Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

Fabric Faces: A Framework for 3D Printing Foldable Textile Structures

Thesis
submitted to the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Paul Miles Preuschoff

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Jan Bender

Registration date: 27.01.2021
Submission date: 12.04.2021

Zentrales Prüfungsamt/Central Examination Office

RWTH AACHEN UNIVERSITY

Eidesstattliche Versicherung
Statutory Declaration in Lieu of an Oath

Preuschoff, Paul Miles
Name, Vorname/Last Name, First Name

369243
Matrikelnummer (freiwillige Angabe)
Matriculation No. (optional)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende ~~Arbeit~~/Bachelorarbeit/ ~~Masterarbeit~~* mit dem Titel
I hereby declare in lieu of an oath that I have completed the present paper/Bachelor thesis/Master thesis* entitled

Fabric Faces: A Framework for 3D Printing Foldable Textile Structures

selbstständig und ohne unzulässige fremde Hilfe (insbes. akademisches Ghostwriting) erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
independently and without illegitimate assistance from third parties (such as academic ghostwriters). I have used no other than the specified sources and aids. In case that the thesis is additionally submitted in an electronic format, I declare that the written and electronic versions are fully identical. The thesis has not been submitted to any examination body in this, or similar, form.

Ort, Datum/City, Date

Unterschrift/Signature

*Nichtzutreffendes bitte streichen

*Please delete as appropriate

Belehrung:
Official Notification:

§ 156 StGB: Falsche Versicherung an Eides Statt
Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
Para. 156 StGB (German Criminal Code): False Statutory Declarations
Whoever before a public authority competent to administer statutory declarations falsely makes such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment not exceeding three years or a fine.
§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt
(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.
(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.
Para. 161 StGB (German Criminal Code): False Statutory Declarations Due to Negligence
(1) If a person commits one of the offences listed in sections 154 through 156 negligently the penalty shall be imprisonment not exceeding one year or a fine.
(2) The offender shall be exempt from liability if he or she corrects their false testimony in time. The provisions of section 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:
I have read and understood the above official notification:

Ort, Datum/City, Date

Unterschrift/Signature

Contents

List of Figures

List of Tables

Abstract

FabricFaces broadens the design space of personal fabrication by using foldable support structures combined with fabric surfaces. Features of the support structures that simplify the manual assembly are beveled edges, connecting features and adjacent face placement. We introduce a convenient pipeline for digitally unfolding 3D shapes and computing flat support structures for them. Afterwards, those flat structures can be efficiently 3D printed on fabric to manufacture the object.

In this thesis, we developed a fast and intuitive workflow for the *FabricFaces* design process by implementing a tool that computes the required support structures. It allows users to generate and tweak those complex structures easily inside the popular 3D software *Blender*, instead of manually designing them. To be easy and quick to use, our tool provides preview features, four core parameters and value suggestion. If needed, the user can manually adjust the parameters, design connectors with a blueprint-system and specify unfolding details.

We evaluated our tool in a study to analyze how users work with it and how it can be improved. The results show, that all users could use the tool well. Users with low *Blender* expertise could achieve standard results fast, while advanced *Blender* users customized their solutions. Most of the usability features we added turned out to be helpful. From our observations, we derived promising design improvements for the future, for example a simplified previewing mechanism that would display changes live.

In general, our work contributes to the development of the *FabricFaces* approach by automating a complicated process and informing future development decisions.

Überblick

FabricFaces ist eine neue Methode zur individuellen Fertigung die vielseitig einsetzbar ist. Hierbei wird Stoff für die Oberflächen des Objektes genutzt und durch eine faltbare Struktur aufgespannt. Diese Struktur ist mit abgeschrägten Kanten, Steckverbindern und nebenanderliegenden Flächen ausgestattet, die ein einfaches Zusammensetzen ermöglichen. Wir entwickeln eine Anwendung, um 3D Formen digital aufzufalten und deren Stützstrukturen zu erzeugen. Um das Objekt herzustellen, können danach diese Strukturen effizient auf Stoff 3D-gedruckt werden.

Das Werkzeug, dass in dieser Arbeit entwickelt wird, berechnet die erforderlichen Stützstrukturen und vereinfacht den Arbeitsablauf, indem Benutzer sie in der verbreiteten 3D-Software *Blender* zügig designen können. Vorschau-Funktionen, Kernparameter und Vorschläge sollen dafür sorgen, dass das Programm einfach und effektiv zu bedienen ist. Bei Bedarf kann der Benutzer die Parameter manuell einstellen, Steckverbinder mit einem Modell-System entwerfen und definieren wie das Objekt aufgefaltet werden soll.

Wir haben unsere Anwendung in einer Studie analysiert, um herauszufinden wie Nutzer es verwenden, und wie es verbessert werden kann. Die Ergebnisse zeigen, dass alle Nutzer gut mit der Anwendung zurecht kamen. Nutzer die noch nicht viel Blender Erfahrung haben, konnten schnell Standardergebnisse erzielen. Während erfahrene *Blender* Nutzer das Resultat spezifischer angepasst haben. Funktionen, welche die Bedienung erleichtern sollen haben dies meistens geschafft. Aus unseren Beobachtung haben wir vielversprechende zukünftige Weiterentwicklungen unseres Designs abgeleitet. Beispielsweise eine vereinfachte Art der Vorschau, die Änderungen in Echtzeit anzeigen könnte.

Im Allgemeinen trägt unsere Arbeit zur Entwicklung der *FabricFaces* Methode bei, indem sie einen komplizierten Schritt automatisiert und eine Grundlage für zukünftige Entwicklungen geschaffen wird.

Acknowledgements

I would like to thank the people at the Media Computing Group for giving me the opportunity to be part of an interesting project by writing this thesis. I especially want to thank my supervisor Adrian Wagner for his constructive feedback, helpful suggestions and sharing his general expertise.

A big thank you to all the people that participated in the study and helped me gaining fascinating insights and observations. Thank you for your time and effort!

I would also like to thank my family and friends for their general support.

Conventions

Throughout this thesis we use the following conventions.

Names of methods like *Mill and Fold* or products like *Blender* are written italic.

For better clarity, numbers are written as figures when referring to quantities: For example: 8 participants liked this feature.

Definitions of technical terms or short excursus are set off in coloured boxes.

> **EXAMPLE BOX:**
> These boxes contain explanations or definitions for important concepts.

The whole thesis is written in American English.

Chapter 1

Introduction

Personal fabrication is getting more accessible and popular. The wider use triggers innovation and the development of more feasible methods.

Personal fabrication becomes more popular

FabricFaces is such an innovative approach, where the combination of textiles with 3D printed structures opens up new opportunities for personal fabrication. The resulting objects consist of a surface made out of fabric that is supported by a structure (fig. 1.1), which is 3D printed on the textile surface while it is in a flat state. After that, the textile around the print is cut away and the object is assembled by folding it back together (fig. 1.2).

FabricFaces combines textiles and 3D printed structures

This approach enables the fabrication of objects with visual and haptic surface textures on a standard 3D printer that are not accomplishable with standard 3D printing. By selecting soft or patterned textiles for the surfaces, those characteristics could be transferred to the object. More complex properties could be also embedded, for example by using electric embroidery. A key advantage of this method is, that less filament is used for only printing the supporting structure. This presumably results in a cheaper and faster process and lighter objects. In the unfolded state the object is very space efficient and easy to transport.

FabricFaces supplements the design space of personal fabrication

We make *FabricFaces* more accessible

In order to make *FabricFaces* accessible to a wide range of users, the scope of this work is to make the design process of the support structures easy and quick for non-experts, while giving experienced users options for customization. We decided to approach this by implementing an add-on for the 3D software *Blender* since it is well known and open source. The add-on analyzes a given input geometry and computes a corresponding support structure. Instead of labor-intensively deriving a support structure, users only need to adjust some parameters.

We will begin this thesis with reviewing selected related work from the domain of stylized fabrication to show what has been already done, and how our method supplements the domain. We focus on fabrication methods that use two dimensional components. This includes connecting flat parts, pop-ups and folding based approaches. In the next step, work regarding the combination of 3D printing with other materials is explored.

Our tool is presented and evaluated in a user study

We then introduce the tool we developed. Starting from the user perspective, we outline how the expected workflows could look like. After that, we switch to the developer perspective and explain how the add-on is implemented. Features and functionalities are documented and explained. In the following qualitative study with 13 participants, we compare our ideal workflow with the workflow of real users and evaluate the overall usability of the tool. After the procedure of the study is outlined, the findings and conclusions are presented. We propose additional features to further enhance the usability and outline promising avenues for future work.

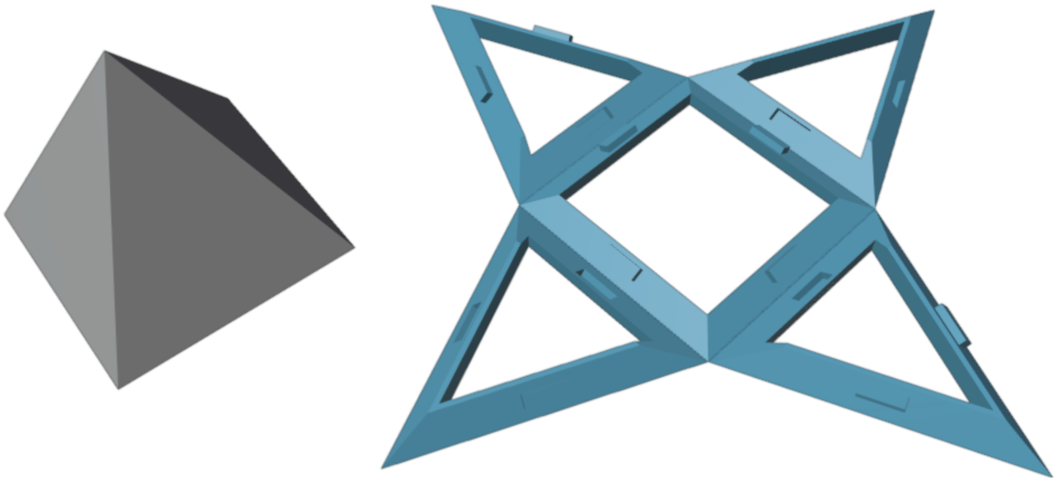

Figure 1.1: Example output of our implementation. Left: A pyramid as the input mesh. Right: A support structure that was generated by our implementation. It consists of face edges that span the fabric, and connectors on those edges.

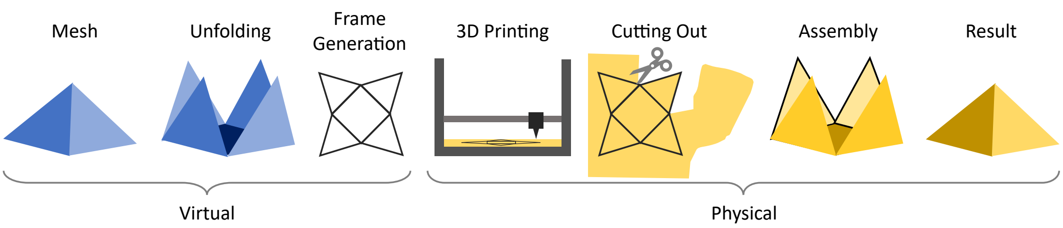

Figure 1.2: The complete fabrication process for the *FabricFaces* approach. From left to right: The initial virtual mesh, algorithmically finding cuts and unfolding the mesh, generating the support frame, 3D printing the support frame on fabric, cutting the fabric outside of the frame away, assembling the object, the final object. This thesis focuses on the virtual part.

Chapter 2

Related Work

In this chapter selected approaches from the domain of stylized fabrication are presented, with focus on methods that use 2-dimensional fabrication to create 3D objects. In the second part of this chapter, we will present selected fabrication approaches that use 3D printing in combination with other fabrication technologies and materials, especially fabric.

2.1 Stylized Fabrication

Stylized fabrication is a subdomain of personal fabrication. It describes fabrication concepts where objects get stylized in certain ways, to use characteristics of a special fabrication technique. It is used to increase efficiency or to achieve artistic effects (Bickel et al. [2018]).

It is feasible to create 3D objects by assembling 2D objects

There are exotic approaches that use light and shadow, inflation or *Lego* bricks (Mueller et al. [2014]) to create shapes. More common methods create 3D objects from flat shapes. For example, by creating the object from connecting flat parts, using pop-ups or folding cut-outs in the desired form (Bickel et al. [2018]). The presented related work gives an overview over existing approaches. *FabricFaces* complements this area.

Figure 2.1: Objects designed in *FlatFitLab* consist of interlocked 2D parts. Image adapted from McCrae et al. [2014].

2.1.1 Connecting Flat Parts

There are several approaches that assemble objects from flat components. *FabricFaces* supplements existing fabrication methods in this domain. They are faster but limited, because they cut the parts from materials like wood.

FlatFitLab creates open and organic shapes by interlocking parts with slits

A popular technique is to interlock parts by using slits to slide them into each other. There are some constrains on how slits and parts can be arranged to make them physically constructible. This was investigated and formalized by Schwartzburg and Pauly [2013]. *FlatFitLab* by McCrae et al. [2014] uses this technique. It "is a comprehensive system for the interactive modeling, simulation and fabrication of planar section assemblies." It allows the user to virtually construct a 3D object (fig. 2.1). The constructability is analyzed and features for interlocking are generated. When the user has designed the object, the generated flat parts can be cut out and assembled. This approach focuses on the object contours and allows to create shapes with organic silhouettes.

Kyub creates closed and cubic shapes by joining flat parts on their edges

A second approach for creating 3D representations from flat parts is *Kyub* by Baudisch et al. [2019]. The user creates 3D objects by merging and deforming cubes. This results in closed surfaces and an intuitive design workflow comparable to building with *Lego*-bricks. It is better suitable for closed industrial shapes, than for organic design (fig. 2.2). The resulting objects are capable of withstanding large forces.

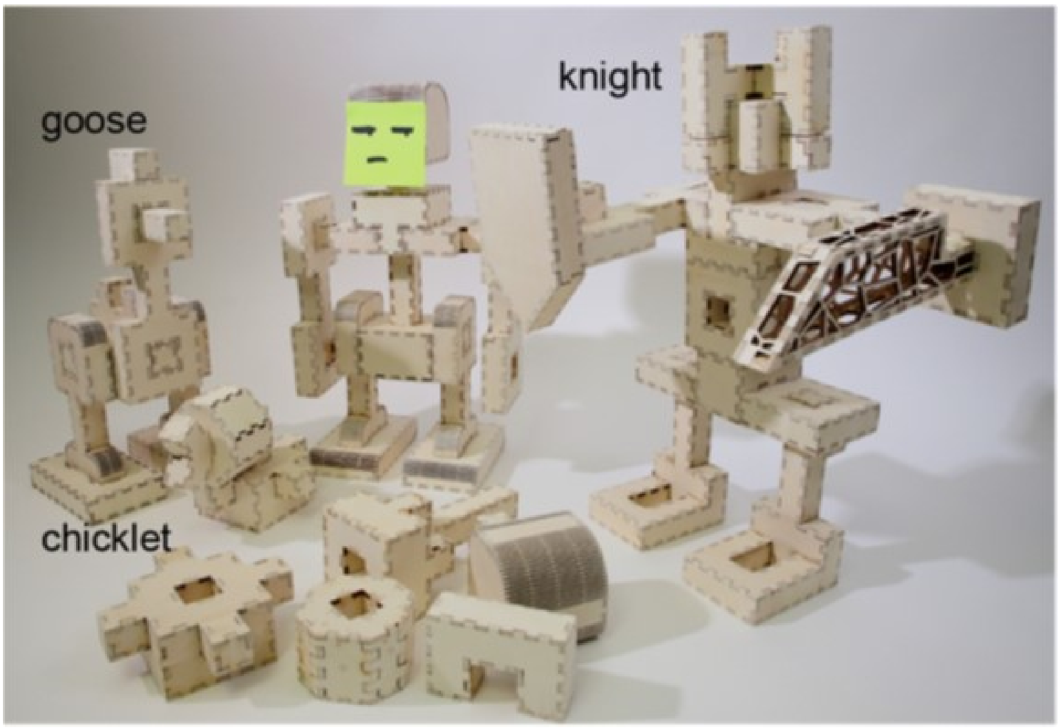

Figure 2.2: Objects in *Kyub* are designed by merging voxels. Cutting patterns are computed to create the walls. Image adapted from Baudisch et al. [2019].

Manufacturing objects by connecting flat parts makes the fabrication often faster but results in low-fidelity objects. There is a range of approaches that aim at combining those techniques with 3D printing to create more detailed objects faster (Beyer et al. [2015], Muntoni et al. [2019b], Mueller et al. [2014]). This is further described in section 2.2.1.

2.1.2 Pop-Ups

Pop-ups are easy to fold and unfold, but not very stable

Pop-up approaches are a special kind of techniques that use a single folding motion to transition between a flat state and a three dimensional state effectively. They are inspired from paper pop-up illustrations, and primary applied for artistic usage (Bickel et al. [2018]). Pop-ups can be made out of paper, and are material efficient. Once they are assembled, the folding and unfolding is quick and easy.

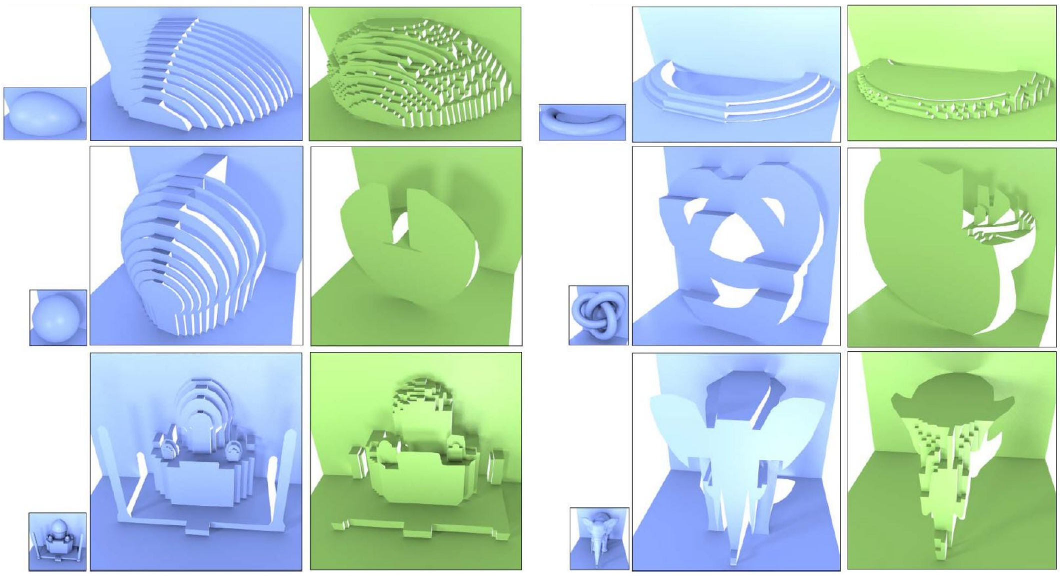

Figure 2.3: A comparison between two 90 degree pop-up techniques. Green: the approach of Li et al. [2010], Blue: the corresponding results by the refined version of Le et al. [2014]. Image adapted from Le et al. [2014], cropped.

Computing visually pleasing pop-ups is hard

It is not easy to find cuts and folding edges for pop-ups to closely represent shapes. Li et al. [2010] presented an approach that focuses on 90 degree pop-ups depicting architectural designs. 90 degree pop-ups are in the opened target state when the two base plates are opened to a 90 degree angle. The output of this method is a mapping of cuts and folding lines on a piece of paper. The result is 2,5 dimensional like a half relief and requires only one piece of paper. This approach was further developed by Le et al. [2014] by using a refined set of geometric conditions. This results in solutions that are closer to designs by real artists, enhancing stability and aesthetic quality of the output (fig. 2.3).A second approach by Ruiz et al. [2014] is more generalized and complex. It creates 180 degree pop-ups and a texture mapping for correctly coloring the output (fig. 2.4). The input object gets abstracted to different primitives, which can then be assembled into a pop-up, combining different techniques pop-up artists use (Ruiz et al. [2014]). This allows to use patches to add details and organic contours to the object that could not be generated otherwise.

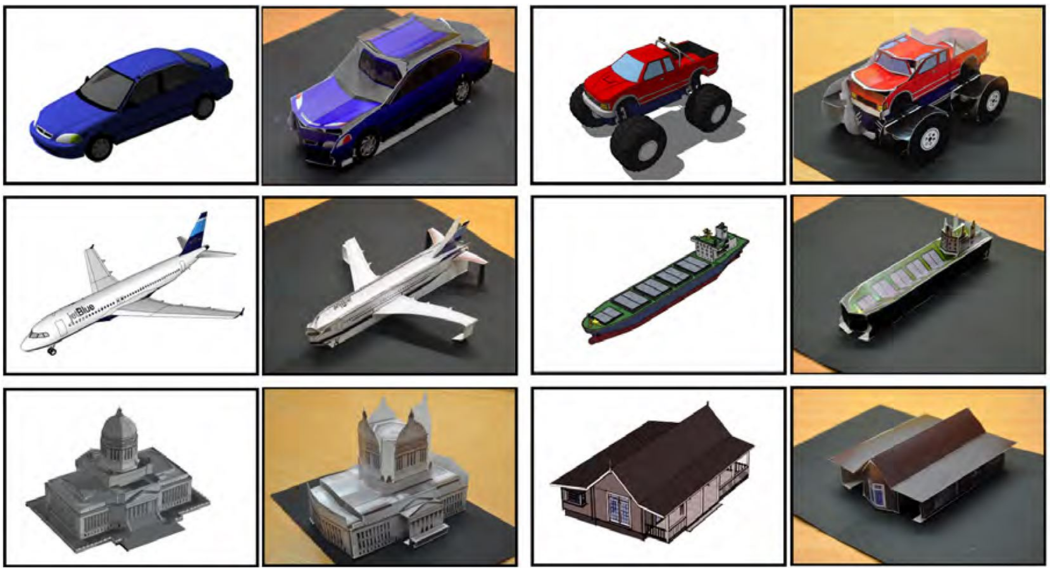

Figure 2.4: Input models (left in pairs) and the corresponding 180 degree pop-ups (right in pairs) with extra patches and textures to add detail. Image adapted from Ruiz et al. [2014].

Pop-ups can be used as tangibles

Pop-ups are not limited to artistic applications. Zhao et al. [2017] proposed a method to create tangible user interfaces by combining pop-ups with conductive ink printing. This enables a fast and cheap way to design and prototype tangibles.

In their flat state, pop-ups are easy to transport, and they can be assembled quickly. *FabricFaces* combines those advantages with a higher stability, by using support structures and connecting features. As a trade of, this also makes the assembly of a *FabricFaces* object more complex than the simple opening motion that is characteristic for pop-ups.

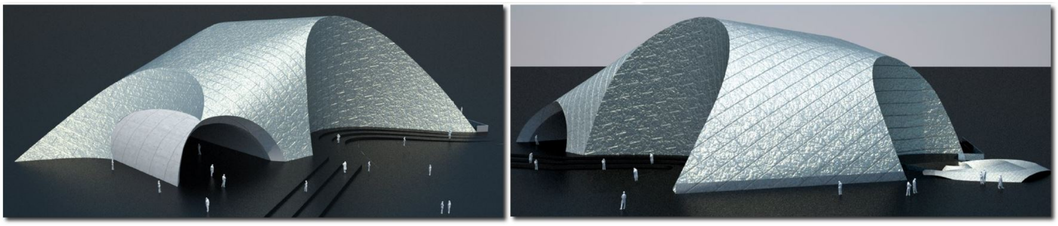

Figure 2.5: Two different examples for curved folding in architectural design. The texture can be mapped on a plane without distortion. Image adapted from Kilian et al. [2008], cropped.

2.1.3 Fabrication by Folding

Folding-based approaches are common in the domain of stylized fabrication (Bickel et al. [2018]). Shen and Nagai [2017] categorized two different ways of creating objects by folding them from planes: rigid folding and curved folding.

Rigid Folding

Rigid folding is simpler than curved folding

Rigid folding refers to techniques where edges of the object function as hinges between planar faces. Curved faces do not occur. While this limits the design space, flat faces allow to use unbendable plane materials. For example origami-like robots can be created with established 2D fabrication techniques by actuating the edge hinges (Onal et al. [2011]).

Curved Folding

Curved folding allows to create complex shapes without cuts

On the other hand curved folding is "more general and complex than *rigid folding*" (Shen and Nagai [2017]), because it allows for curved edges, and thus curved faces in the folded state. It can only be used with bendable materials. It is possible to create fascinating 3 dimensional shapes with this method without cutting on the edges or shearing the material. This allows direct texture mapping and makes it suitable for architectural and industrial design (fig. 2.5). (Kilian et al. [2008])

Curved folding results in complicated folding operations. Kilian et al. [2017] proposed an approach that aims at assisting those complicated folding processes with strings.

Figure 2.6: A large physical Stanford Bunny Structure constructed by quad-edge panels. Image adapted from Akleman et al. [2016].

Mitani [2004] presented a method where they approached curved surfaces computationally with an alternating pattern of narrow triangles, forming paper strips. This allows to create paper craft toys with curved surfaces, that are approximated by simpler rigid folding computations.

When using rigid materials like our 3D printed structures, rigid folding is more feasible. Tachi [2011] presented an approach to consider the thickness of rigid materials in his work about rigid origami, making rigid folding more accessible for architectural design. Akleman et al. [2016] propose a work flow that translates an input model into the quad edge data structure, then translates this into stripes that can be laser cut, folded and connected to create a representation of that object (fig. 2.6). This approach combines rigid folding with curved faces and results in material efficient representations with evenly distributed holes.

Rigid materials work best with rigid folding

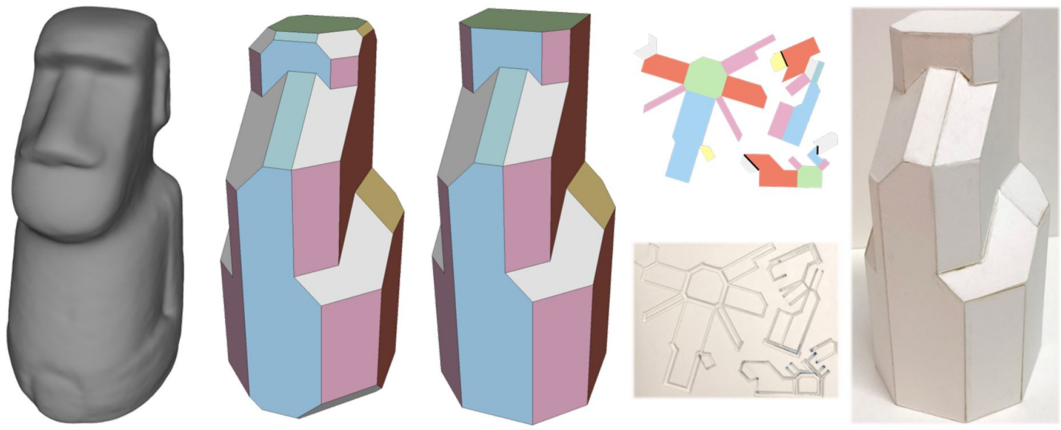

Figure 2.7: An overview over the *Mill and Fold* process. From left to right: the input digital model; the result of the automatic simplification step; the model after the user removed some faces; on top the unfolding plan and on bottom the corresponding carved sheet; the final assembled model manufactured and manually folded and glued (Muntoni et al. [2019a]). Image adapted from Muntoni et al. [2019a].

Rigid folding can be combined with CNC milling

Muntoni et al. [2019a] presented the *Mill and Fold* process (fig. 2.7). It allows to fabricate objects with a computerized numerical control (CNC) mill. Comparable to *Kyub* an object is created by mapping its surfaces to a flat layer. However, instead of designing the object in a special application, this approach uses a given input model and simplifies it. Common CNC mills are limited to create grooves with angles from a fix set, because every angle needs a dedicated milling head. If for example the two faces of an edge should meet in a 90° angle in the assembled object, the edges must be cut out of a plane using the corresponding 45° tool. To ensure a fix set of angles between faces, the *Marching Cubes* algorithm by Lorensen and Cline [1987] is used to simplify the input model. *Mill and Fold* has the advantage that in most cases neighboring faces can be folded to the three dimensional state which helps to assemble the model.

FabricFaces transfers the basic concept of *Mill and Fold* to 3D printing, with the benefit that the angle constraint does not apply and that connecting features can be added.

2.1.4 Stylizing Geometry

Object simplification is an important first step in many stylized fabrication methods. Most of the stylized techniques that create real world objects have constrains, that limit how detailed, curved or angled the manufactured object can be. So, processing the input 3D model can be necessary. For example *Marching Cubes* was used by Muntoni et al. [2019a] because it results in a discrete set of possible face angles.

Simplification is the first step in stylizing

There is a wide range of general simplification methods, that can be used to reduce the degree of detail an object has (Cohen [1999]). It is important to note, that many simplification techniques aim at making a model less computational intense. This does not necessarily result in an easier fabrication process. For example the *Variational Shape Approximation* algorithm by Cohen-Steiner et al. [2004] creates non-uniform faces, but has characteristics that are very fitting for folding based fabrication techniques like our *FabricFaces*. It can lower the face count of an object while trying to keep geometric features. It does this without triangulating the mesh, or doing other changes for computational simplicity that would not help in the manufacturing process.

Models that are easier to compute are not necessarily easier to manufacture

Further simplifying an object becomes a subject of aesthetics and personal preference at some point. To give the user control in this step, Muntoni et al. [2019a] proposed a simplification approach to eliminate faces on already simplified objects. If possible, it deletes the selected faces and enlarges the neighboring faces until they meet. This allows the user to manually further simplify the object where it fits the aesthetic requirements and make it easier to assemble.

Stylizing is also a subject of personal taste

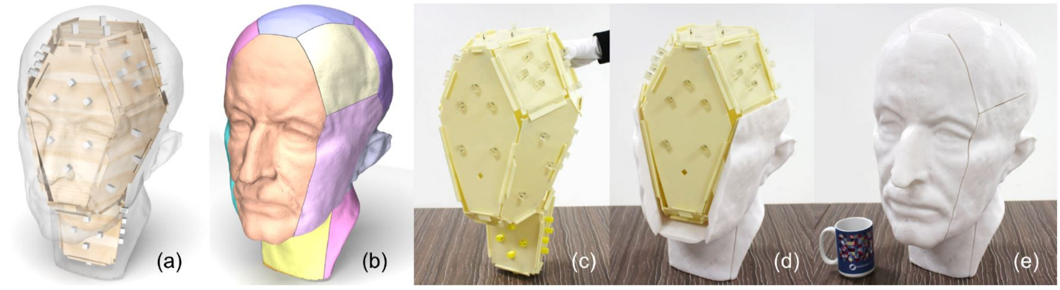

Figure 2.8: In the *CofiFab* process the core is assembled from milled parts and the outer shell is assembled from printed parts, to increase speed while keeping detail. Image adapted from Song et al. [2016].

2.2 3D Prints Combined with Other Materials

Including other materials to 3D printing to make it faster or better

It is possible to influence the 3D printing process by combining it with other materials. This section describes two different approaches to that. The first approach is about selectively printing parts of the object and using another method for a support structure. The most important reason to do this is to increase the iteration speed. The second approach is about embedding other materials in the print during the printing process to alter the physical behavior of the resulting object.

2.2.1 Support Structures

faBrickator combines 3D printing and *Lego* bricks

An easy way to create a support structure for a 3D print is to built it from *Lego* bricks. *faBrickator* by Mueller et al. [2014] works like this. It is aimed at rapid prototyping and allows to specify high and low resolution areas. For low resolution areas a *Lego* representation is generated, while high resolution areas are 3D printed and equipped with connecting features to attach them to the *Lego* parts.

CofiFab combines laser cutting and 3D printing

Cutting the support structure from flat materials like plywood is another possibility to produce stable results. Song et al. [2016] proposed a method for this called *CofiFab*. It is a coarse-to-fine 3D fabrication method. Objects are fabricated by constructing a hollow core volume from flat laser cut pieces, which are surrounded by an outer shell of multiple 3D printed parts to obtain the surface details (fig. 2.8).

An approach that goes even further is *Split and Mill* by Muntoni et al. [2019b]. It completely skips the 3D printer and splits the object into relief-like pieces and produces those pieces with a CNC mill.

2.2.2 Embedding Fabric

There are many possibilities to add characteristics to a 3D printed object, for example by adding copper structures (Shemelya et al. [2015]), optic parts (Willis et al. [2012]) or fabric (Rivera et al. [2017]). We will focus on an approach that uses fabric.

The properties of fabric and 3D prints can be combined

Rivera et al. [2017] investigated the suitability of fabric to be used in 3D printing contexts. They focused at embedding fabric inside of printed objects and at printing on fabric, to enable complex functional behaviors. 3D printing can be used to alter the bending behavior of fabric or to add elements to it for decorative or functional reasons. Including fabric into 3D printed objects can be used to build sensors or to close surfaces.

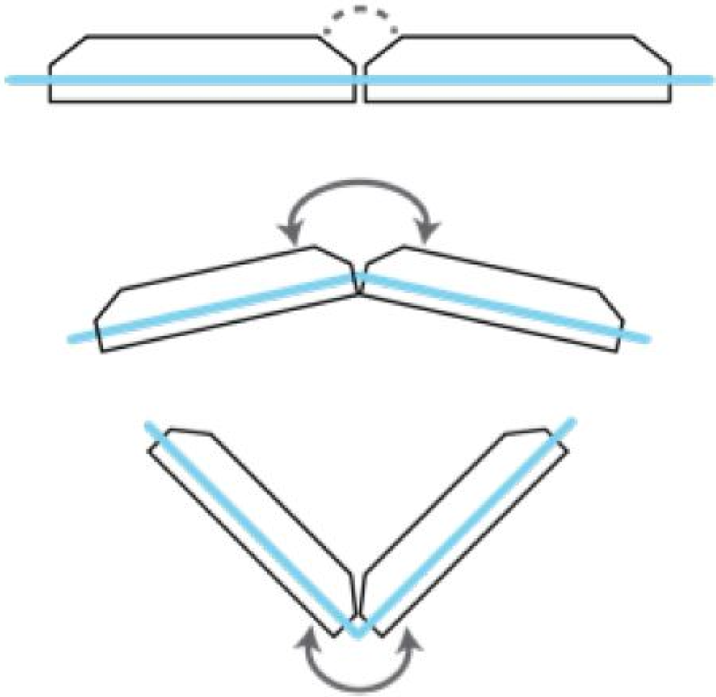

Figure 2.9: The bending properties of fabric can be controlled by 3D printing on it. Image adapted from Rivera et al. [2017].

Bending

The bending behavior of textiles can be altered by 3D printing on it

By embedding fabric in two rigid parts with a gap between them, the resulting object can be bend exactly along the axis of the gap. The angular range in which bending is possible can be constrained by applying angled cut outs to the rigid part (fig. 2.9). The authors presented an example for using the fabric as hinges between the faces of an unfolded dodecahedron. It can be assembled by folding it to the 3 dimensional state and fixing the edges (Rivera et al. [2017]).

Figure 2.10: 3D printed fabric connectors. Image adapted from Rivera et al. [2017].

Adding Elements

3D printing elements on fabric allows to create soft structures with defined rigid features. Holes in the fabric can be stabilized with a rim and features for connecting different parts of fabric can be generated (fig. 2.10). It is possible to create a strong connection between the printing material and the fabric (Rivera et al. [2017]).

3D printed features can reversibly connect textile layers

Sensors

By adding elastic fabric inside a 3D printed object sensors can be build. This allows parts of the object to move and obstruct a light sensor. This way turning nobs, switches or sliders (fig. 2.11) can be build (Rivera et al. [2017]). More advanced sensors are possible when also using electric embroidery. *FabriClick* by Goudswaard et al. [2020] uses electric embroidery and 3D printing to apply integrated touch buttons to clothing.

Complex functional behavior is possible by combining 3D printing, textiles and electric embroidery

Closed Surfaces

Closed surfaces can be generated by printing frames on the fabric and cutting them out (fig. 2.12). This allows to use the characteristics of fabric (visual texture, tactile texture, transparency) for 3D models (Rivera et al. [2017]). This application is important for the *FabricFaces* approach.

Fabric surfaces for 3D printed structures are possible

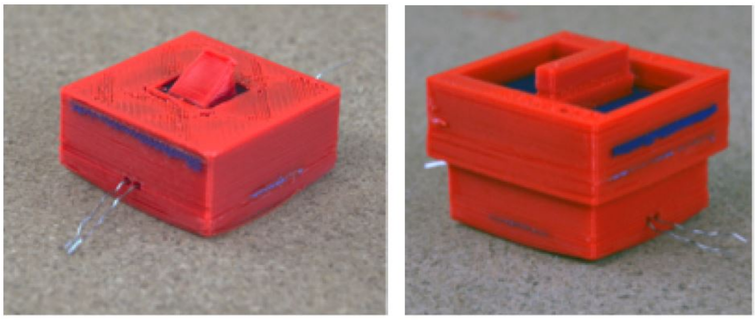

Figure 2.11: 3D printed sensors using fabric. Switch (left) and Slider (right). In the 3D print (red), a light sensor and a light source are embedded. An elastic piece of fabric (blue) is used to suspend an obstructing piece. The piece is moved by moving the switch or the slider. Image adapted from Rivera et al. [2017].

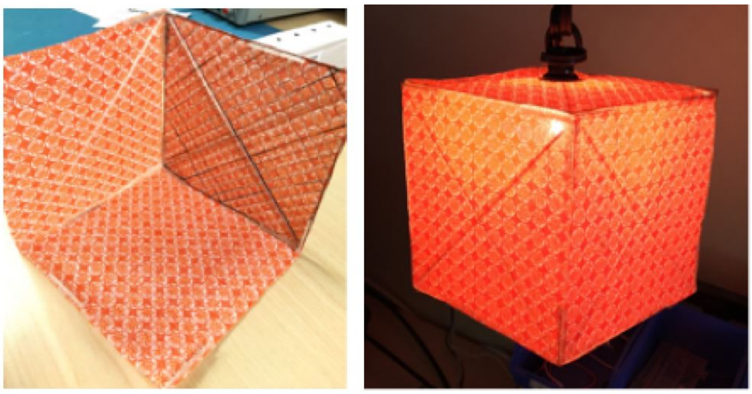

Figure 2.12: Fabric used as the surface material of an 3D printed object. Every face is printed separately and then glued together. Image adapted from Rivera et al. [2017].

2.3 Context to *FabricFaces*

FabricFaces complements the methods for stylized fabrication that already exist

The presented related work gives an overview over existing approaches for stylized fabrication, where flat parts are assembled to 3D models in different ways. *FabricFaces* complements this area. It combines *Kyubs* concept of closed surface models with connectors, with the concept of *Mill and Folds* unfold patterns that get generated from input meshes.

FabricFaces adapts these ideas to 3D printing, a well accessible technology, which also eliminates the angle constraint *Mill and Fold* has. By generating reversible connectors the objects should be easy to assemble (from flat to object) and to disassemble (from object to flat), comparable to pop-up structures. By using fabric for the surface, a new realm of design possibilities is opened. The basic requirements for this are already investigated by Rivera et al. [2017], showing that combining textiles with 3D printing can be done in the way that is needed for our approach. This should also speed the process up, making *FabricFaces* a supplementary method to *faBrickator* or *CofiFab*. Interesting future additions to *FabrickFaces* are inspired by the mentioned related work and are presented in section 5.2.

Chapter 3

FabricFaces Add-on

For the *FabricFaces* approach a support structure must be created. It consists of different segments, each for one object face (fig. 3.11).

FRAME:
In the context of *FabricFaces*, a frame is the structure that supports the objects textile surface (fig. 1.1). The frame gets 3D printed on a layer of fabric, while being in an unfolded, flat state. Once this is done, it can be folded to assemble the final object (fig. 1.2). While the fabric creates the objects surface, the frames purpose is to stabilize it. Frames have connectors on their edges to increase stability.

Prior to the 3D printing, the required frame has to be designed. For this, the desired object has to be virtually unfolded, which may require cutting it into several parts (islands). The resulting unfolding is than used to create the frame with edges and connecting features. For each island a frame is generated. Performing this process manually in CAD-software is a difficult and time consuming task (Mattiussi [2020]).

To design the frames, the object gets virtually unfolded

ISLANDS:
The parts that an object decomposes to when it gets unfolded are called islands. Simple or convex objects can be unfolded as one island.

We create a *Blender* tool for accessible frame design

The objective of this thesis is to make *FabricFaces* more accessible by developing a solution for the frame design. We implemented a tool that automates the process. It takes a predefined virtual geometry and computes a corresponding frame. Users can influence the results by adjusting values like the thickness of the frame, defining the connectors, and changing the suggested unfolding. Our tool is an add-on for *Blender*, written in *Python*. As *Python* scripting is deeply integrated into *Blender*, this allows an efficient integration.

3.1 Application

In this section an example usage of the add-on is described. We begin with the installation. Next, we describe a normal workflow for unfolding the object, adjusting the frame and exporting the result.

3D printing and physically assembling the object are out of scope of this thesis and will therefore not be described in detail. An example print is shown in appendix B.

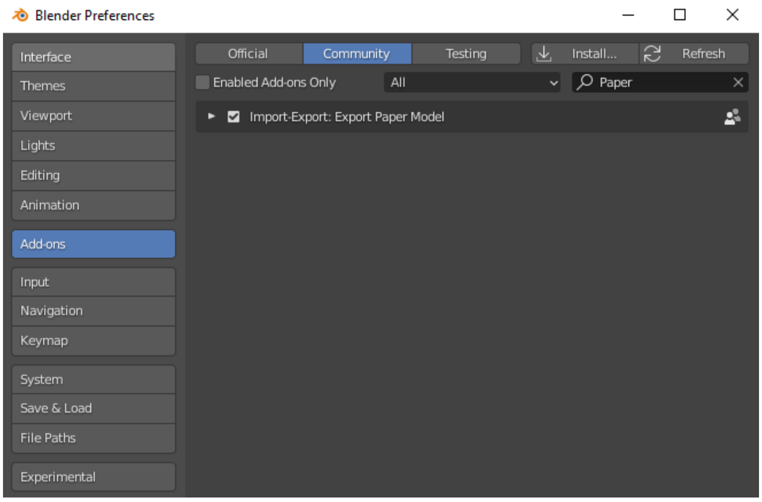

Figure 3.1: Before installing the *FabricFaces* add-on, the *Paper Export Model* add-on needs to be activated. The installation is also done in this dialogue.

3.1.1 Setup and Installation

The *FabricFaces* add-on uses the *Paper Export* add-on. This is a community add-on that is included in *Blender*, but it needs to be activated. This is done by navigating to *Edit, Preferences, Add-ons, Community*. Search for "Export Paper Model" and activate it by ticking the checkbox (fig. 3.25). Next, the *FabricFaces* add-on is installed in the same dialogue by clicking the "Import"-button, selecting the "main.py"-file of the add-on and clicking *Install add-on*. This add-on must also be activated by ticking its checkbox.

The Paper Export add-on must be activated and the *FabricFaces* add-on installed

3.1.2 First Step

The *FabriFcaces* user interface (UI) is located at the *Blender* tool shelf. This is a panel at the top right corner of the 3D viewport. It can be opened and closed by pressing 'N'. A new tab labeled "FabricFaces" is created in this area. Users might open the tab and see that everything is grayed out and a small warning is shown: "Select an object". This

The add-on UI is located at the N-panel

warning is shown, because it is not specified which object should be processed. As soon as the users create and select an object, the warning disappears and the UI elements becomes active (fig. 4.9).

3.1.3 Regions Preview

The region view color-codes the different cutting islands on the object

The first thing users may want to do, is to find out how the object will be unfolded. They can hover over the button "Toggle Region-View". The appearing tooltip says "Generate a preview of the cutting". When pressing the button, the selected objects faces get marked in different colors for each region that the cutting resulted in. In this case it were five regions, so five different colors are shown on areas of neighboring faces on the object. In *Blenders* info-panel the message "There are 5 cutting regions" is printed.

The perspective is adjusted

The camera view and its rotation center-point are set to the object. This allows users to inspect the result easily. Users can change the color-scheme by turning the region-view off and on again. The colors of cutting regions with neighboring indices will always be clearly different and each region has a unique color.

3.1.4 Suggested Parameters

Suggested parameter values can be computed

Next, users likely want to get the suggested settings for the object. They load them by clicking the "Suggest Settings"-button (fig. 3.2). The add-on computes and enters suggested parameter values based on the objects average edge length. The name of every parameter that was changed is marked with an asterisk. The buttons label changes to "Reload Suggestion". If another object would be selected, the buttons name would change back to "Suggest Settings". This allows users to see if the settings they are currently tweaking are based on the currently selected object.

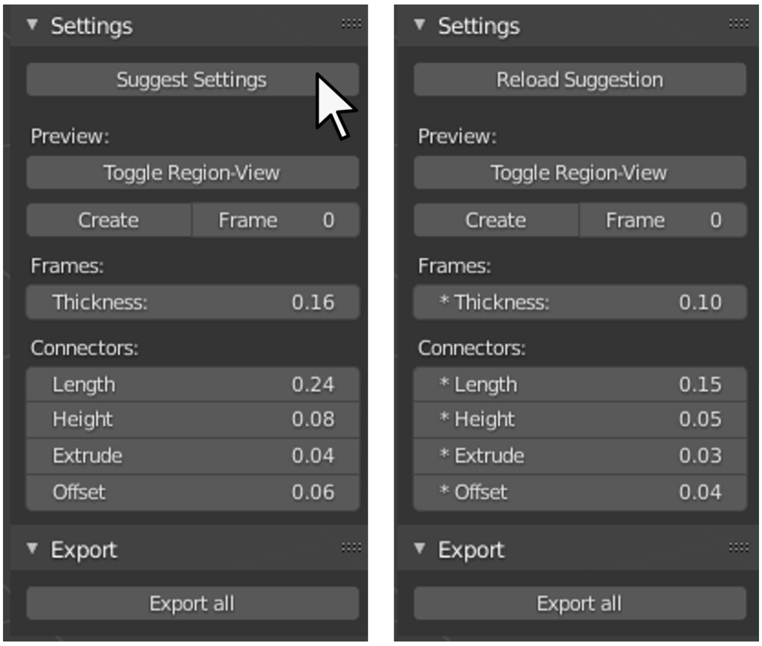

Figure 3.2: Left: The UI before loading the suggested settings the first time. Right: After loading the suggested settings, the computed values are entered, marked with asterisks if they changed, and the buttons label changes to "Reload Suggestion" as long as this object is selected.

3.1.5 Creating a Frame

To see how the suggested settings work, the frame must be generated. This is triggered by clicking on the "Create"-button. The frame for cutting region 0 is generated (fig. 3.3).

The frames get shown one at a time

The cutting regions are ordered by their face count, so 0 is the one with the most faces. Frames are always colored in a light blue. All of the marked settings get unmarked, because these values are now used in the currently shown frame. As long as a frame exists, the button is labeled "Refresh" to empathize that it is not necessary to delete the currently shown frame manually. The cameras view and anchor point are set to the generated frame.

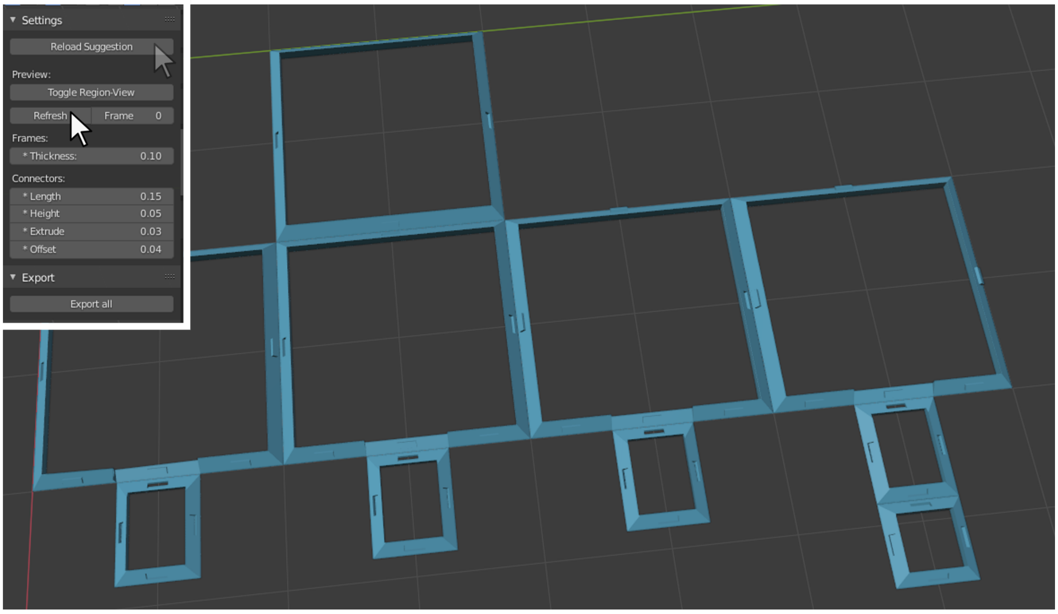

Figure 3.3: Creating frame 0 after loading the suggested settings.

Users could now enter different numbers in the "Frame" field or increment/decrease them using the arrow buttons. In this case, the range of valid frame numbers is 0 to 4. If a user enters a 5 (or larger) and refreshes, the number jumps back to 0.

3.1.6 Adjusting Parameters and Blueprints

Asterisks mark values with unapplied changes

Users may want to make the frame thicker, so they increase the thickness value and refresh the frame (fig. 3.4). Between the value change and the refreshing, the parameter name is marked with an asterisk.

Warnings notify about possibly undesired effects

Some users might decide now, that the connectors are too small, so they increase the "Length" and "Height" values and refresh again (fig. 3.5). In some cases, a warning might be shown in the "Info" - panel: "The connectors will peak through the fabric when you fold it. Consider smaller connectors."

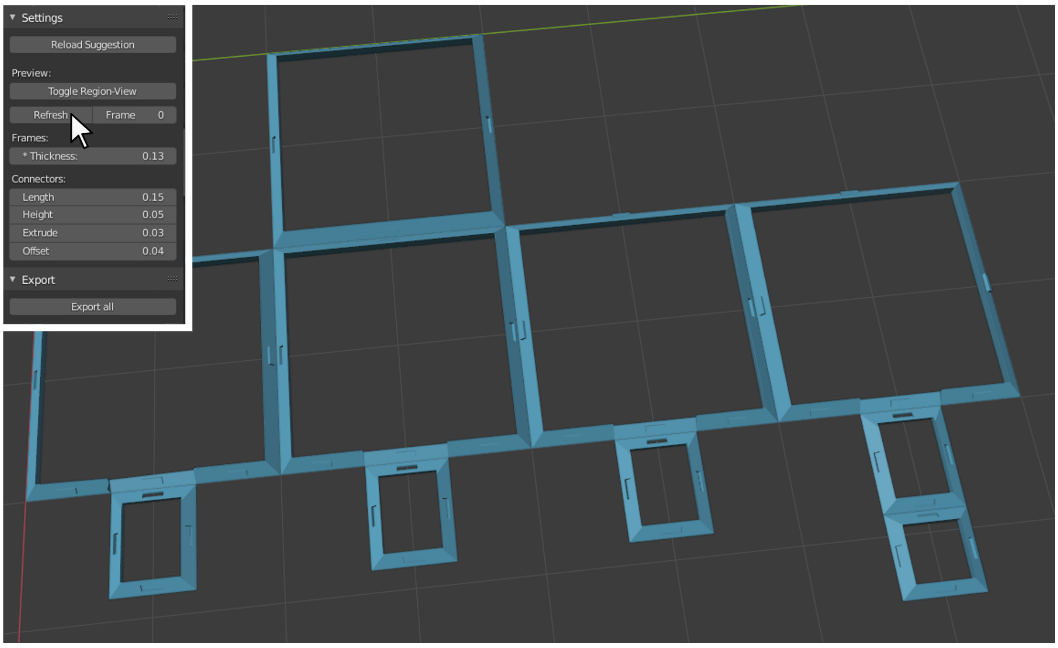

Figure 3.4: Increasing the thickness parameter and refreshing.

This tells the users, that if they would print this frame and fold it together, parts of the connectors would stand out from the surface of the objects faces.

If the users want to keep the larger connectors, they might decide to increase the offset and refresh (fig. 3.6). Possibly the offset was increased too much. In this case another warning would appear: ”You might have floating connectors.” By reducing the offset again, this could be solved (fig. 3.7).

Next, users might decide to change the folding-edge connectors to make assembly easier (fig. 3.8). This is done by selecting the red connector blueprint, switching *Blender* to edit mode and decreasing the size of the top face.

Connecting features can be designed

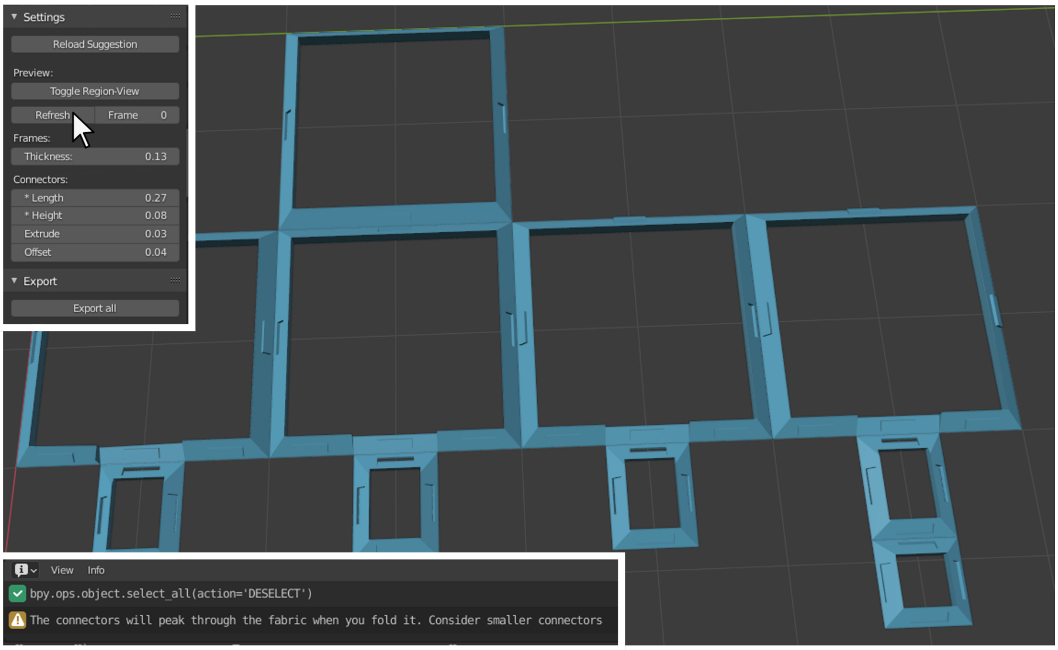

Figure 3.5: Increasing the connector size and refreshing. The *connector peak* warning is triggered.

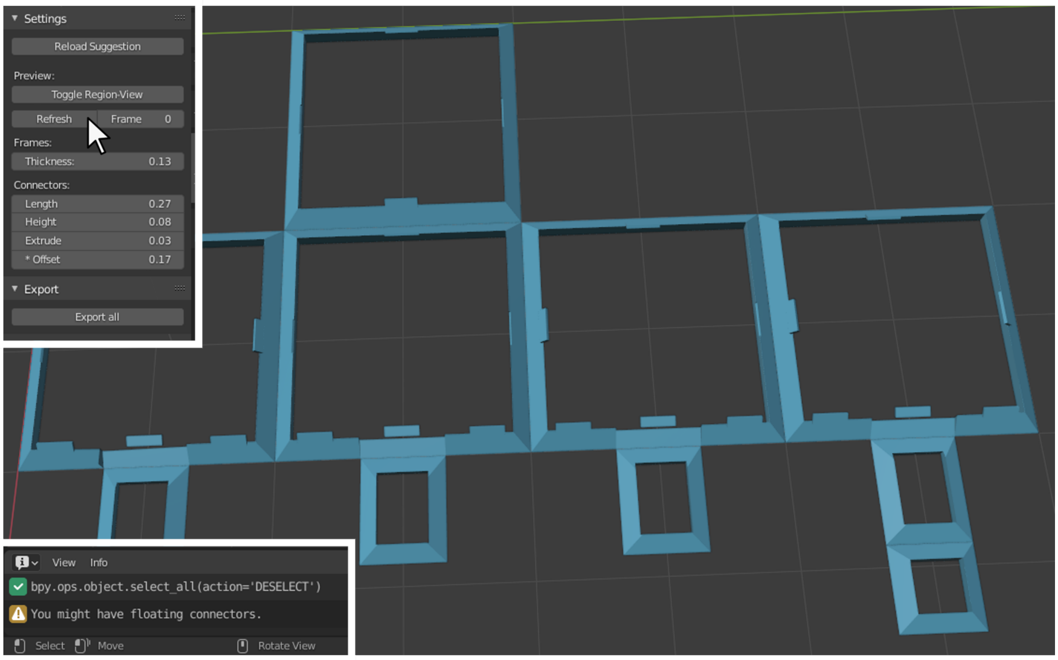

Figure 3.6: Increasing the offset and refreshing. The *floating connector* warning is triggered.

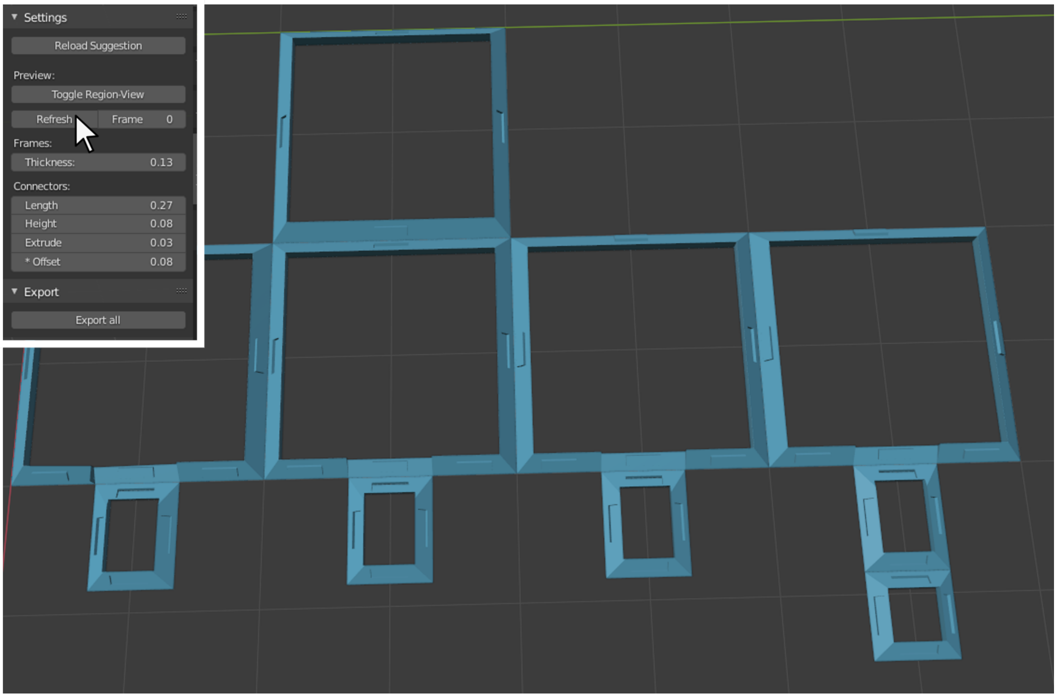

Figure 3.7: Decreasing the offset and refreshing. No more warnings.

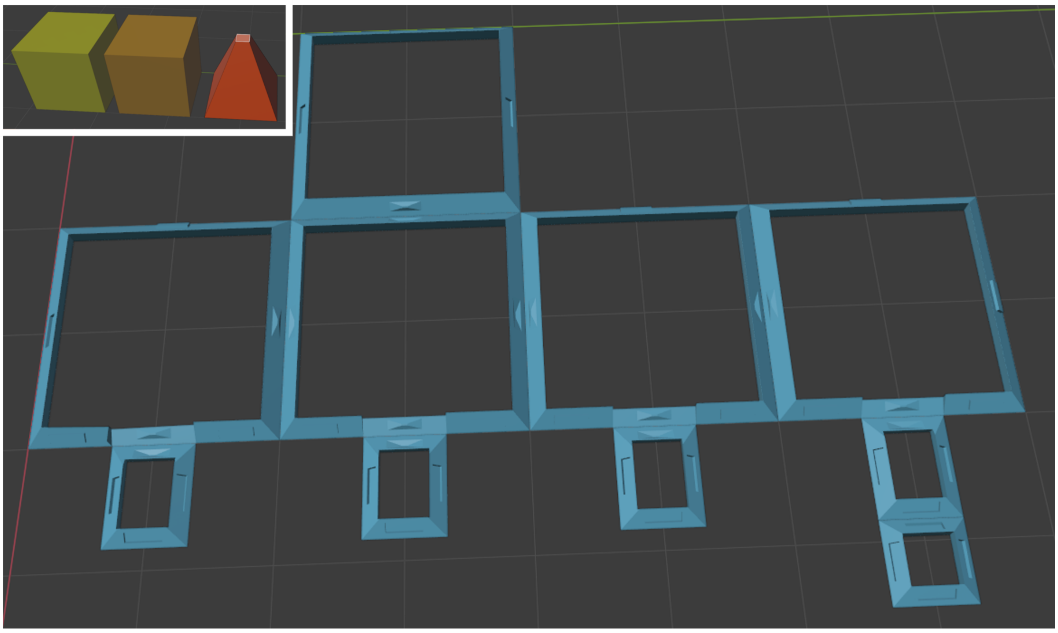

Figure 3.8: Adjusting connector blueprints on folding edges, for example to make folding easier. Refreshing.

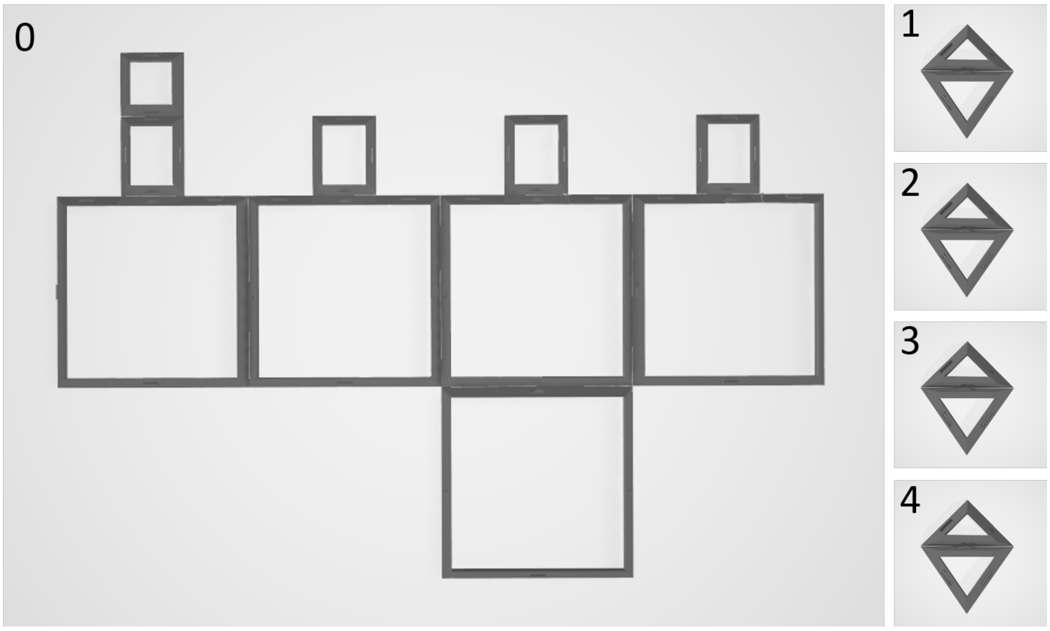

Figure 3.9: The 5 frames that got generated and exported as individual files.

3.1.7 Export

Additional helpful information can be exported when needed

When the parameters are tweaked the users export everything. For that the standard export dialogue from *Blender* is used. It is expanded by the three export settings. In this example the export contains 21 files. 5 frame .stl files (fig. 3.9). 12 concave connector .stl files (one for each concave edge). One .stl file of the complete object for reference and one .obj file that represents the region view (with .mtl file for the color information). And the text file that contains the settings. An overview about the export and the resulting files is presented in figure 3.10.

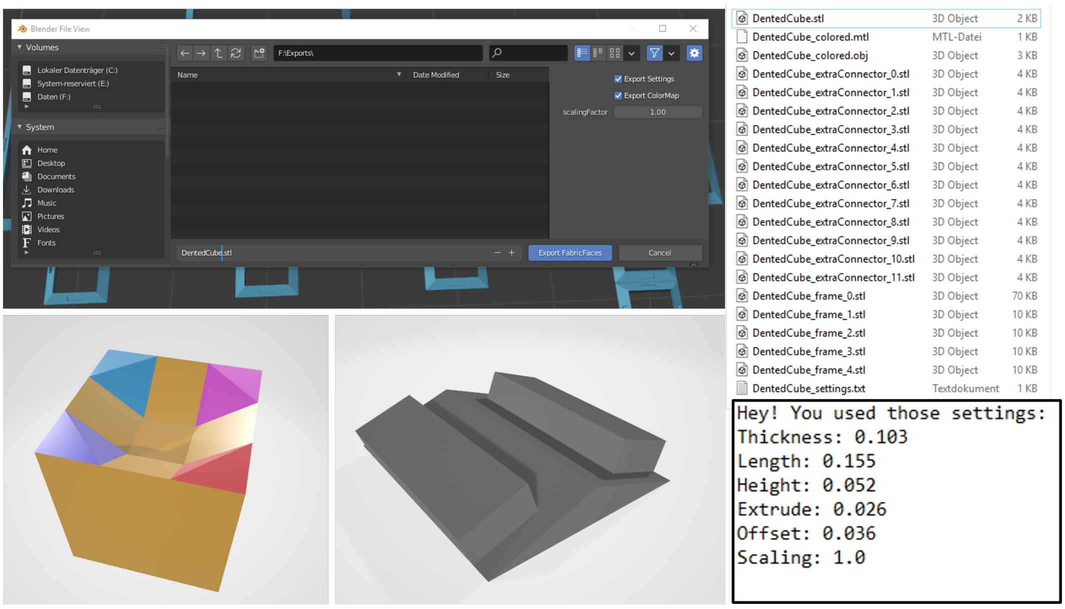

Figure 3.10: Top left: The export dialogue, containing the export options. Top right: List of the exported files. Bottom left: The "colored.obj" file that represents the region view. Bottom middle: One of the 12 identical concave extra connectors. Bottom right: The content of the "settings.txt" file.

3.2 Implementation

After we described the workflow of the add-on from a user perspective in the previous section, we will now explain how it works.

First, the overall process and the main functionalities of the add-on will be described. After that, the details for each major processing step will be explained. Finally, an overview over general usability features is given.

3.2.1 Processing Overview

Once the add-on is triggered, it checks some general prerequisites. In case something critical is detected, it aborts and provides the user with a short notification and proposes a solution. If the input meets the criteria, the add-on continues the process by computing the unfolding data.

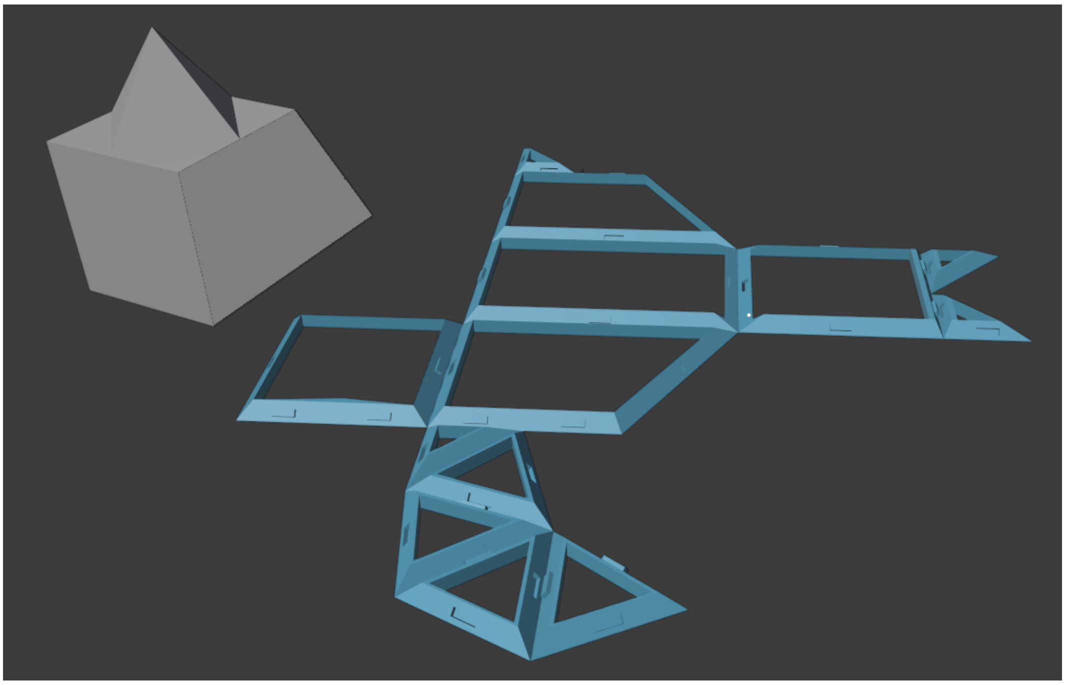

Figure 3.11: *Blender*: An input mesh (left) and the generated frame (right). In this example the unfolding resulted in one island, so only one frame is needed.

In this step, the add-on generates the information for every edge of the object, by determining if it will be cut or just folded. It then maps the vertices to the unfolded two dimensional state. This will result in one or more islands.

The frames have beveled edges and connectors to allow a correct assembly

When the unfolding is finished, the information is used to generate the frames. The frames are flat support structures with a specified thickness, that will support the object in the folded state (fig. 3.11). They span the fabric on the edges. Those edges are beveled according to the angles where the faces will meet and they are equipped with connectors.

3.2.2 Checking the Prerequisites

In order to prevent the generation of invalid output, the program first tests applicability. For convenience, problems will be fixed silently whenever possible (section 3.2.9). If necessary, the user is informed about problems to ease solution development.

Detected problems are fixed silently if possible

If the problem might not be grave, the add-on notifies, but generates the result anyway so that the user can decide. Notifications get collected and presented to the user. They aim at helping to solve the problem. An overview over the warnings is given in table 3.1.

Warnings notify about potential problems that were not fixed

3.2.3 Cutting and Unfolding

The add-on uses the integrated "Paper Export" *Blender* add-on by *Addam Dominec*[1] for the unfolding process. It is used to create paper models that represent a given low poly object and allows to map a 3D model to one or more two dimensional parts. It fits well into this use case, because it produces UV-maps with no distortion and aligns faces next to each other if possible.

We use another add-on for unfolding objects

UV-MAP:
A UV-map is a mapping of three dimensional coordinates of object vertices to a plane. UV is not an abbreviation, it is used to describe two dimensional coordinates like XYZ is used for three dimensional ones.

The *Paper Export* add-on defines the edges that should be cut as *split edges*. They are marked in red when the object is viewed in edit mode. The user can also manually specify lines on where cuttings should be done (fig. 3.12).

[1] https://github.com/addam/Export-Paper-Model-from-Blender

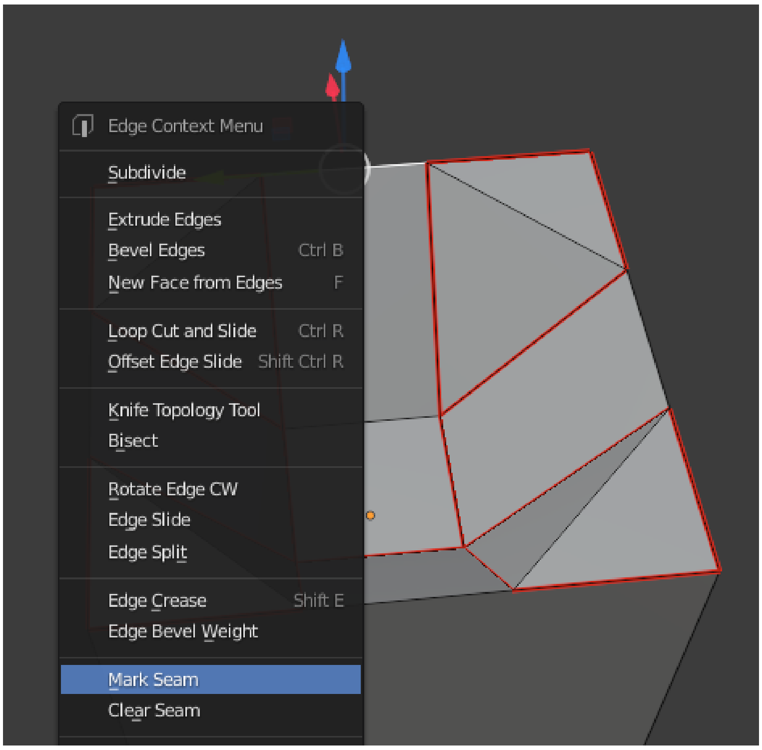

Figure 3.12: *Blender*: Cutting edges of an object are marked red in "Edit Mode". Cutting edges can be set and cleared manually by using the "Mark/Clear Seam" options.

Our implementation unfolds the object and analyzes one island at a time. The output of the unfolding operation is ordered by the islands face count. Per default the first island-frame gets shown, because it is likely to be the most representative one.

By using another add-on we freed resources for specialized features

Using the "Paper Export" unfolding method limits the possibilities to customize the unfolding behavior. This is accepted, as it drastically decreases the development time for this step, allowing to spend more time on the overall process. The possibility of user driven changes is given, so all unfoldings can be manually achieved by experienced users.

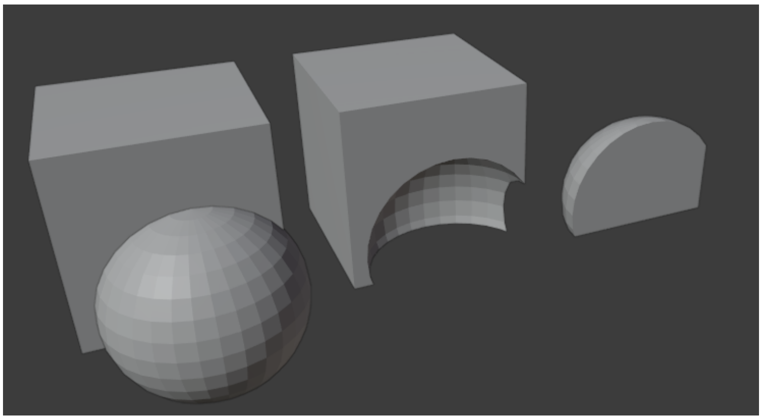

Figure 3.13: *Blender*: A cube modified by a hidden sphere using the three different boolean operator modes. From left to right: *Union*, *Subtract* and *Intersect*.

3.2.4 Generating the Frame

When the two dimensional unfolding information for an island is derived successfully, it is used to create the model for the physical frame that can be 3D printed on fabric afterwards.

Combining Shapes

The frame is generated by combining different shapes with *Blenders boolean operator*, which behaves similarly to constructive solid geometry (CSG). This allows to create the complex frames step by step. *Blender* also offers these three basic operations (fig. 3.13).

We combine basic shapes with boolean operators

CONSTRUCTIVE SOLID GEOMETRY:
Constructive Solid Geometry (CSG) uses geometric boolean operators that use two input shapes to construct a new output shape. The three basic combination modes are union, intersection and difference. Combinations and concatenations of those three objects can be used to create complex shapes (Laidlaw et al. [1986]).

Boolean operators were recently enhanced in *Blender*

In *Blender* 2.91.0, a new implementation of the boolean modifiers was added to *Blender*.[2] That is the reason why we chose this version for our add-on. *Blender* now includes a precise mode, that allows to handle situations where two objects have faces that lay in the same plane and overlap. This special edge case led to bad artifacts in earlier versions.

Boolean operators need correct face normals

It is important to note, that *Blenders* boolean operator uses the face normals of the given shape for computing the result. Combining objects where one or more faces have wrong normals, results in mesh-glitches. Solving them can become rather time and sanity consuming. Turning on the face normal markers in *Blender* is extremely helpful to find and solve this kind of anomalies.

Creating Objects
To create objects, "face lists" are used as internal object representations.

FACE LISTS:
A face list is a simple data structure for representing a 3D object. It is a list of faces that together makeup the object. Each face is a list of its points, and each point is a vector of the three coordinates the point has.

Face lists allow a very quick merging of objects

While this is not optimal regarding storage size and editability, it has some advantages in important domains: It is easy to construct and to iterate, faces are not connected and it allows super quick union operations. One can merge two objects in this data format just by appending the two lists. Of course this results in a "dirty" merging, where it is possible for faces to overlap and cross through each other. But this is not a problem, as long as the unified objects do not overlap, or if the resulting object is just an intermediate step that is not part of the final output.

[2] `https://wiki.blender.org/wiki/Reference/Release_Notes/2.91/Modeling`

Simple shapes can be generated by computing the coordinates of their vertices, and returning the corresponding face list. Additionally, we implemented a method for deriving the face list of an object after it has been modified, and a method for creating an object in *Blender* from a face list.

Faces of an object that had been created from a face list are not connected at their edges. A disconnection can be useful in some cases, but in most cases a connection would be beneficial as it makes manual editing easier. In those cases the *merge by distance* option in *Blender* is applied. It merges vertices that are very near to each other, resulting in connected faces.

Faces are connected, to ease the editing of objects created from face lists

Frame Generation
The frame generation process consists of three major steps (fig. 3.14) that will be explained in the following paragraphs.

Step 1: Basic Frame
The first step is to create the basic frame. This frame is an object that contains the edges of every face of an unfolded island (fig. 3.15). The basic frame is constructed by first creating a two dimensional representation of the unfolded island. Then a second representation is generated, where the edges of the faces are pulled inwards. By subtracting this smaller representation from the larger one, a two dimensional frame is generated (fig. 3.14). It has a fix thickness orthogonal to the edge direction. Finally *Blenders* solidify-modifier is used, to extrude the two dimensional frame by the same thickness to a three dimensional one as shown in figure 3.14.

The basic frame is an extruded two dimensional frame

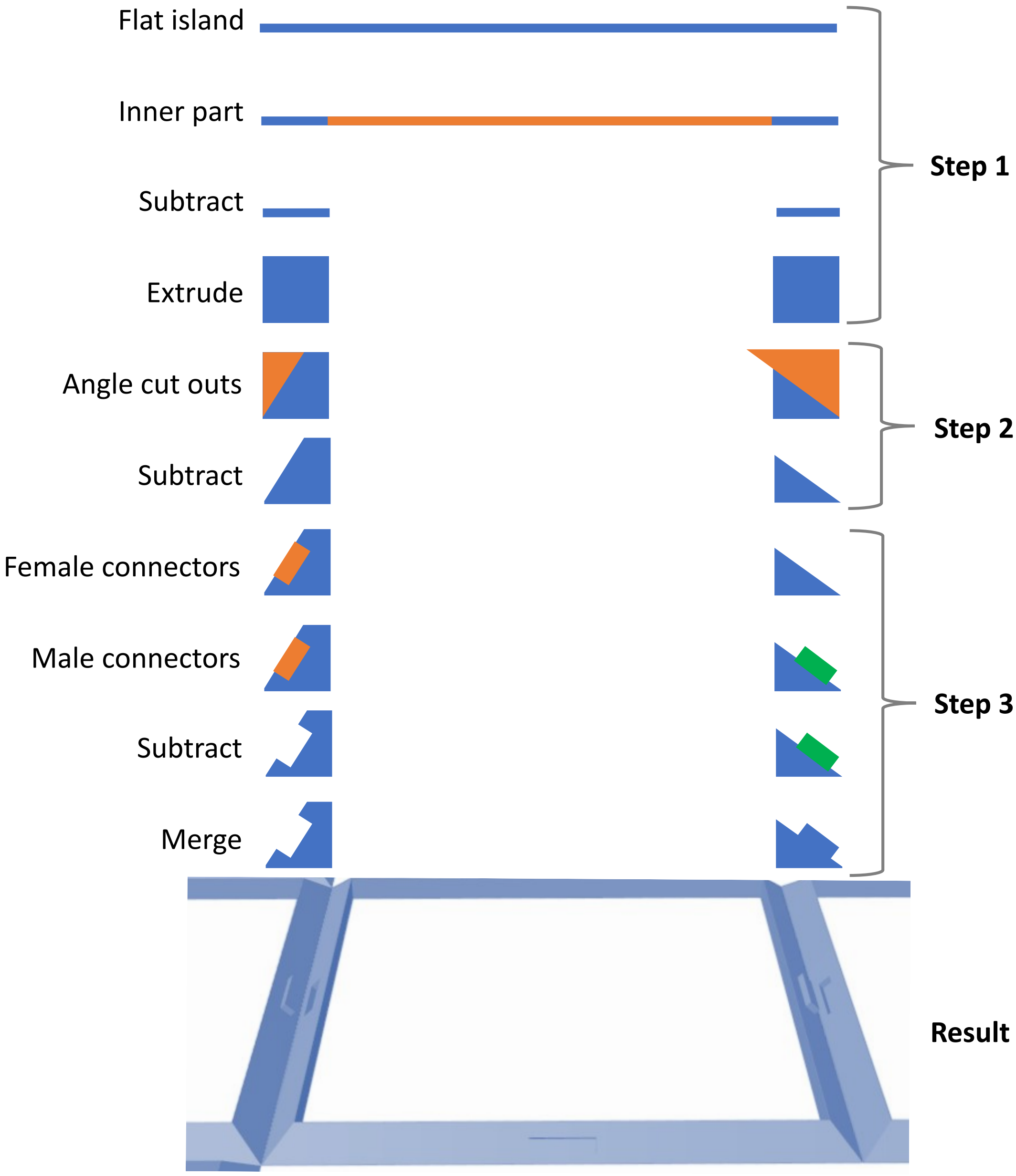

Figure 3.14: Cross-section of one face of a frame, evolving in the frame generation process (Step 1: Basic frame, Step 2: Angled frame, Step 3: Frame with connectors). On the left edge a female connector is generated, on the right side a male connector. Orange parts get subtracted and green parts get merged with the basis (blue).

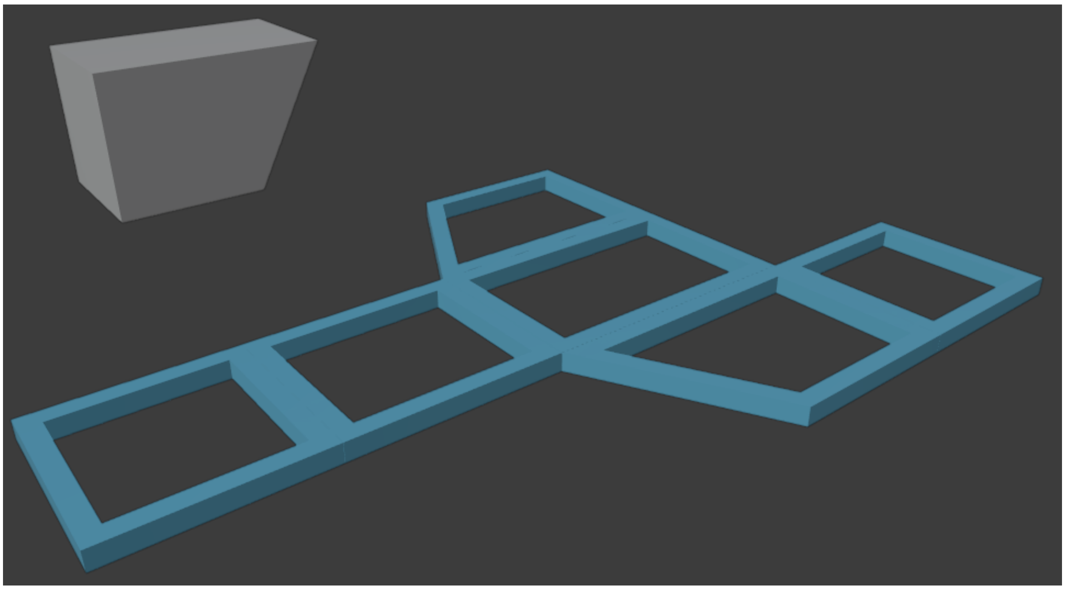

Figure 3.15: *Blender*: Step 1: The basic frame (blue) generated for the input (grey).

Step 2: Angled Frame

The angled frame is a beveled basic frame

The basic frame from step 1 would not be foldable, because it does not consider the angles at which the faces will meet in the folded state. So half the face angle needs to be cut away from the edges of the basic frame. If for example an object edge has a 90 degree angle, an angle of 45 degree is cut away in the two corresponding frame edges. First, a mapping between the faces of the island and the faces of the model is generated and the face angles in the object are measured and stored. Next, the "cut out prisms" are generated (orange triangles in fig. 3.14). They define the space that should be removed from the edge. All the prisms for every edge get generated and merged together with the simple and quick merging method that is made possible by the face lists. They are then subtracted from the basic frame (fig. 3.14). The angled frame (fig. 3.16) is basically functional. It can be printed and folded to represent the given object. However it might be unstable, as the connectors are not added yet.

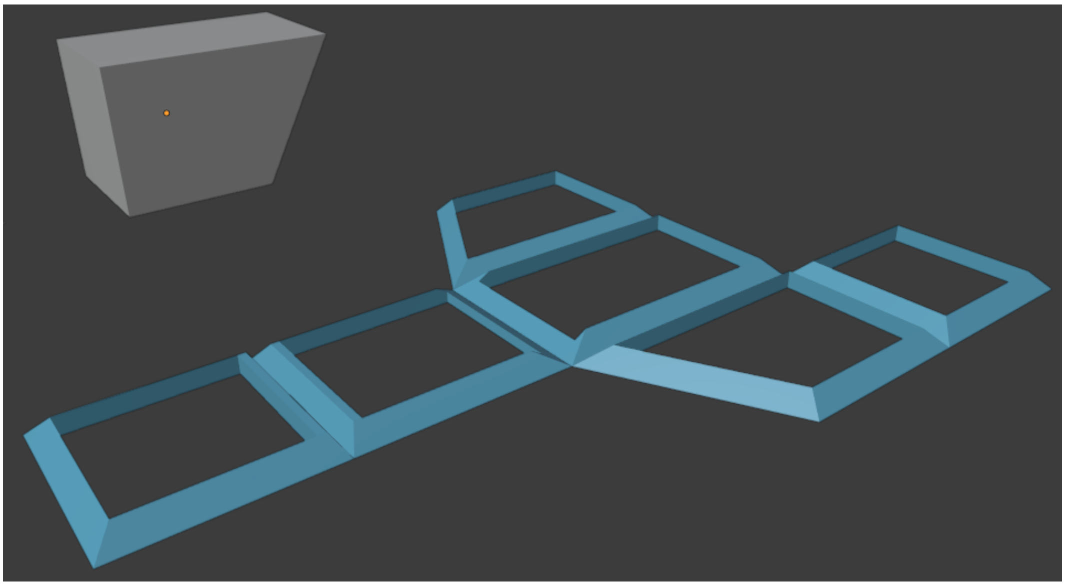

Figure 3.16: *Blender*: Step 2: The angular cut outs are applied to the frame (blue) that is generated for the input (grey).

Step 3: Connector Frame

The connector frame is an angled frame with male and female connectors on the edges

To enable an easier assembly and to increase stability, connectors are added to the frame (fig. 3.17). The add-on distinguishes between four different kinds of object edges:

- Edges where two different islands meet.
- Edges where a cut was made between two faces of the same island.
- Edges that only need to be folded.
- A special case that can occur in all above are concave angles. Those require external connecting parts.

The add-on computes this information to determine what connectors to create on each edge (fig. 3.20).

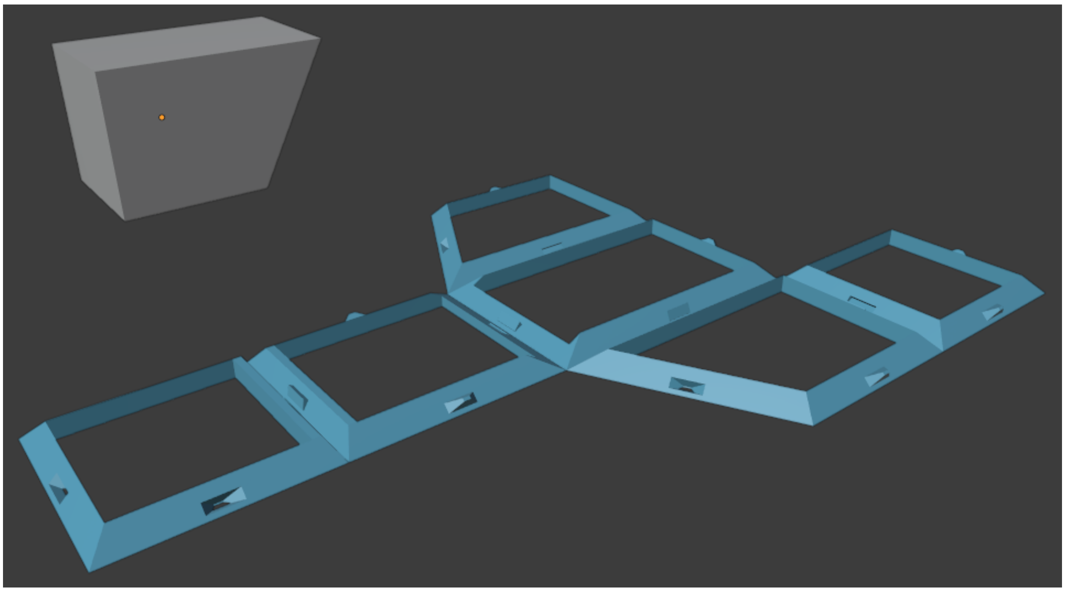

Figure 3.17: *Blender*: Step 3: The connecting features are applied to the frame (blue) that is generated for the input (grey). This is the final output for one cutting island.

The user can edit the connector blueprints in *Blender*. It is possible to specify connectors for every type of edge. Simple cubic connector blueprints are generated as a starting point for the users design process (fig. 3.18). The connector base objects are color coded, to make it easier to tell them apart. The user can edit every type of connector independently (fig. 3.19). The male connectors are merged with and the female connectors are subtracted from the frame (fig. 3.14).

Connectors for a type of edge can be designed by editing one of three connector blueprint cubes

3.2.5 Parameters

The user has access to four parameters in the UI to adjust the frame (fig. 3.21). Default values provide a good starting point.

Suggested Settings
For an object, a setting suggestion can be computed and the suggested value for each of the four parameters is entered. Each parameter has a default value that has been picked to work well with a 2x2x2 *Blender*-units large cube.

The suggested settings are based on the average edge length

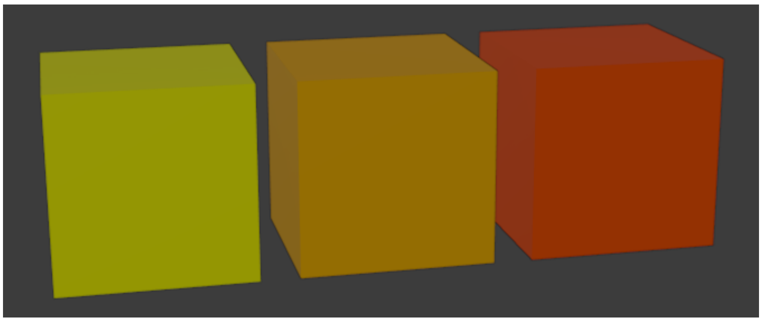

Figure 3.18: *Blender*: The three cubic default connector blueprints. Red: Folding edges. Orange: Cutting edges, where two faces of the same part touch. Yellow: Cutting edges, where two faces of different parts touch. Each blueprint can be changed to change the connectors on the corresponding edges.

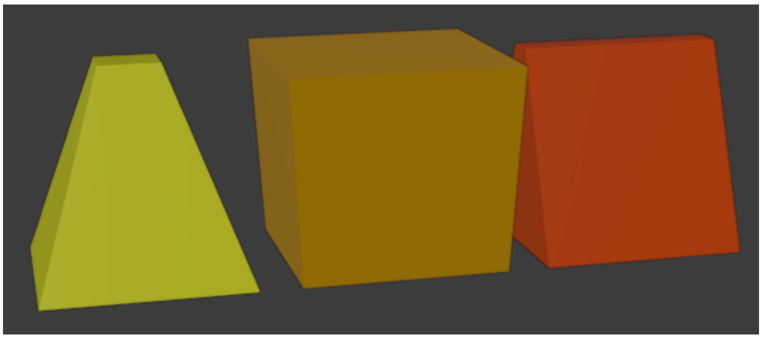

Figure 3.19: *Blender*: Edited connector blueprints. Each blueprint was changed to change the connectors on the corresponding edges.

The suggested value s_i for a parameter i is generated by using the average edge length e and the parameters default setting d_i:

$$s_i = e/2 * d_i$$

This is a simple method, but it is effective for getting the parameters in the right order of magnitude and tweaking them from there.

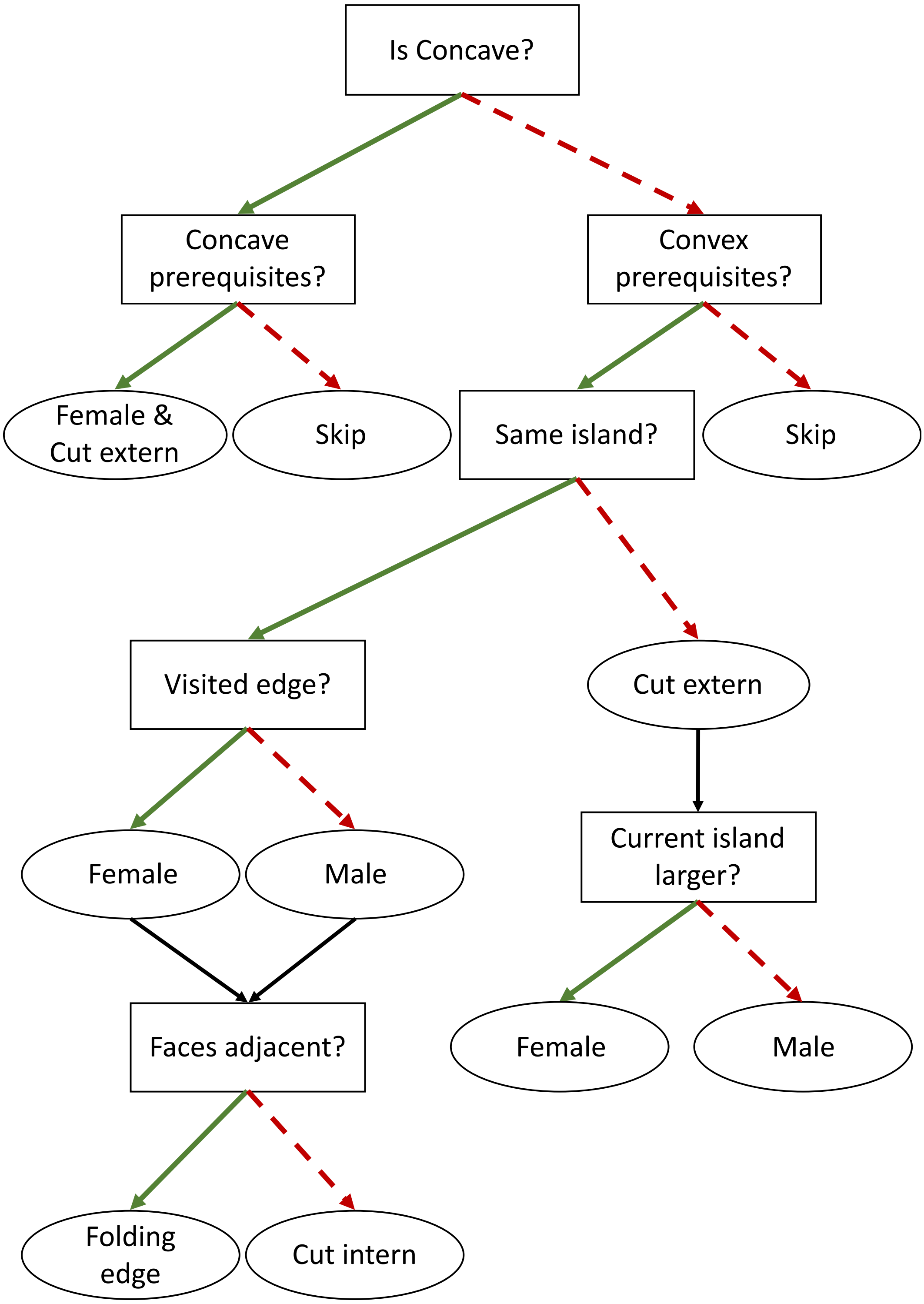

Figure 3.20: This is the decision tree for the connector generation. Ellipses contain decision results and rectangles contain decision questions. Two feature types need to be decided: Connector direction (Male/Female/Skip) and connector type (Folding edge/Cut intern/Cut extern). In the graph, green arrows mean "True" and red dotted arrows mean "False".

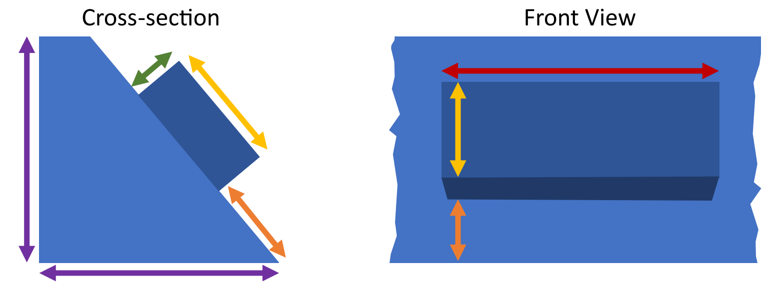

Figure 3.21: Illustration of the parameters: Thickness (purple), Length (red), Height (yellow), Extrude (green) and Offset (orange).

In a former iteration, the suggested settings were computed by using the bounding box dimensions of the object, but we figured out, that the average edge length is more representative for determining the connector sizes.

3.2.6 Previewing

A frame update normally needs between 6 and 20 seconds

The user chooses the frame that should be previewed. Generating a frame takes time, so continuous updates would interrupt the workflow. Therefore, the preview only updates on user request.

Asterisks mark unapplied changes

When a parameter gets changed to a value that is not applied in the currently shown preview, this parameter gets marked with an asterisk. The "Region-View" allows the user to see the different cutting regions on the object itself. Faces that are part of the same cutting region get colored in the same color (fig. 3.22).

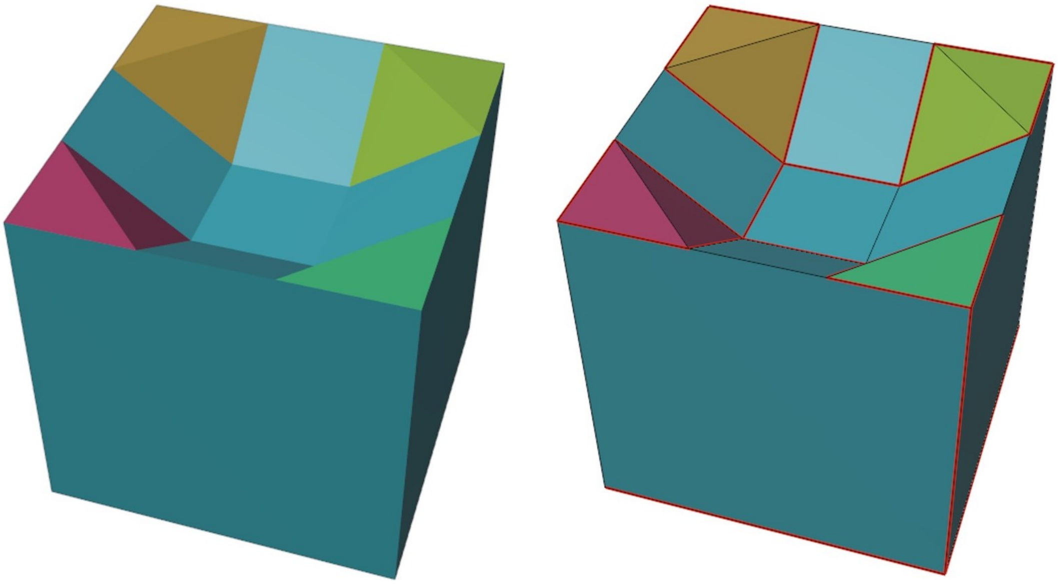

Figure 3.22: Region view of an object. This mesh was split into five cutting regions (blue, green, yellow, orange, purple). Each region corresponds to one frame. Left: Mesh as shown to the user per default. Right: Mesh with cutting lines (red) and folding lines (black), as shown when the user switches to *Blenders* "Edit"-mode.

3.2.7 Export

The export contains multiple files

Finally, when everything is tweaked right, the user can export the output of the add-on. There are different output files: Frames that need to be printed on fabric, extra connectors for concave edges (not printed on fabric) and two files with information for the user.

The two information files are optional. One of them is a text file with the parameter values used, and the other one is an .obj-file that contains a 3D object that shows the region view.

By outputting every mesh in its own file, the user is free to arrange them. For example depending on alignment to texture of the fabric or on print bed size. They can also use packing functionalities of slicers like *Cura*[3] by *Ultimaker*.

[3]https://github.com/Ultimaker/Cura/blob/master/cura/CuraApplication.py

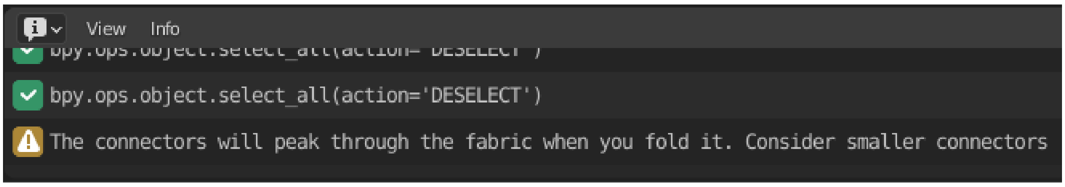

Figure 3.23: A warning is shown in the info-panel. This panel behaves like a log and also contains other *Blender* output.

3.2.8 Warnings

The warnings are aimed to help finding solutions

In case inputs can not be processed correctly, or could have undesired effects, warnings (tab. 3.1) are shown to the user in the *Blenders* info-panel (fig. 3.23). In severe cases the add-on will abort. Where possible, it fixes the issue or ignores it for the rest of the process. Features for automatically fixing issues are described in section 3.2.9. The "Connectors will peak..."-warning is explained in figure 4.10. Not all of the implemented warnings can be triggered in normal usage scenarios.

Warning	Comment	Action
Select an object.	In panel, deactivates UI	Abort
There are x objects selected.	Shown if x is not equal 1	Abort
The selected object is not a mesh.	Only suitable for meshes	Abort
The frame is selected. Select the object!	Could't fix with history	Abort
A connector is selected. Select the object!	Could't fix with history	Abort
The object is too detailed, maybe simplify...	Maximum of 50 faces	Abort
You might have floating connectors.	Offset larger than height	Continue
Only x% of the connectors were generated...	Includes skipping reasons	Continue
Connectors will peak...	Bad for assembly	Continue
There is no frame with the index: x.	Can't happen in normal usage	Abort

Table 3.1: All implemented warnings.

3.2.9 Usability Features and Error Prevention

During the implementation of the add-on, high priority was given to enhancing workflow and usability. This section contains an overview of the resulting features.

Undo and Redo
The add-on is registered in *Blenders Undo-History*, which allows to use undo and redo operations as usual.

undo works

Selection
The add-on needs to know what object should be processed. The user specifies this by selecting the object. There are different cases how this could go wrong: When multiple objects are selected a warning is shown. However, selecting multiple objects by accident barely happens. It is more likely to accidentally not select any object. In this case the "Select an Object" warning is shown in red in the add-on-panel (fig. 4.9) and the complete UI of the add-on is greyed out to communicate early, that the add-on would currently not work. If exactly one object is selected, but it is one of the connector blueprints or the frame, the add-on tries to fix it. It is expected that the user does not want to unfold one of those "tool-objects". That would be comparable to a painter painting his own brush. To fix this automatically, the add-on selects the last valid object it worked with. Only if that does not work, a warning is generated and the add-on aborts.

Common selection mistakes are either prevented or get fixed

Triangulation
Meshes can have non-planar faces. Such faces are not clearly defined and can not be processed correctly. When an object contains them, they are automatically triangulated and the process is continued.

Non-planar faces are automatically fixed

Perspective
The perspective of the 3D viewport is automatically adjusted. When a new frame is generated the perspective is changed to it. When the "Region-View" is activated, the perspective is changed to the object that got colored.

The perspective adapts to actions

Mode Switch

Common mode mistakes get fixed

The operations can only be correctly executed if *Blender* is in "Object-mode". To make adjustments to meshes the user switches to "Edit-mode". When triggering actions in "Edit-mode" the "Object-mode" is automatically activated. This also results in a coherent result presentation.

Short Minimal Workflow

Users quickly see tangible results

The user can see a solid frame just by selecting the object, clicking the "Suggest Settings"-button and the "Create"-button. In simple use cases, the user can directly export a usable result.

Frame Numbers

Frame number mistakes are prevented

When the user enters a frame number that is too large and generates a new frame, the number automatically jumps back to the first frame. Experienced users also have the opportunity to enter -1 to jump to the frame with largest number.

Garbage Collection

The project file stays uncluttered

Objects and meshes that are generated during the usage get deleted from the scene and the project when they are not needed any more. This saves memory and keeps the project data clear.

Mesh Processing

Performant mesh processing

For better performance, mesh operations are done directly on the face-list data structure if possible. This way, *Blender* does not recalculate the mesh data. When creating new meshes, the mesh data is only calculated once at the end and not after each addition of a face.

Optimized Boolean-Operator Usage

Saving time by reducing the number of boolean operations

Boolean operators are computationally expensive. Before applying them, all modifying shapes are quickly merged by appending their face-lists. This way, the boolean operator is only called once and not for every single modifying shape. Creating the frame for the *Blender* standard cylinder without connectors took 2:34 minutes. With the new approach, the same process took only 7 seconds.

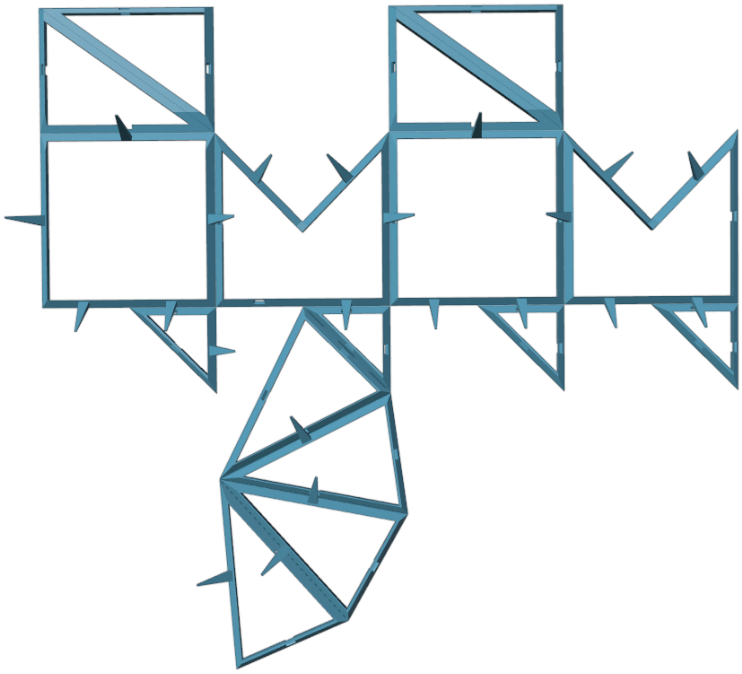

Figure 3.24: Exaggerated connectors that stress the connector order. The big square on the left is the face with the highest priority, because it is one of the two largest ones.

Connector Placement

The decision which side of an edge gets a male and which side a female connector is not random. We expect that the pointing direction of the male connectors can be used as an indicator for the user on how to start assembling the object (fig. 3.24). The largest face has only male connectors and adjacent faces also have male connectors where still possible. This spreads through the entire object. The code is structured in a way, that alternative connector hierarchies could be added easily.

Assembly guidance through systematical connector placement

Connector Blueprints

The cubic connector blueprints that are automatically generated can be changed and replaced intuitively in the "Edit-mode". Users can reposition them in "Object-mode" without side effects to arrange the work-space by personal preference. Deleted connector blueprints will be restored automatically with the default blueprint when the

Blenders extensive "Edit-mode" tools can be used for connector design

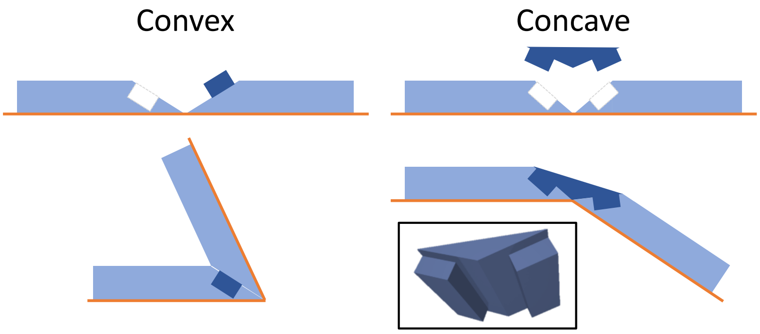

Figure 3.25: Comparison between concave and convex edges in flat (top) and folded state (bottom). Concave edges need extra connector parts (framed in black).

"Create/Refresh"-button is activated. The three different connector blueprints are color coded and named according to their function to make it easier to tell them apart.

Skipping Connectors

Useless connectors are not generated

Individual connector pairs get skipped if it would not make sense to generate them. Reasons for that could be either that the edge is too short to have room for a connector, that the angle of the edge is too big or that the angle is too small. If a huge portion of connectors is skipped, the user is warned. The warning includes the proportion of generated connectors, as well as how many connectors were skipped for what reason.

Concave Connectors

All concave connector extra parts are the same, no correct assigning is necessary

Concave edges need extra wedge parts that connect the two faces (fig. 3.25). For an easy assembly, all wedges are identical. So it is not necessary to map them to a corresponding edge. Concave edges can be easily identified, because they have two female connectors. The male connectors are located on the wedge. For each concave edge, a wedge is generated automatically, so that the user can directly print them.

Chapter 4

Evaluation

In order to evaluate the add-on, a qualitative user study with 13 participants was conducted. The goal was to find out how self-explanatory the user interface is, how intuitive the workflow is, where which mistakes and questions occur and how good the overall impression by the users is. Expectations, suggestions for improvements, negative traits of the add-on, general observations, interesting quotes and aspects that stood out positive to the participants were collected.

13 participants with varying 3D software experience

4.1 Procedure

Prior to the study, the *informed consent form* was sent to the participants, to enable them to read it in advance. The study was done remotely via the video communication tool *Zoom*[1]. It consists of four phases (Preparing, Impression and Orientation, Tasks, Follow-up Questions). Since the mother tongue of all the participants was German, the study was conducted in German, too.

The study was conducted in German

[1]https://zoom.us

4.1.1 Phase 1: Preparing

In the first phase, the formalities were managed, general information collected and the participants were introduced to the subject.

The participants were informed that the study would be recorded and that the recordings will not be published. They got an overview what the study is about and that it was not about testing them, but about testing the current state of the product. After that, the consent form was discussed and any questions regarding the study and the form were answered.

Information about demography and experience were collected

Next, the first page of the questionnaire was filled out by the participants. It contains general demographic questions about the digital experience of the participant, starting with general computer experience and narrowing down to *Blender* experience. If the participants had *Blender* experience, they were also asked how they liked *Blender* in general. This was done to gain insights about the general mindset of the participant towards the usability of *Blender*. The participants were encouraged to comment their answers to gain further insights.

The fabrication process was explained to put the add-on into context

After that, a short explanation of the *FabricFaces* concept was given, accompanied by a stylized visual illustration of the process. The facts given in this short explanation were:

- We investigate an approach to fabricate 3D-Objects. It is called *FabricFaces*.
- The object gets unfolded and 3D printed on fabric.
- When unfolding an object, one or more support structures might be generated. Those structures are called frames.
- This study focuses on a *Blender* add-on, that computes those frames.
- *Blender* is a program to edit 3D models.

The participants were asked if they felt confident that they understood the given information and they were encouraged to ask questions about it if they had any.

This part of the study was not recorded, since the consent needed to be given first and it allowed the participants to get used to the situation and the concept with less pressure.

4.1.2 Phase 2: Impression and Orientation

This part of the study is about the expectation of the participants and their first impression of the add-on. The recording was activated at this point.

User expectations were collected

The participants were asked what they would expect from the add-on and after that what features they would appreciate. This was done to get insights into overall expectations like speed or simplicity, but also on features the user might want to see in such a program.

Users were asked to explain the UI before using it, to investigate how self-explaining it is

Next, a screenshot of the add-ons user interface, without the surroundings, was shown to the participants. They were asked what they think what the different buttons and settings purposes are. The ten features on the interface were labeled with numbers to make them easier to address and to encourage the participant to talk about every single one of them in the same order.

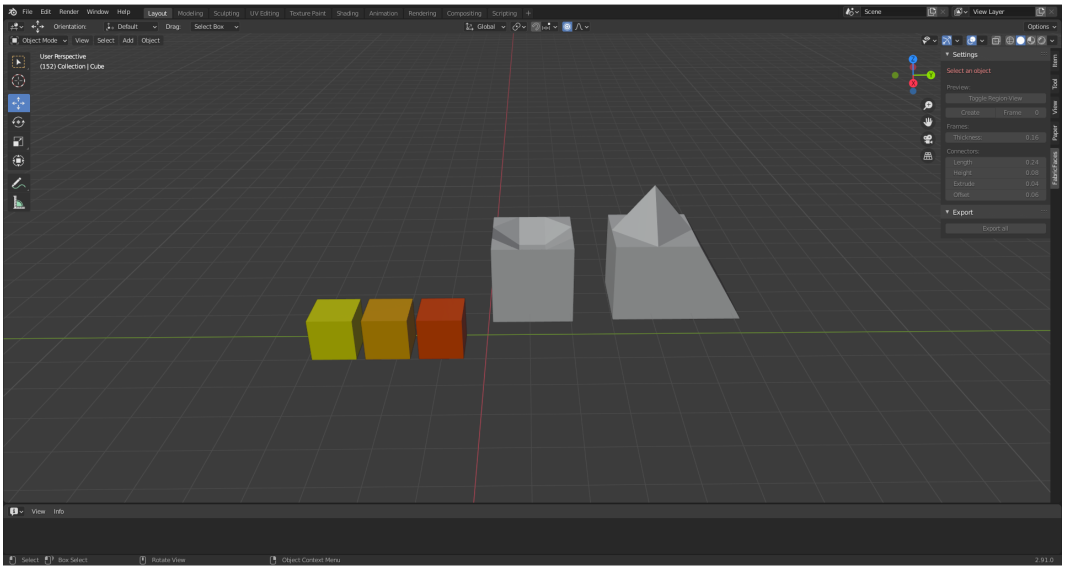

Figure 4.1: *Blender*: The project that was prepared for the users. From left: The three connector prototypes, a cube with a through (five cutting regions) and a strange house (one cutting region). Information output at the bottom.

A short *Blender* introduction was given

After that, a prepared *Blender* project was shown to the participants (fig. 4.1) and a short explanation of relevant interaction was given. It included:

- How to control the camera.
- How to select an object.
- How to center the view on an object.
- How to make simple changes on an object (moving faces and edges).
- Where to find the plugin controls.
- A short explanation of the three shown connector prototypes.

The remote control was now activated for the participants and they were encouraged to explore the add-on while thinking aloud. Questions about expectations were asked and if the participants had a question, we usually asked what they thought the answer would be before answering the question.

When it seemed like the participants tried everything they wanted, we asked some questions about features they might have overlooked. For example the star markings on the changed numbers, or the warnings at the bottom of the screen. The questions were asked in a way that allows to differentiate between: the features were seen, understood and not commented, the features were seen and ignored, or the features were not seen at all.

Users freely explored the add-on before getting the tasks

When the participants decided they were finished exploring and it seemed like there were no major misconceptions, they were given the tasks.

4.1.3 Phase 3: Tasks

In this phase, the participants were asked to perform four different tasks.

The first task was to use the region view on the left object and to interpret the results. This task aims at clarifying the fact, that there are different cutting regions and that they can be viewed this way. It can be solved by selecting the object and clicking the "Toggle Region-View"-button. The participants were asked to explain what they think the regions mean, to investigate how well this characteristic of the technique was overall understood. If the participants were unsure or got something wrong it was explained to them.

It was verified if the users understood the region view right

The second task was to have a look at all the frames of the left object using standard settings. This task aims at testing the frame preview workflow and the understanding of the setting suggestions. In the formulation of the task the phrase "suggested settings" was intentionally replaced by "standard settings", to find out if the participants understood the functionality of the "Suggest Settings"-button correctly.

It was investigated if the user understood the basic workflow

The third task was simply to export the results of task two, to test the export workflow. If this was the first time the participants opened the export dialogue, they were also asked

The exporting was tested

what they thought the export settings mean. If they already opened it in the orientation phase, they had been asked the same at that point. After they exported, the export folder was opened and the results of the export were discussed. The participants had time to view the results, talk about them and ask questions.

It was investigated what the users preferred workflow was

The last task was more open. The participants were asked to adjust the settings and the connectors for the right object, so it made sense to them. The object was designed in a way that it only consists of one frame, because the frame switching was not the focus in this task. Using the suggested settings on this object results in a warning, because the connectors would stick out of the faces in the folded state. This was done to investigate if the participants noticed the warning and understood its meaning. The participants were asked to get rid of the warning if they did not already try that by themselves.

4.1.4 Phase 4: Follow-up questioning

After the usage, the general impression of the users was inquired

This phase aimed at investigating the overall opinion of the participants regarding the add-on. They were first asked if they have general questions regarding the add-on or the *FabricFaces* process in general. Then, they were asked if there were situations or moments during the usage of the product that irritated or that bothered them and to describe them. After that, they were asked if there were aspects that stood out positively to them and to describe them. Then they were asked if they have any remarks or suggestions for the add-on in general.

SUS questions were asked

After that, the suggested questions by Bangor et al. [2008] were asked. They aim at evaluating the usability of the add-on and allow to compute the SUS-score.

SYSTEM USABILITY SCALE:
The System Usability Scale (SUS) aims at evaluating the usability of a product. 10 questions are asked in a defined order. They are statements about the product. The user states how much they apply from 1 (does not apply) to 5 (does apply). The SUS-score is an usability indicator that can be computed from the answers.

The questions were translated to German. The participants were asked to answer the questions with the mindset of a person that has access to a 3D printer and that could in general use objects in the *FabricFaces* style. This was done to get more relevant answers, since people without a general affinity to the subject of course would not use such a program.

At the end, the participants were invited to ask questions they still had, or comment on subjects they wanted to add something to.

4.2 Participants

Engineers provided an interesting additional perspective

The study had 13 participants of which five were female. Overall the participants had an average age of exactly 25 years with a standard deviation of 2.9 in a range from 22 to 32 years. Most of them were students. All their academic backgrounds were scientific. 8 had a background in computer science and 3 in engineering.

Participants with knowledge in *Blender*, CAD or texturing were found

The participants reported to feel confident on using a computer. Most of them had some experience in working with 3D software. 5 were confident in the usage of Computer Aided Design (CAD) programs, and 2 were experienced using texturing tools. 9 had worked with *Blender* before. From the people who knew *Blender*, most of them reported that they like to use it "medium" or "reluctantly". While there was 1 participant that used *Blender* very reluctantly, there where 0 participants that used *Blender* very gladly. 8 participants had varying experience with 3D printing.

4.3 Results

Observed characteristics were systematically compiled

The resulting video recordings were analyzed and relevant characteristics were collected in an *Excel*[2] sheet. Those characteristics were split in six classes: Positive observations, negative observations, neutral observations, suggestions, expectations and quotes. For each characteristic and each interview it was marked if it occurred, and the number of interviews it occurred in was counted. This was done in two iterations.

532 characteristics were observed

First, the notes we made during the study were analyzed to get a rough overview over the observed characteristics. In the second iteration, the recordings were attentively re-watched and the table was expanded and refined. 172 unique characteristics were found. As the same characteristic often occurred in multiple interviews, it resulted in a total of 532 observed characteristics. On average, 41 characteristics per participant were observed.

Here are two examples for characteristics: The most common positive characteristic regarding the user interface was "tooltips helpful" with eleven occurrences. This means that in 11 out of 13 interviews the participant stated that they found the tooltips helpful, or a situation occurred, where the tooltips obviously helped the user. The most common negative characteristic regarding the user interface was "Did not understand offset at first/ irritated while using offset/ offset tooltip did not help" it occurred 9 times. This means that in 9 interviews at least one situation occurred that fits to at least one of those descriptions. In cases like this, similar descriptions were merged to keep the number of characteristics manageable.

Expectations, observations and user suggestions are presented

In the next subsections our findings are presented, divided into different categories. The order of the categories based on the time of their appearance in the workflow. For each category we present the characteristics that occurred the most, followed by unexpected observations and helpful suggestions that occurred less often. We will start with the initial expectations the users had, followed by a detailed

[2] https://www.microsoft.com/de-de/microsoft-365/excel

analysis of observations regarding the user interface and the preview functions. After that, we evaluate the warning system and the export functionality. Finally, we focus on the general impression the users had.

All expectations, observations and suggestions mentioned in this thesis are compiled in appendix A. Design improvements based on the study will be derived in section 4.4.

4.3.1 Expectations

Before the participants saw the add-on, or the photo of the user interface, they were asked what they would expect from such a tool.

The simplicity of the workflow was subject of the two most common expectations. 10 participants expected the add-on to be easy to use with a simple workflow, for example a one button solution. 7 participants expected that many steps are done automatically, for example the decomposition and unfolding of the object. As the third most common, 5 participants expected that they could determine or influence how the object gets unfolded. Those three expectations are met in the current implementation.

The workflow was expected to be short and easy

Other general expectations by 3 or less participants were correctness, integrated explanations or assistance, similarities to UV-texturing tools and assistance for the physical assembly.

There were also some interesting expectations or ideas for features that are not part of the current implementation (1 or 2 participants): Support of screw holes, an animation for the unfolding process, the generation of different suggested unfoldings to choose from and different connector profiles.

Interesting additional features were expected

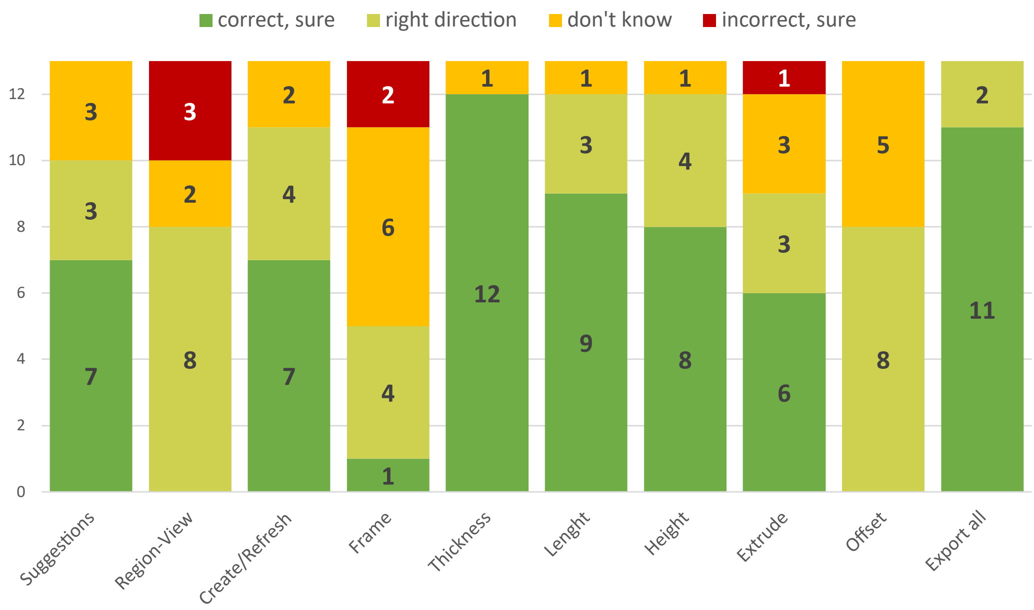

Figure 4.2: The participants described the functionality of the user interface before the first usage, to find out how intuitive the different elements are. The frame selector and the "Toggle Region-View"-button seem to be the least intuitive, while "Export All"-button and the "Thickness"-parameter were understood quite well.

4.3.2 User Interface and Parameters

To find out how self explaining the different UI elements are, the participants were asked to describe the functionality of each one on a photo, before usage.

Overall the UI was understood well

The results (fig. 4.2) demonstrate that for 9 out of 10 UI elements more than half of the participants understood them correctly, or were guided in the right direction.

UI misconceptions often rooted in misunderstandings of the concept

Most of the wrong associations the participants had, were caused by misconceptions of the add-ons output or purpose. For example in case of the "Extrude"-parameter, one participant thought, that this value refers to the nozzle thickness of the 3D printer. Two other participants initially had that idea too, but discarded it later.

The two elements that are the least self explaining are the ”Toggle Region-View”-button and the preview-frame selector. Both of these elements control features of the add-on that were not expected by most users.

The word ”View” seems to trigger the wrong mental model for ”Region View”

The region-view was often misinterpreted as a change of perspective or switching between unfolded and folded object state. 5 participants thought about the perspective or the camera changing when they pressed the button. Some later discarded this idea. No participants understood the button completely, but most of them understood that it would help to somehow identify different regions. When asked what this regions could be, the participants were unsure. This is the only element were all of the participants that felt like they understood it, understood it wrong.

The frame selector that allows to choose which frame gets shown, was only understood correctly by 1 participant. 2 participants thought that it was used to determine how many different frames the object should be cut into, while most were just very unsure about it.

The parameters were understood quite well, with the exception of the ”Offset”-parameter. No participant got it completely right, most of them just knew that it somehow moves the connector.

Tooltips

Tooltips are helpful, but might not get noticed

During the actual usage of the add-on, 11 participants found the tooltips helpful. However 5 participants overlooked the tooltips at first. When they had a problem were the tooltips might help, they were made aware of them after a while. Most of the participants used them frequently from then on. The offset parameter and its tooltip caused some confusion: 9 participants did not understand the parameter at first. In most of these cases the tooltip ”How far should the connectors be away from the ground?” did not help them. For example P2 was still thinking about the object in the folded state. This person suggested, to rephrase the tooltip to ”How far should the connectors be inside of the object?”

Asterisks

Nearly half of the users understood the asterisks intuitively

All parameters that differ from the currently shown frame are marked with an asterisk. 6 participants understood that intuitively. 4 participants thought that they mean that the currently chosen parameters differ from the suggested ones, and 1 participant thought, that they mean that marked values are values that cause problems. The asterisks were easy to recognize. Only 2 participants did not notice them by themselves. During the study, 8 participants found the asterisks explicitly helpful. For example, P4 said "Ah, okay. So this value was not a standard setting before. Good to know." after reloading the suggested settings. "This is similar to unsaved projects", P11 described his understanding of the asterisks. Many programs use an asterisk behind the project name to indicate that the current state of a project is not saved. This fits well to the purpose of the asterisks in the add-on.

"Create/Refresh"-button

A characteristic property of the workflow in the add-on is the "Create/Refresh"-button. This button has to be pressed to apply changes in the settings to the preview. It is the most pressed button in the add-on, so it is desirable that it can be used intuitively.

A quick live update would be the best solution

5 participants did not like the concept of refreshing the frame to see changes. One of them was so annoyed that the person sighted every time the refresh button had to be pressed. One participant had a background in 3D animation and stated that it would be desirable to have the preview updated automatically, even if that would lead to a slower interface. This would be normal in other software, too.

5 participants stated that while not liking the manual refresh, they prefer it to a simple automatic one, that would result in a slow interface. "Having to refresh is not ideal, but it is certainly better than an interface that is always lagging, and one gets used to [the refreshing]" (P2).

During the study we observed that 6 participants had problems with this button, because it was not prominent enough. In many cases participants repeatedly moved their cursor to another button and then changed the trajectory spontaneously to the ”Create/Refresh”-button (fig. 4.3). ”I would expect the button to be big and at the bottom, like an ”Okay” in a dialogue.” P6 said. 3 Participants clicked the big ”Export All”-button at the bottom instead of ”Create/Refresh” and 2 other participants did the same with the big ”Toggle Region-View”-button on the top.

The ”Create/Refresh”-button is not prominent enough

We also observed at least in 3 interviews situations were the participants pressed enter after entering a value, expecting to trigger the recalculation of the frame. This was hard to observe, and it might have happened in other interviews, too.

Frame selector

The frame selector is used to switch between previewing different frames. 5 participants had trouble understanding what the frame selector does, or that it only has an effect after refreshing. 5 other participants however directly used the frame selector intuitively or understood it quickly.

Opinions on the frame selector differ

The frame selector was often used in situations where no other actions were triggered between the frame switch and the press of the ”Create/Refresh”-button. 5 participants suggested that the refresh operation should be triggered automatically when the frame number is changed.

In the current implementation, the wrapping of the numbers in the frame selector is only triggered when the ”Create/Refresh”-button is triggered, due to *Blenders* way of handling properties. This means that if a number outside of the valid range is entered, the add-on jumps to the first number in the range on the other side (max+1 to 0, -1 to max). This results in a behavior that allows to skip over the boarders of the range easily.

The current wrapping is not optimal

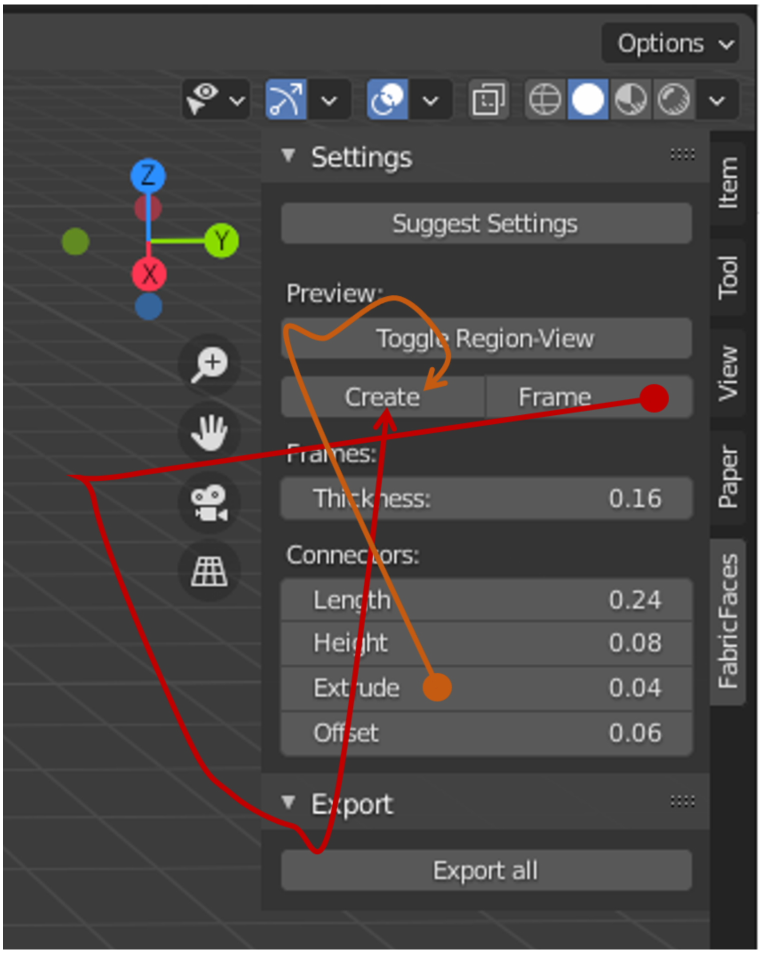

Figure 4.3: Sketches of two typical cursor movements that were observed (red: P6, orange: P12). In both cases the initial target of the cursor movement was one of two big buttons and was later corrected to the correct button. Both typical examples were observed after at least five usages of the "Create/Refresh"-button, when the participants focused on other subjects.

5 Participants disliked this behavior ("The number should not jump around at all" (P8)), while 3 participants liked it ("I like that." (P9)). 6 participants suggested that the wrapping should happen when entering the number "like entry verification in web development" (P10). One participant also suggested to deactivate the frame selector completely when only one frame exists.

8 participants struggled to find out how many frames there are. This could be improved by changing the way the frame numbers are shown. 5 participants suggested to start counting from 1 instead of 0 and 2 suggested to show the maximum with a slash (for example: "Frame: 2/5").

Users struggled to find out how many frames there are

The term "Frame" should also be re-evaluated, 2 participants were confused by it, because they expected the frames to be pictures of an animation or keyframes, which also is a far more common use of the word in software. "I first thought about *Blenders* timeline, although that does not make any sense in this context." (P2).

The term "Frame" is already occupied in *Blender*

Parameters

During the usage, in most cases the parameters were used intuitively. However, 3 participants stated that they were unsure about correctly assigning the purposes of the "Extrude" and the "Offset" parameters and 1 participant mentioned the same with the "Height" and "Offset" parameters.

We observed that 11 participants used extreme values on one or more occasions to see what a parameter does. "Okay, 0.8. Go wild! Let's see what will happen then." P7 said when entering a high value for the connector length. Some participants seemed to prefer that over reading tooltips, or were just curious for the results.

Extreme values help to clarify the impact of a parameter

One participant said that the parameters were "every thing that is needed, and not more than needed" (P2) and in no interview a situation occurred were a participant wanted to adjust a general parameter that was not accessible. One user suggested to mark extreme values in red instead of capping them, "maybe I am an experienced user and really need this high value for experienced user stuff. It then would be nice to be able to enter those values." (P8).

We picked the right parameters to be accessible to users

Suggested Settings

The suggestion feature was called good or helpful by 10 participants. 4 participants later mentioned it as a feature that they were positively surprised by. "You quickly have

The suggested settings were appreciated

something tangible, just by using the suggested settings." (P7). In many cases the participants used the feature as a starting point for the design process or as a way to get back when they felt like they messed something up. Less experienced users were liked to just load the suggested settings and use them without changes.

Users strongly trusted in the suggestions

The trust in the suggested setting was higher than expected. One participant suggested to hide the connector parameters under an "Advanced"-section, which quite surprised us since we had thought about such an "Advanced"-section ourselves, but would have put far more specialized options in it.

Changing button labels might irritate users

The fact that the "Suggest Settings"-button changes its label to "Reload Suggestion" after the suggestions were generated seemed to make some participants uncertain: "When I see a new program, I look were all the buttons are and what they are called. They should not change their names after that. It makes orientation harder." (P5).

4.3.3 Preview

In this subsection, we start with our compiled results regarding the two previewing features "Region View" and "Frame Preview". After that, we focus on special aspects like the camera behavior or the connector blueprints.

Region View
The region view is an unusual concept that was not easy to convey to the users.

The region view is unusual, but helpful

7 participants did not understand the "Toggle Region-View"-button at first. Many of them had not fully grasped the concept of multiple cutting regions at this point. However, 9 participants later called this feature explicitly good or helpful.

The info panel is not a good place to show information

Only one participant realized, that the number of cutting regions was shown in *Blenders* info panel at the bottom. When re-evaluating the name of this button, it is important

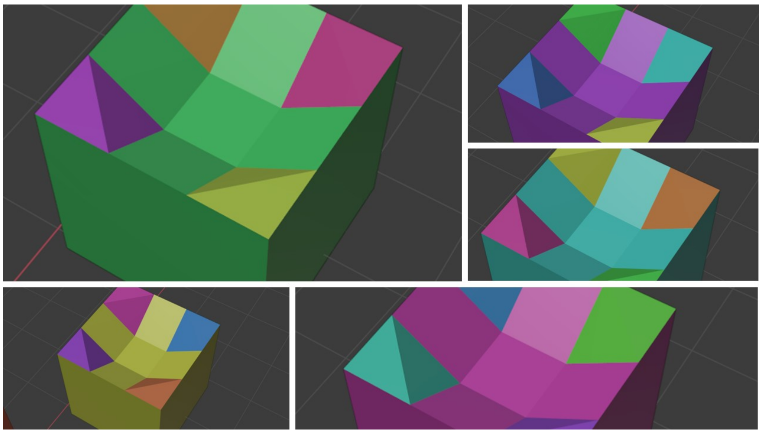

Figure 4.4: The colors change every time when the region view is activated. This allows to find a fitting color scheme, but irritated participants. Two different frames never get the same color. Frames with close indices get very different colors.

to notice that the ”Toggle”-part of the name seems to be understood well. ”Toggle means for me to switch between two states.” (P8). The ”View”-part of the name however lead 5 participants to expecting a switch in perspective or camera.

The colors should not change

The colors change every time the region-view is toggled on again (fig. 4.4). One participant liked that a lot. ”This might allow people with red green weakness to find a fitting color scheme” (P1). 5 participants disliked this or were irritated by it. ”I imagine it to be frustrating to lose the colors when taking notes about different colored parts for example.” (P13).

Frame Preview

Users suggested more complicated, but faster frame preview methods

As mentioned, some participants struggled with the frame selecting and refreshing workflow. Suggestions were made to increase the responsiveness of the frame switching. 4 participants suggested to show all the frames at once. “It would be nice if all parts would be shown, and if they would be arranged how they connect, like a flat exploded view.” (P13).

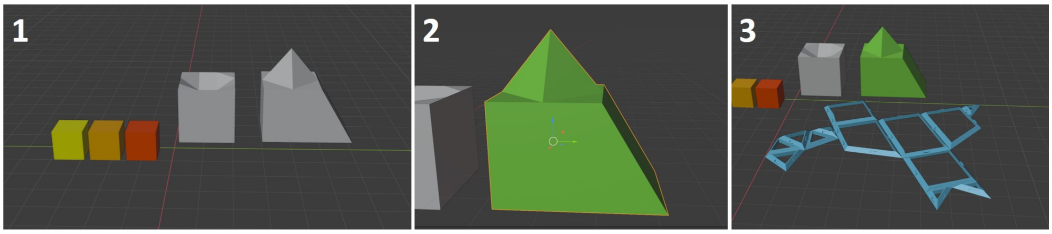

Figure 4.5: Camera behavior: 1) The initial camera view. 2) The view after activating the region view on the right object. 3) The view after creating the frame.

Color coded frames would be helpful

2 participants noticed that the generated frames are sorted by complexity. Both liked that. However, participants had problems to find out where which frame would be located on the object. 9 participants suggested to color the different frames in the same color that was assigned in region view to them. One participant further suggested to show the corresponding color also in the user interface, for example with a colored dot next to the frame selector.

Color coded connectors would be helpful

The add-on allows to specify different connectors for different edge types. The corresponding connectors are color-coded. 2 participants suggested to strengthen this color coding by applying it also to the object, so that every connector generated in the frame-preview has the same color as the corresponding prototype. "This would help to identify which connector does what." (P5).

Camera Behavior

Adaptive perspective is overall good, focusing on details gets difficult

The camera perspective adapts to the users actions like shown in (fig. 4.5). The participants seemed to like this behavior in most cases and took it for granted. 3 participants explicitly mentioned that they like this feature, while 5 participants pointed out that it is hard to observe details that have changed when refreshing. It seems better to only refocus on a frame when it is not the same frame of the same object.

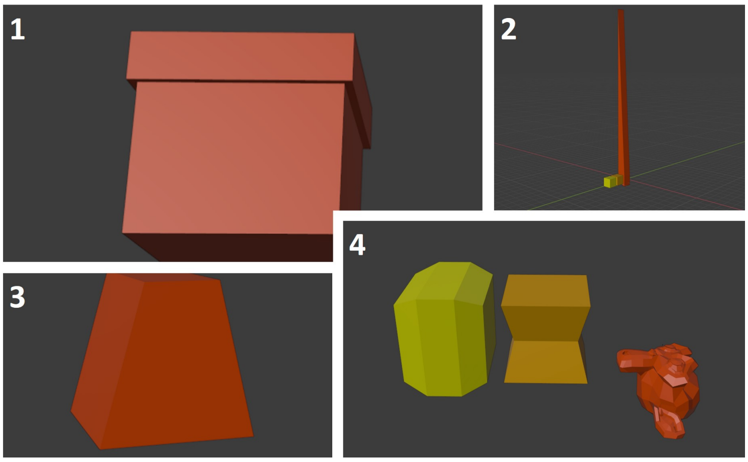

Figure 4.6: Different connector blueprints designed by participants. 1) A simple clip connector. 2) A very large connector. To see what happens. 3) A simple wedge connector for easy assembly. 4) A range of interesting connectors with no particular purpose.

Connector Blueprints

Editing the connector blueprints was the most advanced user task and required the most knowledge about *Blender*. However, 5 participants had fun when using this feature and enjoyed playing around with different shapes (fig. 4.6). For example, one participant invented the "monkey face"-connector (fig. 4.7). Participant and study conductor were equally surprised to find out that the add-on could handle this type of detailed, non-manifold mesh with some non-planar faces as a connector blueprint. The performance did drop slightly, but was far better then expected.

Playing around with connector types is fun for experienced users

A participant with material science background suggested to allow the user to specify separate female connectors to allow the design of bendable clips (fig. 4.8). This would support to include the work of Mattiussi [2020].

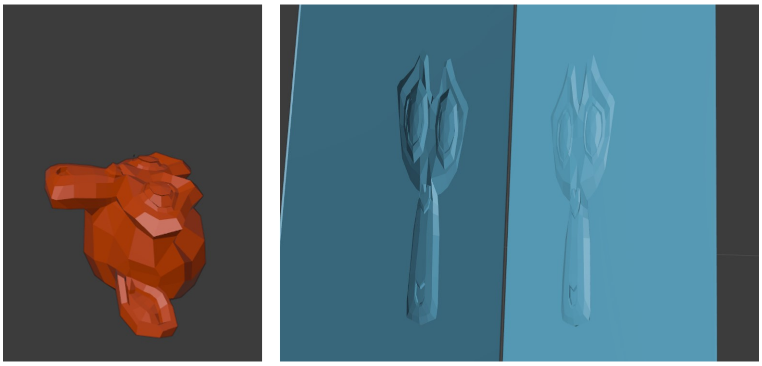

Figure 4.7: The Monkey connector blueprint and its connectors (by P11).

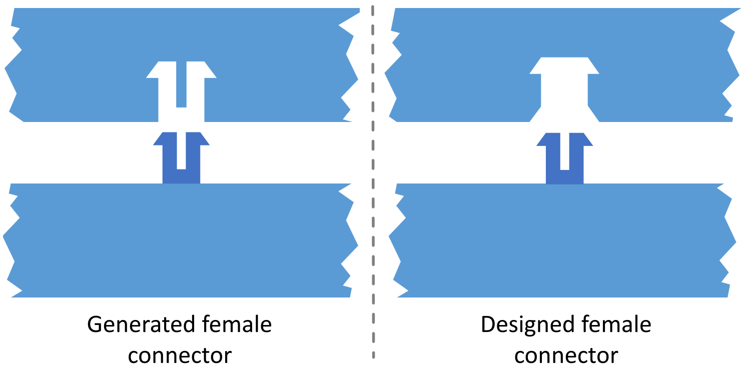

Figure 4.8: New design possibilities arise when allowing to specify female connectors manually as suggested by a participant. The right design would allow a better clip-connection than the left.

Users suggested small changes for easier connector editing

In two cases, participants created connectors that were not symmetrical in the standing axis. In both cases the users were surprised by the orientation of the connectors "I expected them to be the other way around" (P11). To better match the expectations of users, it would be better to turn the connectors by 180° on their standing axis. P13 asked "Is it possible to transfer settings from one connector to another?" This is currently not implemented, but would be a helpful feature. A range of different presets might be helpful here. "I would like to have different connector-presets that I can chose from, like in a construction kit" (P13).

4.3.4 Warnings

In this subsection, we will evaluate the warning visibility and how well the warnings were understood.

Warning visibility
When undesirable properties are detected, a warning is generated. It gets shown at the bottom of the screen and in the info-panel. Only 2 participants directly reacted to those warnings and 10 participants did not see them by themselves. One experienced *Blender* user did see them, but thought that they were *Blender*-intern and ignored them. 6 participants said that they were irritated by the program logging that *Blender* also writes in this panel and thus ignored the area.

The warnings are presented too reluctantly by *Blender*

There is one warning that was shown at another location. The "Select an object."-warning is shown in red at the top of the plugin window, replacing the region view button and graying every other element of the add-on out (fig. 4.9). This warning was only overlooked by 3 participants at some point during the interview. This suggests that users would recognize warnings easier if they were showed in the add-on panel itself.

The warning shown in the add-on panel was noticed better

"Connector peak"-warning
The most triggered warning during the study was, by study design, the "Connectors peak through fabric"-warning. It occurred when creating the frame for the right object with the suggested settings. 8 participants understood this warning by themselves. This warning is triggered when a female connector pokes out at the bottom of the frame. When the object gets folded a male connector will be at the same position, so will peak out of the frame too (fig. 4.10). This is undesirable since the frame is printed on fabric. The warning further suggests "Consider smaller connectors."

The "Connector peak"-warning was well understood

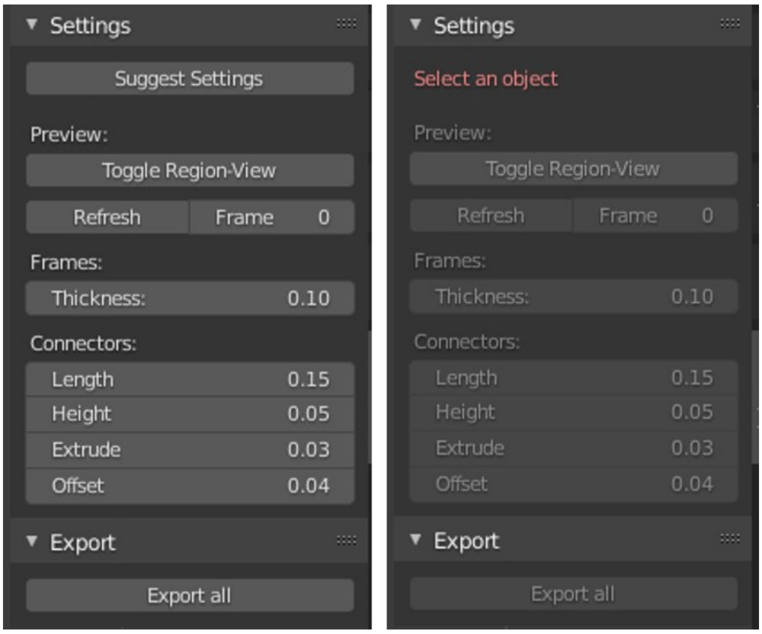

Figure 4.9: Left: The UI in the default state, Right: The UI is deactivated when no object is selected and a warning is shown in red.

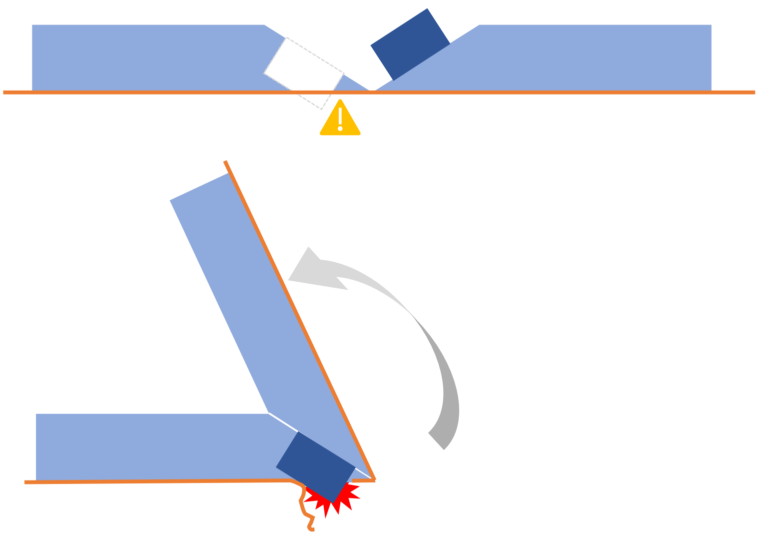

Figure 4.10: The frame (blue) is printed on fabric (orange). The add-on detects situations were the connectors would peak through the fabric.

6 participants reduced the extrude value to solve the problem. Other working methods are to reduce the offset (5 participants) or to change the connectors itself by editing the connector blueprints (2 participants). 2 participants did not try actively to resolve the warning. 2 other participants mentioned that they found this warning “very helpful”. “It is nice to have error messages that are easy to understand [. . .] and to have suggested solutions.” (P10).

Different solving approaches were used: Extrude, offset and edit-mode

Another warning that often occurred in the study was the “floating connectors”-warning that was triggered by 5 participants mostly when using deliberately large offset values. This warning was always understood correctly when triggered. It is possible to enter values that create connectors in a way that the female connectors are not surrounded completely by material. 4 participants realized this and suggested to warn the user in those cases.

When triggered, the “floating connectors”-warning was always understood

4.3.5 Export

The ”Export All”-button was very well understood. 11 participants knew what it would do, just by looking at a photo of the user interface and 5 participants later mentioned that they found the export to be intuitive.

Export options

When clicking the button, the standard *Blender* export dialog opens. It is extended by three add-on specific options. 8 participants did not see those settings and were later made aware of them. The participants were asked what they would expect the different options to do. 12 participants understood the ”Export Settings”-option correctly. 10 participants understood the ”Scaling”-factor correctly, and 6 participants understood the ”Export Color-Map”-option correctly. The tooltip and the name of the last option irritated some participants. 2 participants mentioned that the term *color-map* is already used in texturing, and 6 participants noticed an inconsistency with the already introduced ”Region-View” term. 5 participants suggested to somehow show the dimensions of the export, so that the ”Scaling”-

Many users did not see the export options

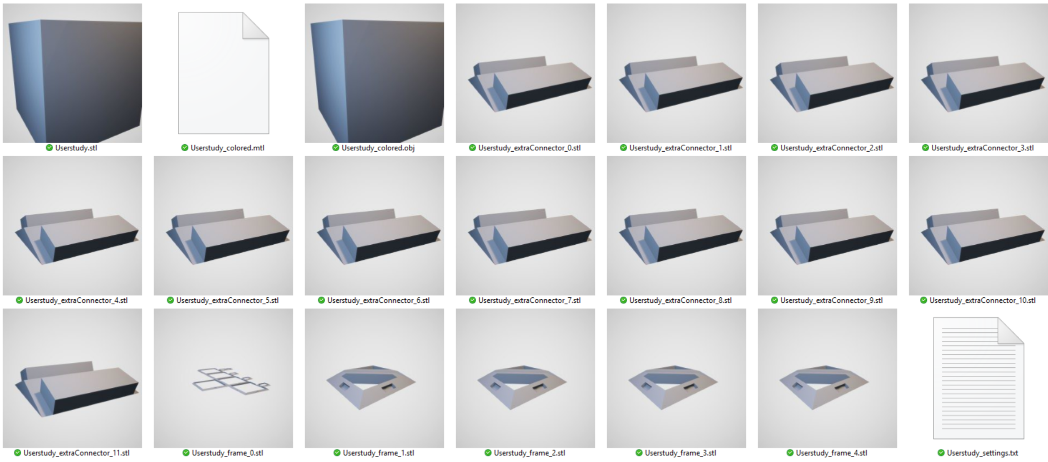

Figure 4.11: The output of a single export operation. The .obj-file is uses color information, that is not correctly shown in the thumbnail.

factor could be adjusted accordingly. It was also suggested by 2 participants to alert the user in the export dialogue if there were open warnings.

Exported files

Many users were surprised by the files for the concave connector

4 participants thought that only the current frame would be exported, or briefly considered this. 6 participants were obviously surprised when they saw the amount of exported files (fig. 4.11). "Wait, that is all mine?" (P7). One reason for that were the extra connectors for concave edges, which were not introduced to the user before. 5 participants confidently but mistakenly assumed that those extra connectors were made for connecting different frames together. The other exported files were understood well, however 3 participants commented, that the .stl-file of the complete object is not necessary when the colored .obj-file exists. It might happen that the .stl-file of the complete object could be printed together with the frames by mistake.

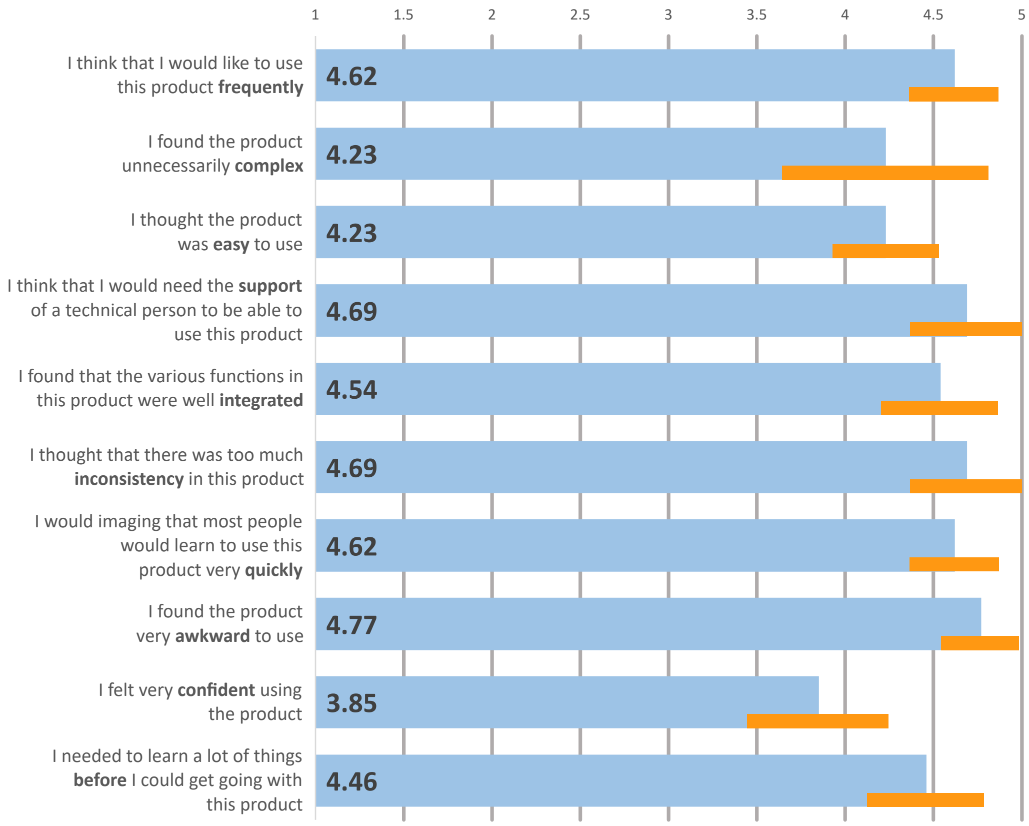

Figure 4.12: The average answers to different usability questions (blue) with standard divergence (orange). The numbers have been normalized, so that 5 means that this aspect is good, and 1 means that this aspect is bad. The average of all normalized answers by all participants is 4.47, with 5 being the optimal result. The standard deviations are (top to bottom): 0.51, 1.17, 0.60, 0.63, 0.66, 0.63, 0.51, 0.44, 0.80, 0.66.

4.3.6 General

After the study the participants were asked to answer ten questions regarding the usability of the add-on (fig. 4.12). The questions are the revised system usability scale (SUS) questions proposed by Bangor et al. [2008]. For our study, the questions were translated into German (appendix C).

Answers might be altered by occasional minimal explanations

During the usage period of the study, we occasionally corrected wrong assumptions the users made and explained things that confused the users, to guide them through the study. These aspects were noted as negative observations. Although we interfered as little as possible, we can not exclude that answers might be affected by this and the translation. We accepted that, because the SUS score was not the core target of the study, and the advantages outweigh those potential deviations.

Our SUS score is 87

SUS scores can range from 0 to 100. The average score calculated from these surveys is 86.73, (standard deviation: 8.9, minimum: 68, maximum: 98.) The average SUS score for graphical user interfaces is 75.24 and in a corresponding adjective rating scale the step "Excellent" ranges from 75.00 to 87.5 (Bangor et al. [2008]). The most room for improvement seems to be in improving the confidence of the user, followed by reducing the complexity and making it easier to use. The best aspect was that the users did not find the add-on awkward, followed by being consistent and usable without technical assistance. The highest deviation had the answers to the question whether the add-on was unnecessary complex.

Overall Impression

Most participants found the add-on intuitive

The overall impression of the add-on was positive. 11 participants mentioned that the add-on is intuitive, pleasant or easy to use. 6 participants mentioned that the add-on was well integrated in *Blender* and did not overlay important elements.

The expectation of the user for a short minimal workflow was met

As mentioned earlier, the most important expectation was that the add-on was easy to use, for example as a "one-button" solution. This expectation seems to be met. 5 participants said that they liked how few mandatory steps the workflow has. "You quickly have something tangible / vivid" P8 and P13 said and P9 stated that "There are less steps than expected." "The add-on does half of it magically! You don't really have to do anything. [...] It's rare that the usage [of a program] works so good so fast." (P11). Two participants stated that the add-on "does exactly what it should do" (P4, P11).

Suggestions
There were some interesting general suggestions made by participants that could improve the workflow or open new possibilities to users.

5 participants suggested to mark detected undesirable features directly in the viewport. For example when the "Connectors peak"-warning is triggered, the connectors that would peak could be marked red. Most of the users that suggested this were CAD software users.

Suggestion: mark errors in the viewport

It was also suggested by one user to highlight linked faces in the object or the frame, when the cursor hovers over the corresponding counterpart. This would allow an intuitive relation between folded and unfolded object state. Another interesting suggestion was to make the frame itself foldable in the viewport, so that the user could click on a part of the frame and fold it by dragging it to the desired angle.

Suggestion: Make linking between folded and unfolded state easier

One participant with texturing experience was particularly interested in influencing the unfolding process and suggested to add weights to the different edges. The higher the weight of an edge would be, the higher it would be prioritized to stay a folding edge and not be cut.

When exporting the first test-object, a participant noticed that four of the frames look the same and drew attention to the fact that if they would not be exactly the same, complications may arise during the object assembly. A clear marking of the exported files, for example by also coloring them or writing the color in the name of the object, was suggested.

It is hard to tell similar frames in the export apart

We were also asked a few times, whether the current result would be printable or stable. Some functionality that checks for stability and printing restrictions might be helpful.

4.4 Discussion

The add-on was overall received very positive by the participants and they learned fast how to use it. Some participants asked to use it longer, because they had fun experimenting. However, a range of valuable suggestions was made and negative situations occurred. Next, we will present redesign possibilities that address them.

4.4.1 Preview Improvement

Based on the results of the study, we conclude that the preview functionality could be enhanced in the following ways:

Automatic Updating

Automatic updating would be ideal, if the updates were much faster

The manual refreshing seems to be the root cause for many of the observed negative situations. In the current implementation, the frames that get shown to the user are as complex as the frames that get exported. This results in a computational expensive updating operation, which makes live refreshing impossible. By reducing the computational cost of the updating operation, automatically updating the preview becomes realistic. The main portion of the computing time for a frame is used for the boolean operators that add male and female connectors to the frame.

Simplified previews could drastically decrease update duration

The female connectors are hard to see, since they are invisible in profile. The male connectors are very characteristic, but they do not need to be merged with the frame. Putting each male connector in a separate object that overlaps with the frame, looks the same to the user as merging them. This way the two expensive connector-merging operations could be skipped entirely. This approach makes it also possible to implement features like color coded connector types or moving connectors independently.

Parameter Updates

It was observed in 11 cases that participants used extreme values to see what a parameter does. By introducing a fast and automatic update, this behavior would be strongly supported, likely increasing the intuitiveness of the parameters. Another way to support this behavior would be to show additional translucent connectors when the user hovers over a parameter, which are animated in a sinus curve between minimal and maximal value of this parameter. This way, the user would not even need to change the current value of a parameter to see what it does.

Applying extreme values is an intuitive way to show what a parameter does

All Frames

Showing all the frames at once would likely be a huge improvement. Simplifying the preview as described would support that. The ideal way to present the frames according to the user suggestions would be to arrange them how they would connect later, and to color code them fitting to the region view (fig. 4.13).

Region View

The region view should not shuffle the colors when activated again. When color coding the frames, the region view should be on, too. An easy way to achieve this would be to replace it with a "Toggle Preview"-button. This button would activate the region view and create all the frames like described. As long as this mode is active, all the frames would update live. When deactivated, the region view would be turned off and the frames disappear.

The color coding that is introduced by the region view should be consolidated

Perspective

When implementing the preview changes, the camera should only switch perspective when the preview mode is activated. The focus should be on the center of all generated frames.

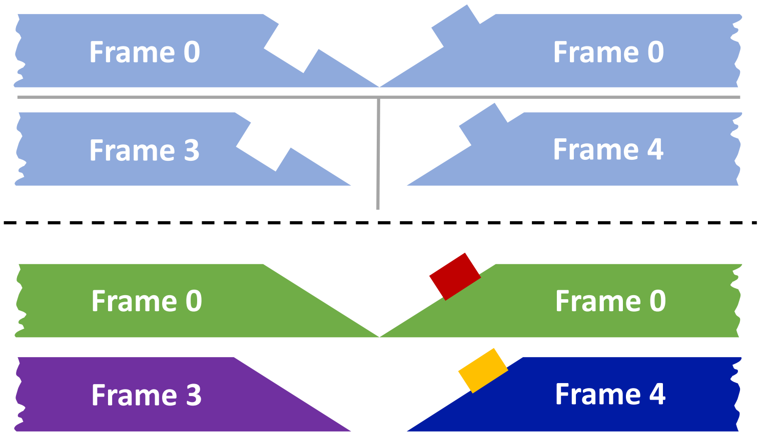

Figure 4.13: Top: Illustration of the current frame preview. Bottom: Illustration of the refined frame preview. All frames can be seen at once, they are color coded and connectors are colored depending on the edge type. Female connectors are left out.

The changes to the preview are not trivial to implement, but they could reduce or eliminate 85 of the 146 occurrences of negative characteristics observed in the study.

4.4.2 UI Improvement

Warnings should be more prominent

Regarding usability, it seems to be helpful to break with *Blenders* convention to show warnings in the info panel. Most users did not notice them or were confused by *Blenders* logging output, that got mixed with the warnings. Also a warning in the "Info"-panel does not disappear after it is resolved, since it is part of a log. A better solution could be to show the warnings directly in the add-on panel. The total frame number could also be shown in the add-on panel.

The custom warning panel could additionally be embedded in the export dialog to make sure that users do not export a faulty frame by accident. This might also help to draw the users attention to the export settings.

By implementing the proposed previewing strategy, the frame selector and the ”Create/Refresh”-button would not be needed anymore. The ”Toggle Region-View”-button could be replaced by a ”Toggle Preview”-button. The coloring of the object that indicates cutting regions should be given a more precise name, that is used consistently across the UI.

The frame preview should be coupled with the region view when color coding is used

The name of the button for suggesting settings should not change. It should express that the user gets suggestions by the program, not the other way around. A short introduction tutorial that explains the concept of *FabricFaces* and the parameters would help inexperienced users to skip the first slow part of the learning curve.

Those changes together with the new previewing strategy could reduce or eliminate 136 of the 146 occurrences of negative characteristics observed in the study.

Chapter 5

Summary and Future Work

In this last chapter a short summary is given and it is described what this thesis has contributed to the development of the *FabricFaces* method. We close by outlining promising avenues for future work.

5.1 Summary and Contributions

Our tool makes *FabricFaces* accessible, to unlock its innovative potential

The *FabricFaces* method introduces new design and prototyping possibilities to personal fabrication. It combines 3D printing on common 3D printers with fabric and allows for example to texture the surfaces and speed up the fabrication. In this thesis, we developed a tool that allows users to quickly compute the support structures that are essential for the *FabricFaces* approach. This is a major step for making the method accessible to a wide range of expert and non-expert users. We expect that by easing the creation of *FabricFaces* objects, innovation potential is unlocked to embed functionality or certain characteristics in objects, and to create new kinds of prototypes and objects with embedded functionality.

FabricFaces complements existing approaches

First, we gave an overview over several methods that aim at creating three dimensional shapes with two dimensional materials. We presented methods that use flat parts, pop-ups or folding based approaches. *Mill and Fold* by Muntoni et al. [2019a] was the approach that had the most similarities with *FabricFaces*. It shows the general feasibility of folding based approaches. Since *Mill and Fold* uses CNC-milling instead of 3D printing, it is faster but also more limited. *FabricFaces* works with all angles and includes connecting features. It well complements the existing methods with new use cases and advantages.

Our tool is easy to use, with optional expert features

Our tool was implemented in *Python* as an add-on for *Blender*. To offer an easy workflow, we implemented setting suggestions and previews of how the add-on divides the object and the computed support structure. Users also have the possibility to change the appearance of the connecting features, depending on how they will meet during assembly. They can manually adjust four basic parameters of the structure. The way how the object is split, can be controlled manually if required. A range of features was implemented that deal with *Blender* specific procedures or improve the workflow in general.

Users learned and understood our tool well

We have evaluated our tool in a qualitative user study. It was investigated how self-explaining the user interface is and how the users used the tool. Expectations, key observations and suggestions were compiled. Based on the study, we conclude that the tool is fast to learn and easy to use. Even users with no *Blender* experience were able to use it after a short teaching period. Finally, we proposed some design changes based on the study results, that would further increase the usability of the tool.

5.2 Future Work

Despite the good results of the user study, we identified room for improvement in some aspects of the add-on. Promising future work can be directly derived from the suggested design changes that were proposed when discussing the study results in section 4.4.

Suggestions for design adjustments in section 4.4

During the implementation, a wide range of glitches in the frame generation was detected and fixed. However, there are occasional cases where the result does not behave as expected. Those cases mostly occur on objects with very narrow faces or extreme angles, combined with a high frame thickness. In most cases, they disappear when changing the parameters.

No game-breaking glitches were found

Our implementation of this tool is a fully functional, expandable foundation for further features. In addition to the design adjustments derived from the study, promising features include:

Many promising features could be integrated in our tool

Connector Exceptions
The users would have much more freedom to chose connector parameters if they could create exceptions for single connectors, or at least delete them. If an object contains a single sharp angle, this limits the extrude value for the whole object.

Advanced Suggestions
The current method of our add-on for suggesting parameter values works well to give users meaningful starting values, but they are not based on experimental knowledge or material properties. The suggestion also delivers results that produce warnings, when the object contains sharp angles. It would be desirable to develop the suggestion system to a stage, that it automatically computes the optimal connector parameters for every edge.

Determining values for individual connectors is desirable

Smart Unfolding
One approach to improve the tool could be to learn from the *smart unfolding* presented by Muntoni et al. [2019a], the unfolding does not create a part that is as large as possible with some additional small parts, instead it creates as few parts as possible, with roughly the same area. This would allow to scale the object up more.

Curved Faces
3D printing can be used to control the bendability of fabric (Rivera et al. [2017]). This might allow to introduce stable curved surfaces to the design space of *FabricFaces*.

Packing
Controlling the packing allows additional helpful features

Another improvement could be to automatically pack the resulting frames for optimal print bed usage. This would allow helpful follow up features, like automatically scaling the object as large as possible while using a fix number of defined print bed sizes. It would also allow to create instructions for laser cutters to cut away the surrounding fabric or to texture the object by color-printing the unwrapped texture on the fabric.

String Actuated Folding
Structures that allow string actuated folding (Kilian et al. [2017]) could be added, to allow to fold objects just by pulling on a string.

The assembly of the generated objects should be evaluated

A study to evaluate the physical section of the process that uses the output of our tool, would likely result in valuable insights to improve it. Here, other unfolding algorithms (Hao et al. [2018]) could be tested to optimize assembly. Furthermore, additional undesirable characteristics could be found and counter-measured in the add-on during the design process.

Appendix A

USER STUDY DATA OVERVIEW

This is a list of all user study characteristics mentioned in the evaluation chapter. It contains expectations (table A.1), observations (table A.2, table A.3) and suggestions (table A.4).

Expectations	Category	No.
Simple usage (e.g. One button)	Workflow	10
Many things get done automatically	Workflow	7
Influence cut-edges when unfolding	Feature	5
Correct results	General	3
Integrated Tutorial/Explanation	UI	3
Similarities to UV unwrapping/Texturing-tools	General	3
Screw holes	Feature	2
Assembly instructions	Feature	2
Unfolding animation	Feature	2
Different unfold suggestions to choose from	Feature	1
Choice between different connector types	Feature	1

Table A.1: The expectations of the participants, before seeing or using the add-on. *No.* refers to the number of interviews it occurred in.

Rating	Observations (Part 1)	Category	No.
+	”Export settings” is intuitive	UI	12
0	Using extreme values to understand parameters	Behavior	11
+	Tooltips are helpful	UI	11
+	Overall intuitive/easy to understand/pleasant usage	General	11
+	Suggested settings are helpful/good/trustworthy	Workflow	10
-	Warnings/Information-output not realized	Workflow	10
+	”Scaling Factor” is intuitive	UI	10
-	”Offset” not understood/unintuitive/tooltip not helping	UI	9
+	”Region View” helpful	Preview	9
+	”Peak”-warning understood	Warning	8
+	Asterisks are helpful	UI	8
-	Settings in export window overlooked	UI	8
-	Count of frames is not easy to find out	UI	8
-	”Region View” not easy to understand	Preview	7
+	Asterisks intuitive/directly right understood	UI	6
0	To resolve ”peak”-warning: ”Extrude” used	Behavior	6
-	”Create/Refresh”-button not prominent enough	Behavior	6
-	Inconsistent terms: ”Region view”, ”ColorMap”	UI	6
-	Program output in Info-panel causes confusion	UI	6
+	Add-on is well integrated in Blender	General	6
+	”ColorMap”-option intuitive/well understood	UI	6
0	Obviously surprised by the amount of exported files	General	6
+	Triggered floating warning (all understood it)	Warning	5
+	Likes manual refreshing because it takes time	General	5
0	To resolve ”peak”-warning: ”Offset” used	Behavior	5
+	Few mandatory steps	Workflow	5
-	Did not like manual refreshing	Workflow	5
-	Did at first not understand frame selection process	Preview	5
+	Export used very intuitively	Export	5
-	Camera jumps when focusing on a detail and refreshing	Preview	5
0	Did not notice the tooltips	UI	5
-	Colors change each time when activating ”Region View”	Preview	5
-	Frame number wrapping irritates	UI	5
0	Enjoyed tinkering with connector blueprints	Behavior	5
0	Mistakes concave connectors for inter-frame connectors	Export	5
-	Expecting Perspective or Camera shift on ”Region View”	UI	5
+	Changes between frames intuitively/understands fast	Workflow	4
-	Mistakes asterisks for: Not the suggested value	UI	4
+	”Suggest Settings” stood out positively	Feature	4
-	Are all or just the current frame exported?	UI	4

Table A.2: The observations and comments that occurred during the study. *Rating* gives context about the implications of an observation. + means that it is good, - that it is bad and 0 that it is neutral, but interesting. *No.* refers to the number of interviews it occurred in. Part 1 of 2.

Rating	Observations (Part 2)	Category	No.
-	”Extrude” and ”Offset”, what is what?	UI	3
+	Obviously liked auto centering camera	Preview	3
+	Liked wrapping of the frame numbers	UI	3
-	In what unit are the values?	UI	3
-	”Select an Object”-warning/grayed out not noticed	UI	3
-	Cursor moved to ”Export All” instead of ”Refresh”	UI	3
-	Why exporting complete object es .stl and .obj?	Export	3
0	Expected refreshing when pressing enter	UI	3
-	Extrude is nozzle thickness of 3D printer	UI	3
+	Frames are sorted by complexity (all who noticed, liked that)	Feature	2
0	To resolve ”peak”-warning: editing connector blueprints used	Behavior	2
-	”Frames” can be mistaken for time frames	UI	2
0	Asterisks not noticed	UI	2
+	Noticed warnings by themselves	Warning	2
-	”Suggest Settings”: Suggested by me or to me?	UI	2
+	”Peak”-warning very helpful	Warning	2
-	Clicked ”Region View” instead of ”Refresh”	UI	2
-	Clicked ”Export All” instead of ”Refresh”	UI	2
-	”Height” and ”Extrude, what is what?	UI	1
+	Likes that colors change, each time they are activated	Preview	1
+	Parameters are reduced to the necessary ones	UI	1
-	Asterisks mean that a value is bad	UI	1
-	Button names should not change	UI	1
+	Noticed region-count info when ”region View” is activated	Preview	1
+	Understood Frame selection/switching immediately	Preview	1
-	Did not use ”Suggested Settings” when feasible	Behavior	1

Table A.3: The observations and comments that occurred during the study. *Rating* gives context about the implications of an observation. + means that it is good, - that it is bad and 0 that it is neutral, but interesting. *No.* refers to the number of interviews it occurred in. Part 2 of 2.

Suggestions	Category	No.
Color Frames according to ”region View”	Preview	9
Wrapping should happen live	UI	6
Frame counting should start from 1	UI	5
Mark bad situations locations in the 3D view	Preview	5
Show current scale when exporting to give reference	Export	5
Switching frames should automatically reload	UI	5
All frames should be shown as once	Preview	4
Warn when connectors stand out of the frame inwards	Warning	4
Color connectors on the frame according to their type	Preview	2
Allow separate female connectors, for clips.	Feature	2
Show extra dialogue if warnings are open before exporting	Warning	2
Turn Connector blueprints 180° around standing axis	Preview	2
Create all frames in background process while showing one.	Preview	1
Generate all frames, show one. To allow faster switching	Preview	1
Mark color of a frame also in UI. (E.g. Colored dot/Number)	UI	1
Assign frames to ColorMap to prevent mix-ups	Export	1
Do not cap values. Mark them red instead	UI	1
Highlight the fitting object face when hovering frame face	Feature	1
Hide connector parameters in advanced section	UI	1
Folding the frame with the by dragging in the viewport	Feature	1
Edge-Weights to formalize cutting preferences	Feature	1
Deactivate frame selector if there is only one	UI	1
Allow copying connectors E.g. basic -¿ intern	Feature	1
Different connector presets	Feature	1
Change ”Offset”-tooltip: ”... connectors inside the object”	UI	1
Check for stability	Feature	1

Table A.4: The suggestions by the participants, after or during using the add-on. *No.* refers to the number of interviews it occurred in.

Appendix B

EXAMPLE 3D PRINT

A quick low fidelity 3D print was made and assembled to verify, that the results of our tool can be assembled correctly. This was confirmed. The connectors match and the faces fit together (fig. B.1, fig. B.2).

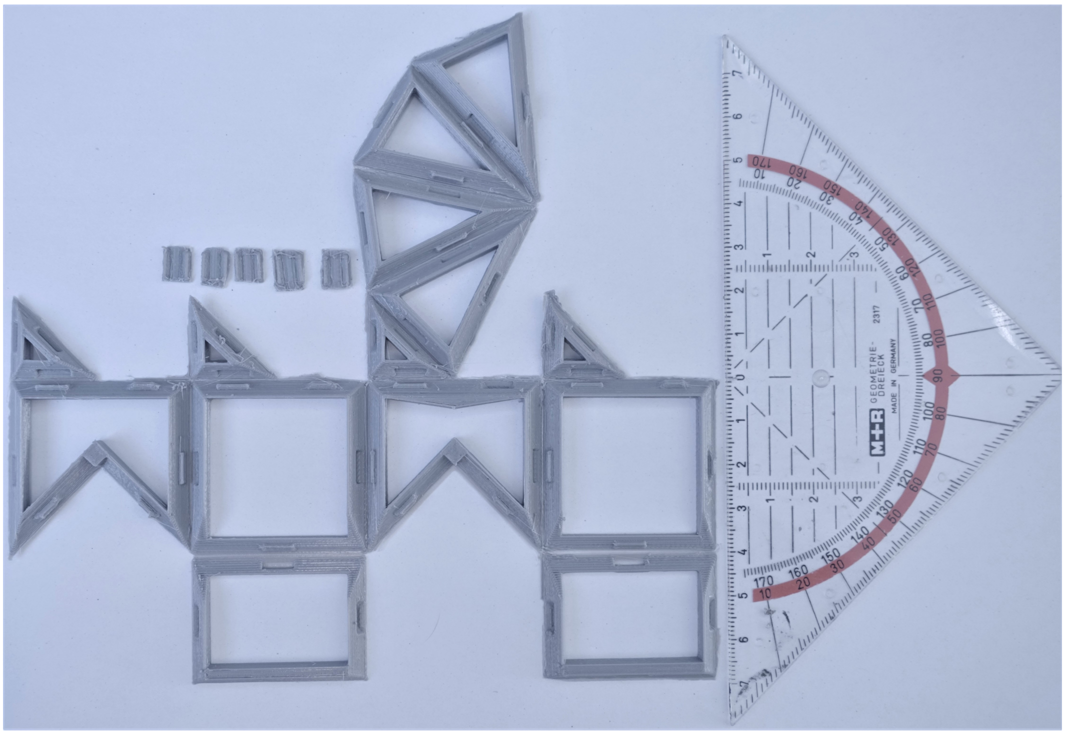

Figure B.1: Small low fidelity 3D print in the unfolded state. The five small parts are the extra connectors.

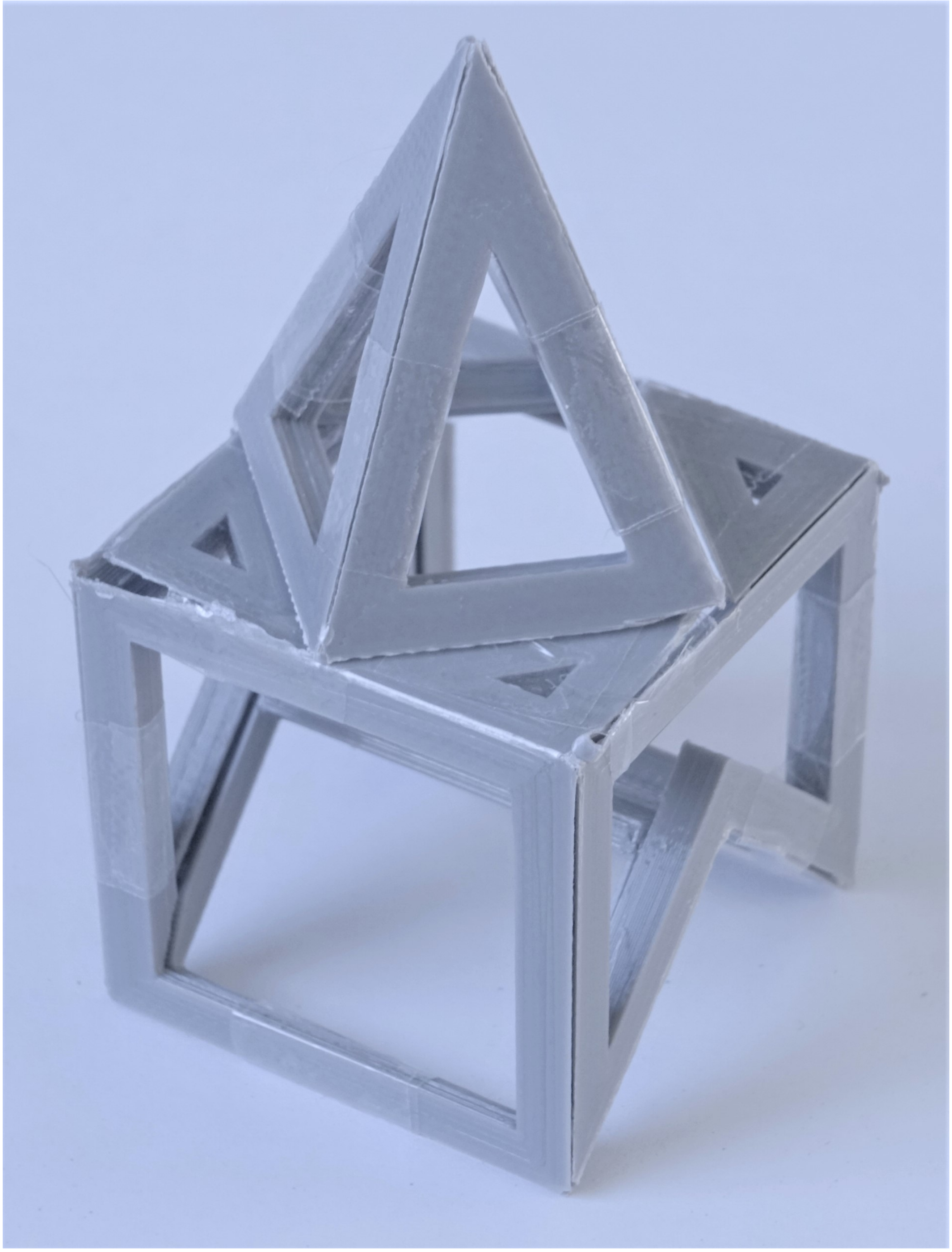

Figure B.2: Small low fidelity 3D print in the assembled state. It is 8cm high. It is not assembled optimally because of poor 3D print quality and a missing fabric surface to hold the pieces together. The objects can be assembled correctly, everything fits together.

Appendix C

QUESTIONNAIRE

This is the questionnaire that was used in the study. Like the rest of the study, it is in German.

Bitte dieses Dokument erst öffnen und ausfüllen, wenn du dazu aufgefordert wurdest.

Fragen zum Studien Beginn:

Demographie:

1. Alter: _______
2. Geschlecht: __________

Vorkenntnisse:
Kreuze hier bitte das zutreffendste an

3. Wie sicher fühlst du dich im Umgang mit Computern?
   [ ] Sicher [ ] Meistens Sicher [ ] Meistens Unsicher [ ] Unsicher
4. Wie häufig nutzt du 3D Modeling Software?
   [ ] In etwa wöchentlich [ ] In etwa monatlich [ ] In etwa jährlich
   [ ] Seltener als jährlich [ ] nie
5. Wie häufig nutzt du Blender?
   [ ] In etwa wöchentlich [ ] In etwa monatlich [ ] In etwa jährlich
   [ ] Seltener als jährlich [ ] nie
6. Falls du Blender bereits genutzt hast, wie gerne nutzt du es?
   [ ] sehr gerne [ ] gerne [ ] mittel [ ] ungerne [ ] sehr ungerne
7. Wie häufig nutzt du einen 3D Drucker?
   [ ] In etwa wöchentlich [ ] In etwa monatlich [ ] In etwa jährlich
   [ ] Seltener als jährlich [ ] nie

--- hier nicht weiter! ---

Figure C.1: Questionnaire page 1. At the beginning of the interview.

Fragen zum Studien Ende:

Kreuze hier bitte eine Zahl pro Frage an, je nachdem wie weit du mit der Aussage übereinstimmst.

Angenommen du hättest Zugriff auf 3D Druck und könntest Objekte im FabricFaces Stil grundsätzlich gebrauchen.

1. Ich denke ich würde dieses Produkt gerne häufig verwenden.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
2. Ich fand das Produkt unnötig kompliziert.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
3. Ich fand das Produkt einfach zu benutzen.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
4. Ich denke ich bräuchte die Unterstützung einer technisch bewanderten Person, um das Produkt zu nutzen.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
5. Ich fand, dass die verschiedenen Funktionen gut im Produkt integriert sind.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
6. Ich fand, dass es zu viele Inkonsistenzen in dem Produkt gab.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
7. Ich denke, dass die meisten Menschen dieses Produkt zügig erlernen würden.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
8. Ich fand es seltsam das Produkt zu verwenden.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
9. Ich war mir sehr sicher in der Handhabung des Produktes.
   Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*
10. Ich musste viel lernen bevor ich mit der Nutzung des Produktes beginnen konnte.
    Trifft nicht zu [ ] 1 [ ] 2 [ ] 3 [ ] 4 [ ] 5 *Trifft voll zu*

Anmerkungen:

Figure C.2: Questionnaire page 2. At the end of the interview.

Bibliography

Ergun Akleman, Shenyao Ke, You Wu, Negar Kalantar, AliReza Borhani, and Jianer Chen. Construction with physical version of quad-edge data structures. *Computers & Graphics*, 58:172–183, August 2016. ISSN 00978493. doi: 10.1016/j.cag.2016.05.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0097849316300516.

Aaron Bangor, Philip T. Kortum, and James T. Miller. An Empirical Evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction*, 24(6):574–594, July 2008. ISSN 1044-7318. doi: 10.1080/10447310802205776. URL https://doi.org/10.1080/10447310802205776. Number: 6 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447310802205776.

Patrick Baudisch, Arthur Silber, Yannis Kommana, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechlitz, and Thijs Roumen. Kyub: A 3D Editor for Modeling Sturdy Laser-Cut Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Glasgow Scotland Uk, May 2019. ACM. ISBN 978-1-4503-5970-2. doi: 10.1145/3290605.3300796. URL https://dl.acm.org/doi/10.1145/3290605.3300796.

Dustin Beyer, Serafima Gurevich, Stefanie Mueller, Hsiang-Ting Chen, and Patrick Baudisch. Platener: Low-Fidelity Fabrication of 3D Objects by Substituting 3D Print with Laser-Cut Plates. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, pages 1799–1806, Seoul, Republic of

Korea, 2015. ACM Press. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702225. URL http://dl.acm.org/citation.cfm?doid=2702123.2702225.

Bernd Bickel, Paolo Cignoni, Luigi Malomo, and Nico Pietroni. State of the Art on Stylized Fabrication. *Computer Graphics Forum*, 37(6):325–342, 2018. ISSN 1467-8659. doi: 10.1111/cgf.13327. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13327. Number: 6 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13327.

Jonathan D Cohen. Concepts and Algorithms for Polygonal Simplification. page 17, 1999.

David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational Shape Approximation. page 10, 2004.

Maas Goudswaard, Abel Abraham, Bruna Goveia da Rocha, Kristina Andersen, and Rong-Hao Liang. FabriClick: Interweaving Pushbuttons into Fabrics Using 3D Printing and Digital Embroidery. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pages 379–393. Association for Computing Machinery, New York, NY, USA, July 2020. ISBN 978-1-4503-6974-9. URL https://doi.org/10.1145/3357236.3395569.

Yue Hao, Yun-hyeong Kim, Zhonghua Xi, and Jyh-Ming Lien. Creating Foldable Polyhedral Nets Using Evolution Control. In *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, June 2018. ISBN 978-0-9923747-4-7. doi: 10.15607/RSS.2018.XIV.007. URL http://www.roboticsproceedings.org/rss14/p07.pdf.

Martin Kilian, Simon Flöry, Zhonggui Chen, Niloy J. Mitra, Alla Sheffer, and Helmut Pottmann. Curved folding. *ACM Transactions on Graphics*, 27(3):1–9, August 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360674. URL https://doi.org/10.1145/1360612.1360674.

Martin Kilian, Aron Monszpart, and Niloy J. Mitra. String Actuated Curved Folded Surfaces. *ACM Transactions on Graphics*, 36(3):1–13, July 2017. ISSN 0730-0301, 1557-7368. doi: 10.1145/3015460. URL https://dl.acm.org/doi/10.1145/3015460. Number: 3.

David H. Laidlaw, W. Benjamin Trumbore, and John F. Hughes. Constructive solid geometry for polyhedral objects. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, pages 161–170, New York, NY, USA, August 1986. Association for Computing Machinery. ISBN 978-0-89791-196-2. doi: 10.1145/15922.15904. URL `https://doi.org/10.1145/15922.15904`.

S. N. Le, S. Leow, T. Le-Nguyen, C. Ruiz, and K. Low. Surface and contour-preserving origamic architecture paper pop-ups. *IEEE Transactions on Visualization and Computer Graphics*, 20(2):276–288, February 2014. ISSN 1941-0506. doi: 10.1109/TVCG.2013.108. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

Xian-Ying Li, Chao-Hui Sheng, Shi-Sheng Huang, Tao Ju, and Shi-Min Hu. Popup: Automatic Paper Architectures from 3D Models. page 9, 2010.

William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, August 1987. ISSN 0097-8930. doi: 10.1145/37402.37422. URL `https://doi.org/10.1145/37402.37422`.

Sam Mattiussi. Fabric faces library: Designing connectors for foldable textile structures. Bachelor's thesis, RWTH Aachen University, Aachen, November 2020.

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, NY, USA, October 2014. Association for Computing Machinery. ISBN 978-1-4503-3069-5. doi: 10.1145/2642918.2647388. URL `https://doi.org/10.1145/2642918.2647388`.

Jun Mitani. Making Papercraft Toys from Meshes using Strip-based Approximate Unfolding. page 5, 2004.

Stefanie Mueller, Tobias Mohr, Kerstin Guenther, Johannes Frohnhofen, and Patrick Baudisch. faBrickation: fast 3D printing of functional objects by integrating construction kit building blocks. In *Proceedings of the*

SIGCHI Conference on Human Factors in Computing Systems, pages 3827–3834, Toronto Ontario Canada, April 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557005. URL `https://dl.acm.org/doi/10.1145/2556288.2557005`.

Alessandro Muntoni, Stefano Nuvoli, Andreas Scalas, Alessandro Tola, Luigi Malomo, and Riccardo Scateni. Mill and fold: Shape simplification for fabrication. *Computers & Graphics*, 80:17–28, May 2019a. ISSN 00978493. doi: 10.1016/j.cag.2019.03.003. URL `https://linkinghub.elsevier.com/retrieve/pii/S0097849319300263`.

Alessandro Muntoni, Lucio Davide Spano, and Riccardo Scateni. Split and Mill: User Assisted Height-field Block Decomposition for Fabrication. *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*, page 10 pages, 2019b. ISSN 2617-4855. doi: 10.2312/STAG.20191364. URL `https://diglib.eg.org/handle/10.2312/stag20191364`. Artwork Size: 10 pages ISBN: 9783038681007 Publisher: The Eurographics Association Version Number: 061-070.

C. D. Onal, R. J. Wood, and D. Rus. Towards printable robotics: Origami-inspired planar fabrication of three-dimensional mechanisms. In *2011 IEEE International Conference on Robotics and Automation*, pages 4608–4613, May 2011. doi: 10.1109/ICRA.2011.5980139. ISSN: 1050-4729.

Michael L. Rivera, Melissa Moukperian, Daniel Ashbrook, Jennifer Mankoff, and Scott E. Hudson. Stretching the Bounds of 3D Printing with Embedded Textiles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 497–508, Denver Colorado USA, May 2017. ACM. ISBN 978-1-4503-4655-9. doi: 10.1145/3025453.3025460. URL `https://dl.acm.org/doi/10.1145/3025453.3025460`.

Conrado R. Ruiz, Sang N. Le, Jinze Yu, and Kok-Lim Low. Multi-style paper pop-up designs from 3D models. *Computer Graphics Forum*, 33(2):487–496, 2014. ISSN 1467-8659. doi: https://doi.org/10.1111/cgf.12320. URL `https://onlinelibrary.wiley.com/`

doi/abs/10.1111/cgf.12320. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12320.

Yuliy Schwartzburg and Mark Pauly. Fabrication-aware Design with Intersecting Planar Pieces. *Computer Graphics Forum*, 32(2pt3):317–326, 2013. ISSN 1467-8659. doi: https://doi.org/10.1111/cgf.12051. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12051. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12051.

C. Shemelya, F. Cedillos, E. Aguilera, D. Espalin, D. Muse, R. Wicker, and E. MacDonald. Encapsulated Copper Wire and Copper Mesh Capacitive Sensing for 3-D Printing Applications. *IEEE Sensors Journal*, 15(2):1280–1286, February 2015. ISSN 1558-1748. doi: 10.1109/JSEN.2014.2356973. Conference Name: IEEE Sensors Journal.

Tao Shen and Yukari Nagai. An Overview of Folding Techniques in Architecture Design. *World Journal of Engineering and Technology*, 05(03):12–19, 2017. ISSN 2331-4222, 2331-4249. doi: 10.4236/wjet.2017.53B002. URL http://www.scirp.org/journal/doi.aspx?DOI=10.4236/wjet.2017.53B002. Number: 03.

Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. CofiFab: coarse-to-fine fabrication of large 3D objects. *ACM Transactions on Graphics*, 35(4):45:1–45:11, July 2016. ISSN 0730-0301. doi: 10.1145/2897824.2925876. URL https://doi.org/10.1145/2897824.2925876.

Tomohiro Tachi. Rigid-Foldable Thick Origami. page 11, 2011.

Karl Willis, Eric Brockmeyer, Scott Hudson, and Ivan Poupyrev. Printed optics: 3D printing of embedded optical elements for interactive devices. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pages 589–598, New York, NY, USA, October 2012. Association for Computing Machinery. ISBN 978-1-4503-1580-7. doi: 10.1145/2380116.2380190. URL https://doi.org/10.1145/2380116.2380190.

Yan Zhao, Yuta Sugiura, Mitsunori Tada, and Jun Mitani. InsTangible: A Tangible User Interface Combining Pop-up Cards with Conductive Ink Printing. In Nagisa Munekata, Itsuki Kunita, and Junichi Hoshino, editors, *Entertainment Computing – ICEC 2017*, Lecture Notes in Computer Science, pages 72–80, Cham, 2017. Springer International Publishing. ISBN 978-3-319-66715-7. doi: 10.1007/978-3-319-66715-7_8.

Index

Typeset April 11, 2021