Daniela Rus. Ingestible, controllable, and degradable origami robot for patching stomach wounds. In *2016 IEEE International Conference on Robotics and Automation*, pages 909–916. IEEE, may 2016. ISBN 978-1-4673-8026-3. doi:10.1109/ICRA.2016.7487222. URL http://ieeexplore.ieee.org/document/7487222/.

[93] Lauren C. Montemayor, Lucas R. Meza, and Julia R. Greer. Design and Fabrication of Hollow Rigid Nanolattices via Two-Photon Lithography. *Advanced Engineering Materials*, 16(2):184–189, feb 2014. ISSN 14381656. doi:10.1002/adem.201300254. URL http://doi.wiley.com/10.1002/adem.201300254.

[94] Yuki Mori and Takeo Igarashi. Plushie: An Interactive Design System for Plush Toys. *ACM Transactions on Graphics*, 26(3), jul 2007. ISSN 07300301. doi:10.1145/1239451.1239496. URL http://portal.acm.org/citation.cfm?doid=1276377.1276433.

[95] Davood Mousanezhad, Sahab Babaee, Hamid Ebrahimi, Ranajay Ghosh, Abdelmagid Salem Hamouda, Katia Bertoldi, and Ashkan Vaziri. Hierarchical honeycomb auxetic metamaterials. *Scientific Reports*, 5:1–8, 2015. ISSN 20452322. doi:10.1038/srep18306. URL http://dx.doi.org/10.1038/srep18306.

[96] Stefanie Mueller, Pedro Lopes, and Patrick Baudisch. Interactive Construction: Interactive Fabrication of Functional Mechanical Devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, page 599, New York, New York, USA, 2012. ACM Press. ISBN 9781450315807. doi:10.1145/2380116.2380191. URL http://dl.acm.org/citation.cfm?id=2380191.

[97] Stefanie Mueller, Bastian Kruck, and Patrick Baudisch. LaserOrigami: Laser-Cutting 3D Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page

2585, New York, New York, USA, 2013. ACM Press. ISBN 9781450318990. doi:10.1145/2470654.2481358. URL http://dl.acm.org/citation.cfm?doid=2470654.2481358.

[98] Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. WirePrint: 3D Printed Previews for Fast Prototyping. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 273–280, New York, New York, USA, 2014. ACM Press. ISBN 9781450330695. doi:10.1145/2642918.2647359. URL http://dl.acm.org/citation.cfm?doid=2642918.2647359.

[99] Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. faBrickation: Fast 3D Printing of Functional Objects by Integrating Construction Kit Building Blocks. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 3827–3834, New York, New York, USA, 2014. ACM Press. ISBN 9781450324731. doi:10.1145/2556288.2557005. URL http://dl.acm.org/citation.cfm?id=2556288.2557005.

[100] Tom Mullin, S. Deschanel, Katia Bertoldi, and Mary C. Boyce. Pattern transformation triggered by deformation. *Physical Review Letters*, 99 (8):1–4, 2007. ISSN 00319007. doi:10.1103/PhysRevLett.99.084301.

[101] Neel Nadkarni, Chiara Daraio, and Dennis M. Kochmann. Dynamics of periodic mechanical structures containing bistable elastic elements: From elastic to solitary awave propagation. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 90(2):1–13, 2014. ISSN 15502376. doi:10.1103/PhysRevE.90.023204.

[102] Robin M Neville, Fabrizio Scarpa, and Alberto Pirrera. Shape morphing Kirigami mechanical metamaterials. *Scientific Reports*, 6:1–12, 2016. ISSN 20452322. doi:10.1038/srep31067. URL http://dx.doi.org/10.1038/srep31067.

[103] Hyunjoo Oh, Tung D. Ta, Ryo Suzuki, Mark D. Gross, Yoshihiro Kawahara, and Lining Yao. PEP (3D Printed Electronic Papercrafts): An Integrated Approach for 3D Sculpting Paper-Based Electronic Devices. In *Proceedings of the 2018 CHI Conference on Human Factors*

in Computing Systems, pages 1–12, New York, New York, USA, 2018. ACM Press. ISBN 9781450356206. doi:10.1145/3173574.3174015. URL http://dl.acm.org/citation.cfm?doid=3173574.3174015.

[104] Jifei Ou, Gershon Dublon, Chin-Yi Cheng, Felix Heibeck, Karl Willis, and Hiroshi Ishii. Cilllia. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, number c, pages 5753–5764, New York, New York, USA, 2016. ACM Press. ISBN 9781450333627. doi:10.1145/2858036.2858257. URL http://dl.acm.org/citation.cfm?doid=2858036.2858257.

[105] Jifei Ou, Mélina Skouras, Nikolaos Vlavianos, Felix Heibeck, Chin-Yi Cheng, Jannik Peters, and Hiroshi Ishii. aeroMorph - Heat-sealing Inflatable Shape-change Materials for Interaction Design. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 121–132, New York, New York, USA, 2016. ACM Press. ISBN 9781450341899. doi:10.1145/2984511.2984520. URL http://dl.acm.org/citation.cfm?doid=2984511.2984520.

[106] Jifei Ou, Zhao Ma, Jannik Peters, Sen Dai, Nikolaos Vlavianos, and Hiroshi Ishii. KinetiX - designing auxetic-inspired deformable material structures. *Computers & Graphics*, 75:72–81, oct 2018. ISSN 00978493. doi:10.1016/j.cag.2018.06.003. URL https://doi.org/10.1016/j.cag.2018.06.003https://linkinghub.elsevier.com/retrieve/pii/S0097849318301006.

[107] Johannes T.B. Overvelde and Katia Bertoldi. Relating pore shape to the non-linear response of periodic elastomeric structures. *Journal of the Mechanics and Physics of Solids*, 64(1):351–366, mar 2014. ISSN 00225096. doi:10.1016/j.jmps.2013.11.014. URL http://dx.doi.org/10.1016/j.jmps.2013.11.014.

[108] Johannes T.B. Overvelde, Sicong Shan, and Katia Bertoldi. Compaction through buckling in 2D periodic, soft and porous structures: Effect of pore shape. *Advanced Materials*, 24(17):2337–2342, 2012. ISSN 09359648. doi:10.1002/adma.201104395.

[109] Johannes T.B. Overvelde, Twan A. de Jong, Yanina Shevchenko, Sergio A. Becerra, George M. Whitesides, James C. Weaver, Chuck Hoberman, and Katia Bertoldi. A three-dimensional actuated origami-inspired transformable metamaterial with multiple degrees

of freedom. *Nature Communications*, 7:1–8, 2016. ISSN 20411723. doi:10.1038/ncomms10929. URL http://dx.doi.org/10.1038/ncomms10929.

[110] Johannes T.B. Overvelde, James C. Weaver, Chuck Hoberman, and Katia Bertoldi. Rational design of reconfigurable prismatic architected materials. *Nature*, 541(7637):347–352, 2017. ISSN 14764687. doi:10.1038/nature20824. URL http://dx.doi.org/10.1038/nature20824.

[111] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. Elastic Textures for Additive Fabrication. *ACM Transactions on Graphics*, 34(4), 2015. doi:10.1145/2766937.

[112] Julian Panetta, Abtin Rahimian, and Denis Zorin. Worst-case stress relief for microstructures. *ACM Transactions on Graphics*, 36(4):1–16, 2017. ISSN 07300301. doi:10.1145/3072959.3073649. URL http://dl.acm.org/citation.cfm?doid=3072959.3073649.

[113] Jayson Paulose, Anne S. Meeussen, and Vincenzo Vitelli. Selective buckling via states of self-stress in topological metamaterials. *arXiv preprint*, page 12, 2015. URL http://arxiv.org/abs/1502.03396.

[114] Huaishu Peng, Jennifer Mankoff, Scott E. Hudson, and James McCann. A Layered Fabric 3D Printer for Soft Interactive Objects. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1789–1798, New York, New York, USA, 2015. ACM Press. ISBN 9781450331456. doi:10.1145/2702123.2702327. URL http://dl.acm.org/citation.cfm?doid=2702123.2702327.

[115] Huaishu Peng, Amit Zoran, and François Guimbretière. D-Coil: A Hands-on Approach to Digital 3D Models Design. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1807–1815, New York, New York, USA, 2015. ACM Press. ISBN 9781450331456. doi:10.1145/2702123.2702381. URL http://dl.acm.org/citation.cfm?doid=2702123.2702381.

[116] Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. On-The-Fly Print: Incremental Printing While Modeling. In *Proceedings of the 2016 CHI Conference on Human Factors in Com-*

puting Systems, pages 887–896, New York, New York, USA, 2016. ACM Press. ISBN 9781450333627. doi:10.1145/2858036.2858106. URL http://dl.acm.org/citation.cfm?doid=2858036.2858106.

[117] Huaishu Peng, Jimmy Briggs, Cheng-Yao Wang, Kevin Guo, Joseph Kider, Stefanie Mueller, Patrick Baudisch, and François Guimbretière. RoMA: Interactive Fabrication with Augmented Reality and a Robotic 3D Printer. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, New York, New York, USA, 2018. ACM Press. ISBN 9781450356206. doi:10.1145/3173574.3174153. URL http://dl.acm.org/citation.cfm?doid=3173574.3174153.

[118] Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. Design and fabrication of flexible rod meshes. *ACM Transactions on Graphics*, 34(4), jul 2015. ISSN 07300301. doi:10.1145/2766998. URL http://dl.acm.org/citation.cfm?doid=2809654.2766998.

[119] Jesús Pérez, Miguel A. Otaduy, and Bernhard Thomaszewski. Computational design and automated fabrication of kirchhoff-plateau surfaces. *ACM Transactions on Graphics*, 36(4):1–12, 2017. ISSN 07300301. doi:10.1145/3072959.3073695. URL http://dl.acm.org/citation.cfm?doid=3072959.3073695.

[120] Austin A. Phoenix and Evan Wilson. Variable thermal conductance metamaterials for passive or active thermal management. In *Proceedings of the ASME 2017 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, pages 1–10, 2017.

[121] Clemens Preisinger. Linking structure and parametric geometry. *Architectural Design*, 83(2):110–113, 2013. ISSN 00038504. doi:10.1002/ad.1564.

[122] Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. Make It Stand: Balancing Shapes for 3D Fabrication. *ACM Transactions on Graphics*, 32(4), 2013. ISSN 0730-0301. doi:10.1145/2461912.2461957. URL http://doi.acm.org/10.1145/2461912.2461957.

[123] Romain Prévost, Wojciech Jarosz, Moritz Bächer, Wojciech Jarosz, and Olga Sorkine-Hornung. Balancing 3D Models with Movable Masses. In *Vision, Modeling & Visualization*, page 8. The Eurographics Association, 2016. ISBN 9783038680253. doi:10.2312/vmv.20161337. URL https://diglib.eg.org:443/handle/10.2312/vmv20161337.

[124] J. Qiu, J.H. Lang, and A.H. Slocum. A Curved-Beam Bistable Mechanism. *Journal of Microelectromechanical Systems*, 13(2): 137–146, 2004. ISSN 1057-7157. doi:10.1109/JMEMS.2004.825308. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1284352.

[125] Ahmad Rafsanjani and Katia Bertoldi. Buckling-Induced Kirigami. *Physical Review Letters*, 118(8):1–5, 2017. ISSN 10797114. doi:10.1103/PhysRevLett.118.084301.

[126] Ahmad Rafsanjani and Damiano Pasini. Bistable auxetic mechanical metamaterials inspired by ancient geometric motifs. *Extreme Mechanics Letters*, 9:291–296, 2016. ISSN 23524316. doi:10.1016/j.eml.2016.09.001. URL http://dx.doi.org/10.1016/j.eml.2016.09.001.

[127] Ahmad Rafsanjani, Abdolhamid Akbarzadeh, and Damiano Pasini. Snapping Mechanical Metamaterials under Tension. *Advanced Materials*, 27(39):5931–5935, 2015. ISSN 15214095. doi:10.1002/adma.201502809.

[128] D.Janaki Ram, T.H. Sreenivas, and K.Ganapathy Subramaniam. Parallel Simulated Annealing Algorithms. *Journal of Parallel and Distributed Computing*, 37(2):207–212, 1996. ISSN 0743-7315. doi:10.1006/jpdc.1996.0121.

[129] Jordan R. Raney, Neel Nadkarni, Chiara Daraio, Dennis M. Kochmann, Jennifer A. Lewis, and Katia Bertoldi. Stable propagation of mechanical signals in soft media using stored elastic energy. *Proceedings of the National Academy of Sciences*, page 201604838, 2016. ISSN 0027-8424. doi:10.1073/pnas.1604838113. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1604838113.

[130] Dan Raviv, Wei Zhao, Carrie McKnelly, Athina Papadopoulou, Achuta Kadambi, Boxin Shi, Shai Hirsch, Daniel Dikovsky, Michael Zyracki, Carlos Olguin, Ramesh Raskar, and Skylar Tibbits. Active Printed Materials for Complex Self-Evolving Deformations. *Scientific Reports*, 4(1):7422, 2015. ISSN 2045-2322. doi:10.1038/srep07422. URL http://www.nature.com/articles/srep07422.

[131] Xin Ren, Raj Das, Phuong Tran, Tuan Duc Ngo, and Yi Min Xie. Auxetic metamaterials and structures: a review. *Smart Materials and Structures*, 27, 2018. URL http://iopscience.iop.org/article/10.1088/1361-665X/aaa61c/pdf.

[132] David Restrepo, Nilesh D. Mankame, and Pablo D. Zavattieri. Phase transforming cellular materials. *Extreme Mechanics Letters*, 4:52–60, 2015. ISSN 23524316. doi:10.1016/j.eml.2015.08.001. URL http://dx.doi.org/10.1016/j.eml.2015.08.001.

[133] Michael L. Rivera, Melissa Moukperian, Daniel Ashbrook, Jennifer Mankoff, and Scott E. Hudson. Stretching the Bounds of 3D Printing with Embedded Textiles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, number May, pages 1–12, 2017. ISBN 9781450346559. doi:10.1145/3025453.3025460. URL papers3://publication/doi/10.1145/3025453.3025460.

[134] David W. Rosen. Design for additive manufacturing: A method to explore unexplored regions of the design space. In *Eighteenth Annual Solid Freeform Fabrication Symposium*, pages 402–415, 2007. URL https://sffsymposium.engr.utexas.edu/Manuscripts/2007/2007-34-Rosen.pdf.

[135] Thijs Roumen, Bastian Kruck, Tobias Dürschmid, Tobias Nack, and Patrick Baudisch. Mobile Fabrication. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 3–14, New York, New York, USA, 2016. ACM Press. ISBN 9781450341899. doi:10.1145/2984511.2984586. URL http://dl.acm.org/citation.cfm?doid=2984511.2984586.

[136] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. SketchChair: An All-in-one Chair Design System for End Users. In *Proceedings of the 5th International Conference on Tangible and*

Embedded Interaction, pages 73–80, 2011. ISBN 9781450304788. doi:10.1145/1935701.1935717. URL http://dx.doi.org/10.1145/1935701.1935717.

[137] Valkyrie Savage, Colin Chang, and Björn Hartmann. Sauron: embedded single-camera sensing of printed physical user interfaces. In *Proceedings of the annual ACM symposium on User interface software and technology*, pages 447–456, 2013. ISBN 9781450322683. doi:10.1145/2501988.2501992. URL http://dl.acm.org/citation.cfm?id=2501992.

[138] Valkyrie Savage, Ryan Schmidt, Tovi Grossman, George Fitzmaurice, and Björn Hartmann. A Series of Tubes : Adding Interactivity to 3D Prints Using Internal Pipes. In *Proceedings of the annual ACM symposium on User interface software and technology*, pages 3–12, 2014. ISBN 9781450330695. doi:10.1145/2642918.2647374.

[139] Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B Goldman, Gautham Mysore, and Wilmot Li. Lamello: Passive Acoustic Sensing for Tangible Input Components. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, volume 1, pages 1277–1280, New York, New York, USA, 2015. ACM Press. ISBN 9781450331456. doi:10.1145/2702123.2702207. URL http://dl.acm.org/citation.cfm?doid=2702123.2702207.

[140] Anupam Saxena. Synthesis of Compliant Mechanisms for Path Generation using Genetic Algorithm. *Journal of Mechanical Design*, 127(4):745, 2005. ISSN 10500472. doi:10.1115/1.1899178. URL http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1448589.

[141] Krishna Kumar Saxena, Raj Das, and Emilio P. Calius. Three Decades of Auxetics Research – Materials with Negative Poisson's Ratio: A Review. *Advanced Engineering Materials*, 18(11):1847–1870, nov 2016. ISSN 14381656. doi:10.1002/adem.201600053. URL http://doi.wiley.com/10.1002/adem.201600053.

[142] Mark Schenk and Simon D. Guest. Geometry of Miura-folded metamaterials. *Proceedings of the National Academy of Sciences*, 110 (9):3276–3281, 2013. ISSN 0027-8424. doi:10.1073/pnas.1217998110. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1217998110.

[143] Martin Schmitz, Jürgen Steimle, Jochen Huber, Niloofar Dezfuli, and Max Mühlhäuser. Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1001–1014, 2017. ISBN 9781450346559. doi:10.1145/3025453.3025663. URL http://dx.doi.org/10.1145/3025453.3025663{%}0A.

[144] Oliver Schneider, Karon MacLean, Colin Swindells, and Kellogg Booth. Haptic experience design: What hapticians do and where they need help. *International Journal of Human-Computer Studies*, 107:5–21, nov 2017. ISSN 10715819. doi:10.1016/j.ijhcs.2017.04.004. URL http://dx.doi.org/10.1016/j.ijhcs.2017.04.004.

[145] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. Microstructures to control elasticity in 3D printing. *ACM Transactions on Graphics*, 34(4), 2015. ISSN 07300301. doi:10.1145/2766926. URL http://dl.acm.org/citation.cfm?doid=2809654.2766926.

[146] Christian Schumacher, Steve Marschner, Markus Gross, and Bernhard Thomaszewski. Mechanical Characterization of structured sheet materials. *ACM Transactions on Graphics*, 37(4), 2018. doi:10.1145/3197517.3201278.

[147] Sclupteo. The state of 3D printing 2018. Technical report, 2018. URL https://www.sculpteo.com/en/get/report/state{_}of{_}3D{_}printing{_}2018/.

[148] Hamed Seifi, Anooshe Rezaee Javan, Arash Ghaedizadeh, Jianhu Shen, Shanqing Xu, and Yi Min Xie. Design of hierarchical structures for synchronized deformations. *Scientific Reports*, 7(January): 1–7, 2017. ISSN 20452322. doi:10.1038/srep41183. URL http://dx.doi.org/10.1038/srep41183.

[149] Sicong Shan, Sung H. Kang, Jordan R. Raney, Pai Wang, Lichen Fang, Francisco Candido, Jennifer A. Lewis, and Katia Bertoldi. Multistable Architected Materials for Trapping Elastic Strain Energy. *Advanced Materials*, 27(29):4296–4301, 2015. ISSN 15214095. doi:10.1002/adma.201501708.

[150] Sicong Shan, Sung H. Kang, Zhenhao Zhao, Lichen Fang, and Katia Bertoldi. Design of planar isotropic negative Poisson's ratio structures. *Extreme Mechanics Letters*, 4:96–102, 2015. ISSN 23524316. doi:10.1016/j.eml.2015.05.002. URL http://dx.doi.org/10.1016/j.eml.2015.05.002.

[151] Jongmin Shim, Claude Perdigou, Elizabeth R. Chen, Katia Bertoldi, and Pedro M. Reis. Buckling-induced encapsulation of structured elastic shells under pressure. *Proceedings of the National Academy of Sciences*, 109(16):5978–5983, 2012. ISSN 0027-8424. doi:10.1073/pnas.1115674109.

[152] Jongmin Shim, Sicong Shan, Andrej Košmrlj, Sung H. Kang, Elizabeth R. Chen, James C. Weaver, and Katia Bertoldi. Harnessing instabilities for design of soft reconfigurable auxetic/chiral materials. *Soft Matter*, 9(34):8198–8202, 2013. ISSN 1744-683X. doi:10.1039/c3sm51148k. URL http://xlink.rsc.org/?DOI=c3sm51148k.

[153] Ole Sigmund. On the Design of Compliant Mechanisms Using Topology Optimization. *Mechanics of Structures and Machines*, 25 (4):493–524, 1997. ISSN 0890-5452. doi:10.1080/08905459708945415. URL http://www.tandfonline.com/doi/abs/10.1080/08905459708945415.

[154] Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus Gross. Computational design of rubber balloons. *Proceedings of Computer Graphics Forum/Eurographics 2012*, 31(2):835–844, 2012. ISSN 01677055. doi:10.1111/j.1467-8659.2012.03064.x.

[155] Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. Computational design of actuated deformable characters. *ACM Transactions on Graphics*, 32(4):1, 2013. ISSN 07300301. doi:10.1145/2461912.2461979. URL http://dl.acm.org/citation.cfm?doid=2461912.2461979.

[156] Ronit Slyper and Jessica Hodgins. Prototyping robot appearance, movement, and interactions using flexible 3D printing and air pressure sensors. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, volume 1, pages 6–11, 2012. ISBN 9781467346054. doi:10.1109/ROMAN.2012.6343723.

[157] Zeming Song, Xu Wang, Cheng Lv, Yonghao An, Mengbing Liang, Teng Ma, David He, Ying Jie Zheng, Shi Qing Huang, Hongyu Yu, and Hanqing Jiang. Kirigami-based stretchable lithium-ion batteries. *Scientific Reports*, 5:1–9, 2015. ISSN 20452322. doi:10.1038/srep10988. URL http://dx.doi.org/10.1038/srep10988.

[158] Ivan Edward Sutherland. *Sketchpad, a man-machine graphical communication system*. PhD thesis, Massachusetts Institute of Technology, 1 1963. URL http://hdl.handle.net/1721.1/14979. An optional note.

[159] Joshua G. Tanenbaum, Amanda M. Williams, Audrey Desjardins, and Karen Tanenbaum. Democratizing technology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, page 2603, New York, New York, USA, 2013. ACM Press. ISBN 9781450318990. doi:10.1145/2470654.2481360. URL http://dl.acm.org/citation.cfm?id=2481360{%}5Cnhttp://doi.acm.org/10.1145/2470654.2481360.

[160] Alexander Teibrich, Stefanie Mueller, François Guimbretière, Robert Kovacs, Stefan Neubert, and Patrick Baudisch. Patching Physical Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 83–91, New York, New York, USA, 2015. ACM Press. ISBN 9781450337793. doi:10.1145/2807442.2807467. URL http://dl.acm.org/citation.cfm?doid=2807442.2807467.

[161] Bernhard Thomaszewski, Stelian Coros, Damien Gauge, Vittorio Megaro, Eitan Grinspun, and Markus Gross. Computational Design of Linkage-Based Characters. *ACM Transactions on Graphics*, 33(4): 64:1–64:9, 2014.

[162] Skylar Tibbits. 4D Printing: Multi-Material Shape Change. *Architectural Design*, 84(1):116–121, jan 2014. ISSN 00038504. doi:10.1002/ad.1710. URL http://doi.wiley.com/10.1002/ad.1710.

[163] Michael T. Tolley, Samuel M. Felton, Shuhei Miyashita, Byunghyun Shin, Monica Zhou, Daniela Rus, and Robert J. Wood. Self-Folding Shape Memory Laminates for Automated Fabrication. In *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4931–4936, 2013. ISBN 9781467363570.

[164] Brian P Trease, Yong-Mo Moon, and Sridhar Kota. Design of Large-Displacement Compliant Joints. *Journal of Mechanical Design*, 127(4):788, 2005. ISSN 10500472. doi:10.1115/1.1900149. URL http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1448601.

[165] Steve Tsang, Ravin Balakrishnan, Karan Singh, and Abhishek Ranjan. A suggestive interface for image guided 3D sketching. In *Proceedings of the 2004 conference on Human factors in computing systems*, pages 591–598, New York, New York, USA, 2004. ACM Press. ISBN 1581137028. doi:10.1145/985692.985767. URL http://portal.acm.org/citation.cfm?doid=985692.985767.

[166] University of Twente. SPACAR software package for the dynamic modelling and control of flexible multibody systems (University of Twente). last accessed on 2018-06-19. URL https://www.utwente.nl/en/et/ms3/research-chairs/WAoud{_}niets{_}uit{_}wissen{_}aub/software/spacar/2015/intro/.

[167] Francisca Gil Ureta, Chelsea Tymms, and Denis Zorin. Interactive Modeling of Mechanical Objects. *Computer Graphics Forum*, 35(5):145–155, aug 2016. ISSN 01677055. doi:10.1111/cgf.12971. URL http://doi.wiley.com/10.1111/cgf.12971.

[168] Teunis van Manen, Shahram Janbaz, and Amir A. Zadpoor. Programming the shape-shifting of flat soft matter. *Materials Today*, 21(2):144–163, mar 2018. ISSN 13697021. doi:10.1016/j.mattod.2017.08.026. URL http://linkinghub.elsevier.com/retrieve/pii/S1369702117302237.

[169] Tatyana Vasilevitsky and Amit Zoran. Steel-Sense: Integrating Machine Elements with Sensors by Additive Manufacturing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5731–5742, 2016. ISBN 978-1-4503-3362-7. doi:10.1145/2858036.2858309. URL http://doi.acm.org/10.1145/2858036.2858309.

[170] Kiril Vidimče, Szu-po Wang, and Jonathan Ragan-kelley. OpenFab: A Programmable Pipeline for Multi-Material Fabrication. *ACM Transactions on Graphics*, 32(4):1–11, 2013. ISSN 0730-0301. doi:10.1145/2461912.2461993.

[171] Kiril Vidimče, Alexandre Kaspar, Ye Wang, and Wojciech Matusik. Foundry: Hierarchical Material Design for Multi-Material Fabrication. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 563–574, New York, New York, USA, 2016. ACM Press. ISBN 9781450341899. doi:10.1145/2984511.2984516. URL http://dl.acm.org/citation.cfm?doid=2984511.2984516.

[172] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006. ISSN 1436-4646. doi:10.1007/s10107-004-0559-y. URL https://projects.coinor.org/Ipopt.

[173] Guanyun Wang, Lining Yao, Wen Wang, Jifei Ou, Chin-Yi Cheng, and Hiroshi Ishii. xPrint: A Modularized Liquid Printer for Smart Materials Deposition. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pages 5743–5752, New York, New York, USA, 2016. ACM Press. ISBN 9781450333627. doi:10.1145/2858036.2858281. URL http://dl.acm.org/citation.cfm?doid=2858036.2858281.

[174] Guanyun Wang, Tingyu Cheng, Youngwook Do, Humphrey Yang, Ye Tao, Jianzhe Gu, Byoungkwon An, and Lining Yao. 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 - CHI '18*, pages 1–12, New York, New York, USA, 2018. ACM Press. ISBN 9781450356206. doi:10.1145/3173574.3174143. URL http://dl.acm.org/citation.cfm?doid=3173574.3174143.

[175] Hongqing Wang and David W. Rosen. *Computer-Aided Design Methods ForThe Additive Fabrication Of Truss Structure*. PhD thesis, 2001.

[176] Wen Wang, Lining Yao, Chin-Yi Cheng, Teng Zhang, Hiroshi Atsumi, Luda Wang, Guanyun Wang, Oksana Anilionyte, Helene Steiner, Jifei Ou, Kang Zhou, Chris Wawrousek, Katherine Petrecca, Angela M. Belcher, Rohit Karnik, Xuanhe Zhao, Daniel I. C. Wang, and Hiroshi Ishii. Harnessing the hygroscopic and biofluorescent behaviors of genetically tractable microbial cells to design biohybrid wear-

ables. *Science Advances*, 3(5):e1601984, may 2017. ISSN 2375-2548. doi:10.1126/sciadv.1601984. URL http://advances.sciencemag.org/lookup/doi/10.1126/sciadv.1601984.

[177] Wen Wang, Lining Yao, Teng Zhang, Chin-yi Cheng, Daniel Levine, and Hiroshi Ishii. Transformative Appetite: Shape-Changing Food Transforms from 2D to 3D by Water Interaction through Cooking. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 6123–6132, New York, New York, USA, 2017. ACM Press. ISBN 9781450346559. doi:10.1145/3025453.3026019. URL http://dl.acm.org/citation.cfm?doid=3025453.3026019.

[178] Michael Wehner, Ryan L. Truby, Daniel J. Fitzgerald, Bobak Mosadegh, George M. Whitesides, Jennifer A. Lewis, and Robert J. Wood. An integrated design and fabrication strategy for entirely soft, autonomous robots. *Nature*, 536(7617):451–455, 2016. ISSN 14764687. doi:10.1038/nature19100. URL http://dx.doi.org/10.1038/nature19100.

[179] Christian Weichel, Manfred Lau, and Hans Gellersen. Enclosed: A Component-Centric Interface for Designing Prototype Enclosures. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction - TEI '13*, page 215, New York, New York, USA, 2013. ACM Press. ISBN 9781450318983. doi:10.1145/2460625.2460659. URL http://dl.acm.org/citation.cfm?doid=2460625.2460659.

[180] Christian Weichel, Manfred Lau, David Kim, Nicolas Villar, and Hans W. Gellersen. MixFab: A Mixed-reality Environment for Personal Fabrication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3855–3864, 2014. ISBN 9781450324731. doi:10.1145/2556288.2557090. URL http://dl.acm.org/citation.cfm?doid=2556288.2557090.

[181] Christian Weichel, John Hardy, Jason Alexander, and Hans Gellersen. ReForm: Integrating Physical and Digital Design through Bidirectional Fabrication. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 93–102, New York, New York, USA, 2015. ACM Press. ISBN 9781450337793. doi:10.1145/2807442.2807451. URL http://dl.acm.org/citation.cfm?doid=2807442.2807451.

[182] Karl D. D. Willis, C. Xu, K.-J. Wu, G. Levin, and Mark D. Gross. Interactive Fabrication: New Interfaces for Digital Fabrication. In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 69–72, 2011. ISBN 9781450304788. doi:10.1145/1935701.1935716. URL http://doi.acm.org/10.1145/1935701.1935716.

[183] Karl D. D. Willis, Eric Brockmeyer, Scott E. Hudson, and Ivan Poupyrev. Printed optics: 3D printing of embedded optical elements for interactive devices. In *Proceedings of the annual ACM symposium on User interface software and technology*, pages 589–598, 2012. ISBN 9781450315807. doi:10.1145/2380116.2380190. URL http://dl.acm.org/citation.cfm?doid=2380116.2380190.

[184] Jun Wu, Christian Dick, and Rudiger Westermann. A System for High-Resolution Topology Optimization. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1195–1208, 2016. ISSN 10772626. doi:10.1109/TVCG.2015.2502588.

[185] Z. Wu, Y. Zheng, and K. W. Wang. Metastable modular metastructures for on-demand reconfiguration of band structures and nonreciprocal wave propagation. *Physical Review E*, 97(2):1–10, 2018. ISSN 24700053. doi:10.1103/PhysRevE.97.022209.

[186] Dian Yang, Bobak Mosadegh, Alar Ainla, Benjamin Lee, Fatemeh Khashai, Zhigang Suo, Katia Bertoldi, and George M. Whitesides. Buckling of Elastomeric Beams Enables Actuation of Soft Machines. *Advanced Materials*, 27(41):6323–6327, 2015. ISSN 15214095. doi:10.1002/adma.201503188.

[187] Lining Yao, Jifei Ou, Chin-yi Cheng, Helene Steiner, Wen Wang, Guanyun Wang, and Hiroshi Ishii. bioLogic: Natto Cells as Nanoactuators for Shape Changing Interfaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1–10, New York, New York, USA, 2015. ACM Press. ISBN 9781450331456. doi:10.1145/2702123.2702611. URL http://dl.acm.org/citation.cfm?doid=2702123.2702611.

[188] Lining Yao, Jifei Ou, Guanyun Wang, Chin-Yi Cheng, Wen Wang, Helene Steiner, and Hiroshi Ishii. bioPrint: A Liquid Deposition Printing System for Natural Actuators. *3D Printing and Ad-*

ditive Manufacturing, 2(4):168–179, dec 2015. ISSN 2329-7662. doi:10.1089/3dp.2015.0033. URL http://online.liebertpub.com/doi/10.1089/3dp.2015.0033.

[189] Kai Yu, Alexander Ritchie, Yiqi Mao, Martin L. Dunn, and H. Jerry Qi. Controlled Sequential Shape Changing Components by 3D Printing of Shape Memory Polymer Multimaterials. *Procedia IUTAM*, 12:193–203, 2015. ISSN 22109838. doi:10.1016/j.piutam.2014.12.021. URL http://www.sciencedirect.com/science/article/pii/S2210983814000947.

[190] Mohamed Zanaty, Ilan Vardi, and Simon Henein. Programmable Multistable Mechanisms: Synthesis and Modeling. *Journal of Mechanical Design*, 140, 2018. ISSN 1050-0472. doi:10.1115/1.4038926. URL http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4038926.

[191] Jonas Zehnder, Espen Knoop, Moritz Bächer, and Bernhard Thomaszewski. MetaSilicone: Design and Fabrication of Composite Silicone with Desired Mechanical Properties. *ACM Transactions on Graphics*, 36(6):1–13, nov 2017. ISSN 07300301. doi:10.1145/3130800.3130881. URL http://dl.acm.org/citation.cfm?doid=3130800.3130881.

[192] Ran Zhang, Thomas Auzinger, Duygu Ceylan, Wilmot Li, and Bernd Bickel. Functionality-aware retargeting of mechanisms to 3D shapes. *ACM Transactions on Graphics*, 36(4):1–13, jul 2017. ISSN 07300301. doi:10.1145/3072959.3073710. URL http://dl.acm.org/citation.cfm?doid=3072959.3073710.

[193] Xiaoyu Zheng, Howon Lee, Todd H. Weisgraber, Maxim Shusteff, Joshua DeOtte, Eric B. Duoss, Joshua D. Kuntz, Monika M. Biener, Qi Ge, Julie A. Jackson, Sergei O. Kucheyev, Nicholas X. Fang, and Christopher M. Spadaccini. Ultralight, ultrastiff mechanical metamaterials. *Science*, 344(6190):1373–1377, jun 2014. ISSN 0036-8075. doi:10.1126/science.1252291. URL http://www.sciencemag.org/cgi/doi/10.1126/science.1252291.

[194] Benliang Zhu, Xianmin Zhang, and Nianfeng Wang. Topology optimization of hinge-free compliant mechanisms with multiple outputs using level set method. *Structural and Multidisciplinary Optimization*, 47(5):659–672, 2013. doi:10.1007/s00158-012-0841-1.

[195] Benliang Zhu, Xianmin Zhang, and Sergej Fatikow. A multi-objective method of hinge-free compliant mechanism optimization. *Structural and Multidisciplinary Optimization*, 49:431–440, 2014. ISSN 16151488. doi:10.1007/s00158-013-1003-9.

[196] Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. Two-Scale Topology Optimization with Microstructures. *ACM Transactions on Graphics*, 36(5):1–16, jul 2017. ISSN 07300301. doi:10.1145/3095815. URL http://dl.acm.org/citation.cfm?doid=3127587.3095815.