

SoRoCAD 2.0: Extending a CAD Tool for Soft Robotics

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

by
Sarah Sahabi

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Jan Bender

Registration date: 11.12.2020
Submission date: 12.04.2021

Eidesstattliche Versicherung

Statutory Declaration in Lieu of an Oath

Name, Vorname/Last Name, First Name

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

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) Strafflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtet. 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

Abstract	xiii
Überblick	xv
Acknowledgements	xvii
Conventions	xix
1 Introduction	1
1.1 Soft Robotics	1
1.2 Outline of the Thesis	4
2 Related Work	5
2.1 Human-Soft Robot Interaction	5
2.2 Extending the Physics of Soft Robots	9
2.3 Tools for Designing, Fabricating and Simulating	12
3 SoRoCAD	17
3.1 General	17

3.1.1	Motivation & Intentions	17
3.1.2	The Basic Workflow	19
3.2	User Interaction Elements	20
3.3	Identifying Problems of the Current State . .	23
3.3.1	User Interface & User Interaction Problems	24
3.3.2	Distribution Issues	28
3.4	Extending SoRoCAD	29
3.4.1	User Interface & User Interaction Im- provements	29
3.4.2	Additional User Interaction	36
3.5	Distribution of SoRoCAD	38
3.5.1	Distribution Issues	39
3.5.2	The Run Script Build Phase Solution .	40
3.6	Summary	41
4	Technical Evaluation	43
4.1	Setup	43
4.2	Results	45
4.3	Discussion	50
4.4	Summary	56
5	Summary and Future Work	59
5.1	Summary and Contributions	59

5.2	Future Work	61
5.2.1	Results of the Technical Evaluation . .	61
	The Simulation Accuracy	61
	The Mould Design	62
5.2.2	User Interface & User Interaction . . .	63
	Bibliography	65
	Index	69

List of Figures

2.1	Movements of the helpless soft robot	7
2.2	Martinez et al. [2012]’s fabrication of soft robots with embedded paper	9
2.3	The multigait soft robot crawling and undulating to cross an obstacle	10
2.4	The Magic Ball before and after the vacuum is applied	11
2.5	Algorithm for designing soft pneumatic objects with desired rest shape and deformations	13
2.6	Huang et al. [2020]’s tool for simulating the locomotion of a limbed soft robot.	14
2.7	Components contained in the InflatBits Soft Robotics Toolkit	16
3.1	SoRoCAD’s template selection window	21
3.2	SoRoCAD’s <i>air channels</i> tab	22
3.3	SoRoCAD’s <i>silicone exterior</i> tab	23
3.4	SoRoCAD’s tutorial view	24
3.5	SoRoCAD’s simulation window	25

3.6	SoRoCAD's <i>mold</i> tab	26
3.7	The redesigned <i>air channels</i> tab	30
3.8	The redesigned <i>silicone exterior</i> tab	31
3.9	The redesigned template selection window .	32
3.10	The redesigned tutorial view	34
3.11	The redesigned simulation window	35
3.12	The redesigned <i>mold</i> tab	36
3.13	New functionalities enable the construction of more complex air channels	38
4.1	The experimental setup for the technical evaluation	45
4.2	Contrasting the real and the simulated en- largement actuator	47
4.3	Contrasting the real and the simulated bend- ing actuator	47
4.4	Contrasting the real and the simulated elon- gation actuator	49
4.5	Contrasting the real and the simulated S- curving actuator	49
4.6	Comparison of the smaller and the larger mould designs	51
4.7	The SoRoCAD-generated mould containing snake air channels, in cross section	52
4.8	The cross section of a redesigned mould . . .	54

List of Tables

- 4.1 The designs for the the fishbone-based actuators that were generated from the templates and used for evaluating SoRoCAD. 44

- 4.2 Comparing the percentual elongation, given by l_{max}/l_0 , and the expansion of the circumference c_{max}/c_0 of the real and the simulated enlargement actuators at the time of applying the maximal air pressure p_{max} before exploding. l_{max} denotes the maximal length immediately before the explosion and l_0 the initial length before the inflation. The values for the circumference changes are given analogously by c_{max} and C_0 . The type is given by the Shore value and the cube (C) or fishbone (F) air channel type. 46

- 4.3 Comparing the bending r_{max} and enlargement composed of the horizontal elongation l_{max}/l_0 and the expansion of the circumference c_{max}/c_0 as used before, of the real and the simulated bending actuators at the time of applying the maximal air pressure p_{max} before exploding. The type is given by the Shore value and the cube (C) or fishbone (F) air channel type. 48

Abstract

While in recent years the field of soft robotics has gained in importance, more and more people of different backgrounds take an interest in discovering the various possibilities of a soft robot. To obtain a general understanding of the actuated behaviour, the costly and complex process of self-fabricating different soft robots needs to be passed, including computer-aided modelling and 3D printing of the moulds, casting and curing, and testing of each model. In case a specific deformation is desired, this process has to be reiterated and slightly varied until accomplished. This paper presents *SoRoCAD*, a CAD tool that guides the user through the workflow of modelling soft pneumatic actuators from silicone, offers a 3D-printable model of the mould and simulates the deformation the actuator undergoes when pneumatically inflated. By this, various combinations of parameters and their impacts on the distortion of the actuator can be tried out in real time by users of any level of expertise to help gain intuition. The aim of this thesis is to further enhance the workflow when using *SoRoCAD* to design soft actuators, mainly in terms of ease of use and user interaction elements, with special regard to the inexperienced user. Present problems preventing the deployment of the tool involve the necessity to pre-install the *SOFA* framework, which handles physics-based simulations, as in *SoRoCAD*. This work describes how this problem can be resolved by expanding the project with run script build phases. Finally, this thesis aims at assessing and discussing the simulation's accuracy, as well as the feasibility of the template designs and the moulds provided by *SoRoCAD*, by conducting a technical evaluation.

Überblick

Während in den letzten Jahren der Bereich der Soft Robotik an Bedeutung gewonnen hat, interessieren sich immer mehr Menschen unterschiedlicher Hintergründe dafür, die Vielzahl an Möglichkeiten von Soft Robotern zu entdecken. Um ein allgemeines Verständnis für das mit Luftdruck angetriebene Verhalten zu erlangen, muss der kostspielige und komplexe Herstellungsprozess verschiedener Soft Roboter durchlaufen werden, einschließlich computergestützter Modellierung und 3D-Druck der Gussformen, das Gießen und Aushärten sowie Testen jedes Modells. Falls eine bestimmte Verformung erwünscht ist, muss dieser Vorgang so oft wiederholt und dabei leicht variiert werden, bis die gewünschte Verformung eingetreten ist. In dieser Arbeit wird *SoRoCAD* vorgestellt, ein CAD-Tool, das den Benutzer durch den Workflow der Modellierung weicher pneumatischer Aktoren aus Silikon führt, ein 3D-druckbares Modell der Form bietet und die Deformierung simuliert, die der Aktor beim pneumatischen Aufblasen erfährt. Auf diese Weise können verschiedene Kombinationen von Parametern und deren Auswirkungen auf die Verformung des Aktors in Echtzeit von Benutzern aller Kenntnisstände ausprobiert werden, um eine Intuition dafür zu erlangen. Das Ziel dieser Arbeit ist, den Workflow bei der Verwendung von *SoRoCAD* zum Entwerfen weicher Aktoren weiter zu entwickeln, hauptsächlich im Hinblick auf Benutzerfreundlichkeit und Benutzerinteraktion, und unter besonderer Berücksichtigung der unerfahrenen Benutzer. Bestehende Probleme, die die Bereitstellung des Tools verhindern, erfordern die Notwendigkeit, das *SOFA*-Framework vorzinstallieren, welches physikbasierte Simulationen, wie bei *SoRoCAD*, bearbeitet. In dieser Arbeit wird beschrieben, wie dieses Problem gelöst werden kann, indem das Projekt um *Run Script Build Phases* erweitert wird. Zudem zielt diese Arbeit darauf ab, die Genauigkeit der Simulation sowie die Nutzbarkeit der von *SoRoCAD* bereitgestellten Template Designs und Gussformen durch eine technische Evaluation zu bewerten und mögliche Verbesserungen zu diskutieren.

Acknowledgements

First, I would like to thank my supervisor, Anke Brocker, very much for everything she has done to help me improve my work. I appreciate your valuable advice and enjoyed working with you.

Also, I would like to sincerely thank everyone who helped me for taking the time.

Finally, I want to thank everyone at the i10 for the good times I had throughout this work.

Conventions

Throughout this thesis we use the following conventions:

Text conventions

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

EXCURSUS:

Excursus are detailed discussions of a particular point in a book, usually in an appendix, or digressions in a written text.

Definition:
Excursus

The whole thesis is written in British English.

Chapter 1

Introduction

1.1 Soft Robotics

In recent years, the field of Soft Robotics aroused more and more interest. This subfield of robotics covers robots that consist of highly compliant, soft materials, inspired by living, non-skeletal organisms, such as the tentacles of squids, caterpillars, or lizard's tongues [Ilievski et al., 2011]. The materials that are used to construct the main body of a soft robot are mostly elastomers, often silicone, and are characterised, amongst other attributes, by their Shore hardness [Rus and Tolley, 2015].

SHORE HARDNESS:

The Shore Hardness is a measure for the rigidity of a material, such as elastomer and rubber. The Shore A scale covers a range of 0-100, while 0 marks a very soft material, and 100 a hard one. An alternative measure for the hardness is the Young's modulus.

To actuate soft robots, there are various, common methods, including the pneumatic actuation through so-called 'PneuNets', which describe hollow channels within the actuator that are inflated when pouring in air pressure through valves [Ilievski et al., 2011]. Consequently, the actuator deforms in a certain way, depending on the asymmetry in the

The inspiration behind Soft Robotics

The soft robot's materials

Definition:
Shore Hardness

Common method of actuation: PneuNets

design, for example, caused by different thicknesses of the exterior walls and by the shape of their interior air channel, or other parameters, including the material's Shore hardness or the applied air pressure [Rus and Tolley, 2015].

Definition:
Actuator

ACTUATOR:

An actuator is the component of a robot that moves the joint. In this work, we only consider simple finger actuators.

Alternative actuation
methods

The advantages of PneuNets lie in their simplicity of fabrication, actuation, and control, as well as in their low cost and weight. Still, it is a relatively slow method of actuation. Therefore, Shepherd et al. [2013] discuss the alternative idea to actuate these PneuNets through explosions, triggered by chemical reactions. While, throughout this thesis, only pneumatically actuated soft robots that are made of silicone, will be looked at, other common actuation methods are hydraulic and electrostatic methods, as well as embedding tendons of variable lengths that imitate octopuses arms [Kim et al., 2013], [Rus and Tolley, 2015].

Soft robots have
some advantages
over rigid robots

Contrasting soft robots and traditional, rigid robots, there is a number of benefits and new application fields that are opened up by this new discipline, some of which will be pointed out hereafter. One of the main benefits of soft robots lies in their ability to perform rapid, agile manoeuvres and new classes of motion, due to the large number of degrees of freedom their bodies offer [Rus and Tolley, 2015]. Although rigid robots are efficient and precise in performing preprogrammed, repetitive tasks, their adaptability is often limited. Soft robots, on the other hand, comply with their environment which allows for the manipulation of unidentified objects, as well as the flexible locomotion on unknown, irregular terrain, which makes them suitable, amongst other applications, for disaster response [Rus and Tolley, 2015], [Kim et al., 2013].

The application field
of disaster response

An important
application:
Human-Robot
Interaction

Another important application field of soft robots is the interaction with humans. Because the collision with a rigid robot can cause serious harm to the human, their workspaces are mostly separated. However, there are sev-

eral applications that could exploit close interactions of the human and the robot, including medical uses, such as drug therapy and surgery, or using collaborative robots for human assistance, for example in the industry [Kim et al., 2013], [Milthers et al., 2019]. Because of the soft robot's compliant body, a collision would only deform the robot, absorbing most of the collision's energy and, therefore, pose a significantly smaller risk to the human which enables the use of robots in the previously mentioned interactive applications [Rus and Tolley, 2015].

Besides its benefits, there are limitations to soft robots, nonetheless, including typically small sizes to handle their own body weight without a skeleton, and a limited actuation and moving speed [Kim et al., 2013]. Moreover, most of the common actuation methods that were presented before, rely on air compressors, batteries, or other heavy components for the actuation, which limits the number of applications [Rus and Tolley, 2015]. There are already several approaches addressing these problems, however, this goes beyond the scope of this work and will therefore not be discussed. For further information, we refer to the work of Stokes et al. [2014].

Since there are various parameters that affect the soft robot's behaviour when inflated, another problem that occurs in the context of Soft Robotics is the seeming unpredictability of the actuator's exact deformation from the novices' view. For the beginner to learn and explore the possibilities of motion and, more than that, to understand the different parameters' influences, the time-consuming and complex process of fabricating soft robots of varied designs must be gone through to observe and contrast their behaviours. The fabrication of those soft robots, that are dealt with throughout this thesis, comprises several steps, starting with the modelling and 3D printing of suitable moulds, often with the help of CAD software. This is followed by the casting, curing and glueing of the actuator halves with silicone, and, finally, by the pneumatic actuation to test their motions. Especially the attempt to obtain desired deformations consists of a lot of trial and error, as the costly process has to be reiterated, while slightly varying the parameters, until its accomplishment. As a so-

Limitations of soft robots

Learning the impacts of different parameters is hard for the beginner

The fabrication process

SoRoCAD as a
learning tool

lution to this problem, SoRoCAD was developed, which combines the assistance to design a soft finger actuator, the simulation of its actuated behaviour, and their exportable, auto-generated moulds in one simple tool. Besides its aim of saving the users a lot of time by sparing the repeatedly passing of the fabrication process, SoRoCAD's focus lies on supporting the user on the journey of discovering the basic principles of a soft robot's behaviour by providing assisting tutorials and tips.

1.2 Outline of the Thesis

After the first chapter briefly introduced the field of Soft Robotics and the SoRoCAD tool, which is the main focus of this work, the second chapter presents some related work that relate to the topic of this thesis. Chapter three deals with the tool itself, describing its previous state and contrasting it with the present version, after modifications were executed to solve existing problems. The technical evaluation of the tool's accuracy and practicability is outlined in chapter four, including its setup and results, as well as the discussion of the outcomes and suggested improvements based on the findings. The final chapter sums up this work's main contents and results and reveals further ideas for future work.

Chapter 2

Related Work

As described in chapter 1 'Introduction', Soft Robotics is a growing field that finds use in more and more areas. Among its various applications, its principle also serves as inspiration for other fields, such as interaction design, and might, therefore, become relevant for all humans in the future. Thus, besides spreading a basic understanding of its principles to a wider audience, the question of how soft robots are perceived and interpreted by the human is worth an investigation. This chapter first presents recent work on this question, as well as an approach to creating more naturalness to the interaction, and then points out why this is crucial for realising desirable Human-Robot Interactions. Subsequently, further work on extending a soft robot's capacities by changing its physics or adding features will be described and contrasted. Finally, three different tools for designing, fabricating and simulating will be introduced and compared to each other, with special regards to similarities and differences to the SoRoCAD tool, which will be the main focus of this thesis.

2.1 Human-Soft Robot Interaction

Recent research on Soft Robotics revolutionises the area of interaction design by breaking the previous notion of a sep-

Soft Robotics as
inspiration for
interaction design

aration of static, rigid hardware and dynamic, adaptable software interfaces [Gohlke, 2017]. Gohlke [2017] looked further into how the usage of soft fluidic actuators, known from the field of Soft Robotics, could enhance research on tangible user interfaces. They found that using soft structures could create a new category of user interfaces that allow dynamic and controlled shape changes through actuation, similar to the physical deformation of soft robotic actuators. To further investigate the potential of shape-changing user interfaces, they explored the range of patterns of shape change that fluidic structures could achieve, including extension, contraction, bending, curling, torsion and rotation, as well as dynamic structure modification.

However, besides its uses and application potentials, its limitations need to be taken into account when implementing Soft Robotics principles in interaction design. As discussed in chapter 1 'Introduction', common actuation methods, including pneumatic, hydraulic, and electrostatic actuation, mostly rely on large and heavy components, such as air compressors, that impede mobile uses. Additionally, these actuation methods are not suitable for applying more fine-grained and faster shape changes and must therefore be adapted [Gohlke, 2017].

Human perception of
soft robot's
naturalness and
appeal

In chapter 1 'Introduction' it was mentioned that one of the most interesting uses of Soft Robotics is for interaction with humans doing collaborative or assistance work. As this field gains more importance and can even be applied to other fields as stated previously, it becomes crucial to do further research on Human-Soft Robotic Interaction, more specifically on human perceptions of soft robots, to facilitate interactions for humans and make them feel less uncomfortable. Thus, Jørgensen [2018] scrutinised how soft robots are perceived by their study participants, based on the theory, that there is a correlation between the participant's rating of the soft robot's naturalness and its appeal. Their study, in which the participants were asked to interact with a soft robotic tentacle and fill a questionnaire afterwards, supported the assumption that because of their soft and fluid movements, soft robots are perceived as more natural and, as a result, more appealing.



Figure 2.1: Two movements of the helpless soft robot, gripping and reaching. Human collaboration is needed.

While this finding emphasises the importance of a soft robot's appeal both visually and physically, Milthers et al. [2019] investigated the human interpretation of a soft robot's actions which is just as essential to advance Human-Soft Robotic collaborations. To find out whether the movements of a soft robot were sufficient for the human to interpret the robot's goals correctly and to further examine his willingness to engage in collaboration, two movement patterns were used. In an in-the-wild-study, Milthers et al. [2019] placed a three-legged soft gripper at a distance from a ball and simulated it failing at trying to reach the ball by first repeatedly gripping only air with all three legs being actuated at the same time, and second, by actuating only one leg strongly, mimicking human reaching behaviour to provoke compassion in passers-by (see Fig. 2.1). The random passers-by's reactions were then compared for both movements, in each case with or without an additional cue in the form of a 'Help me' sign placed right next to the constellation. The results show that interactions by humans were only rarely triggered without an additional signifier since most passers-by did not go beyond stopping and watching when no sign was visible. However, even with the sign placed next to the robot, the number of helping interactions was only slightly more, but with a significant increase for the second reaching movement, indicating that empathy does play a major role in Human-Soft Robotic Interaction.

Human interpretation
of a soft robot's
actions

Combining the findings of the presented work, it is es-

Adding sounds to
soft robots

essential to create soft robots as natural as possible to maximise human empathy and the willingness to interact with them. Hence, Bering Christiansen and Jørgensen [2020] presented an approach where, besides creating an organic look, sounds are added to the soft robot since coupling two sensuous modalities are known to improve communication and understanding of objects, therefore enabling more complex interactions. For this purpose, the sounds used in films for imaginary creatures with consistencies and motions that are similar to those of soft robots were analysed and added to the robots. Because no studies or further evaluations were conducted, the impact of adding sounds to human interpretation of soft robot actions is yet to be explored, but, due to our results from previous work presented, is a promising approach for advancing Human-Soft Robotic collaborations.

Contribution to this
thesis

While by now only very little research has been made on human perception and empathy for soft robots, this topic will gain more and more relevance since Soft Robotics might be the answer to many previous Human-Robot Interaction challenges and understanding human conceptions is indispensable for enabling natural and pleasing interactions [Milthers et al., 2019]. The presented papers demonstrate that the field is growing and becoming relevant not only for those who are involved in the metier of general robotics, but also to uncountable other people for everyday use, such as for tangible user interfaces, as introduced by Gohlke [2017], or as collaborative robots in many fields, as mentioned by Milthers et al. [2019]. A step towards spreading the use of soft robots in everyday situations to everyone was taken by Jørgensen [2018] and Bering Christiansen and Jørgensen [2020], who realised and emphasised the potential of Soft Robotics, as well as the importance of making them as compatible with the human as possible. Hence, the related work introduced in this paper stress the need to pass on knowledge on Soft Robotics to help more people of not only technical backgrounds understand its underlying principles, which is taken up by the SoRoCAD tool presented in this paper.

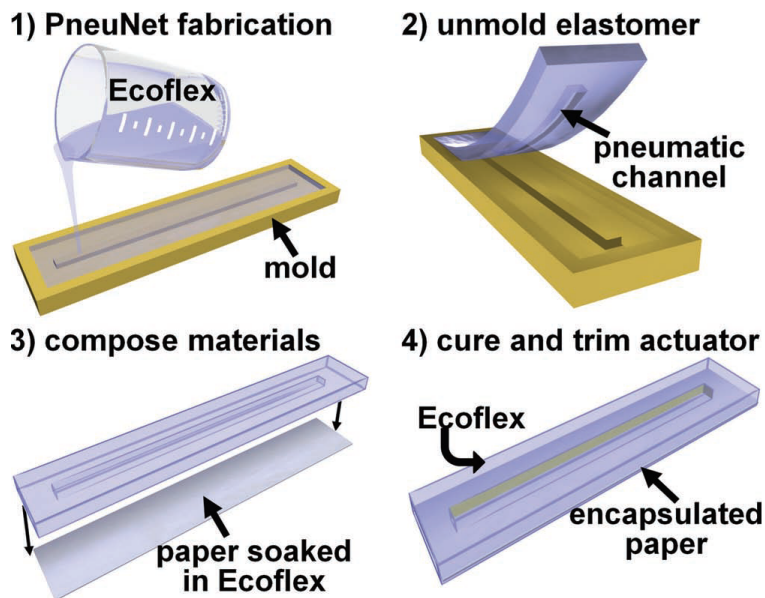


Figure 2.2: Martinez et al. [2012]’s fabrication of soft robots with embedded paper.

2.2 Extending the Physics of Soft Robots

Besides safe use in Human-Robot Interactions, an important benefit of soft robots is their capabilities of motion as presented in 1 ‘Introduction’. Considering the number of use cases of a soft robot, another interesting field of research deals with how the physics of a soft robot can be extended to allow new and more complex functionalities. Martinez et al. [2012], for instance, depicts an approach where an origami-based folded paper is embedded in elastomers, as shown in Fig. 2.2, to create motions that are determined by the folding-pattern of the paper when inflated. This enables a wider range of controlled motions, including anisotropic responses because the paper within limits the extension but hardly the bending.

Other than bending, elongating, and twisting, even more sophisticated movements can be provoked to further expand the fields of application. Hence, Shepherd et al. [2011] invented a new class of locomotive soft robots that allow multiple gaits: the quadrupedal robot with its indepen-

Origami-based
paper-elastomer
composites

Obstacle-crossing
multigait robot

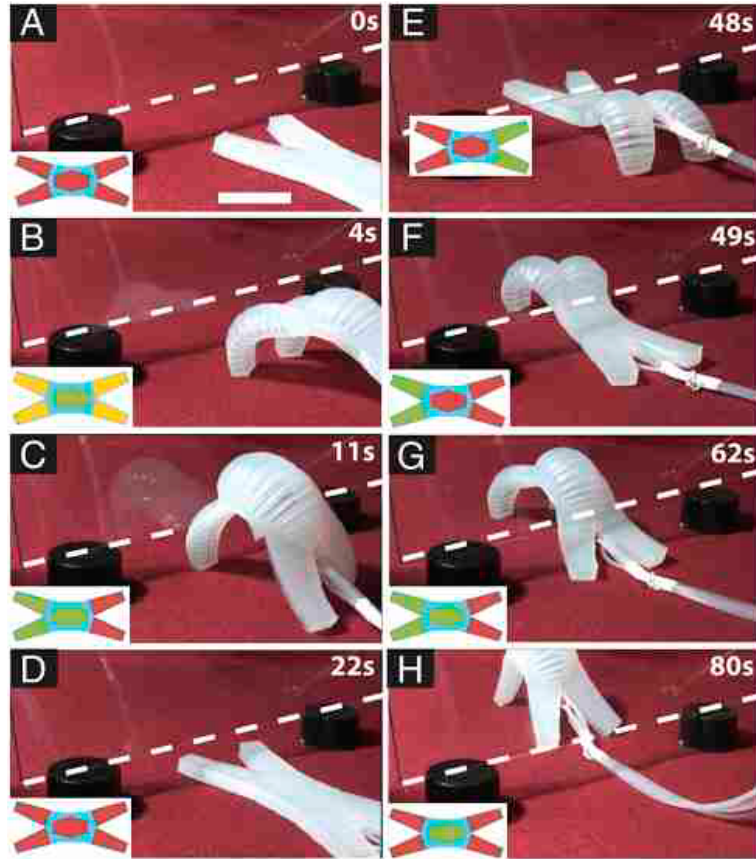


Figure 2.3: The multigait soft robot crawling towards the obstacle (A-C), undulating underneath it (D-G) and then crawling away (H). The schemes show the actuation.

dent pneumatic channels for each leg and one for the spine can, depending on the pressure, both crawl and undulate, which realises the crossing of obstacles by relatively simple means (see Fig. 2.3). Although Martinez et al. [2012]’s work focused on expanding the range of motions with the highest possible efficiency regarding the weight, cost, and complexity in manufacturing, whereas Shepherd et al. [2011] present new types of locomotion to cross obstacles, both address similar uses. Because of the soft robot’s damage-resisting nature, and additionally, because these works enable complex movements, potential applications for both paper-elastomer composites and multigait soft robots are uses that involve moving in complex and rocky environ-

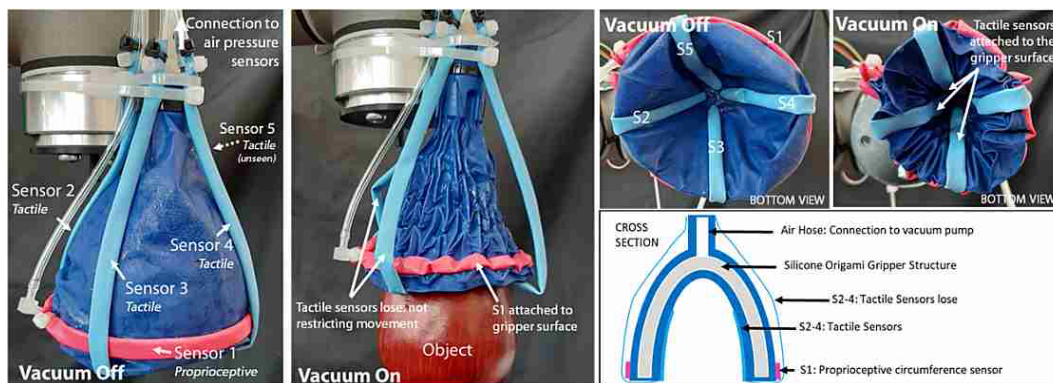


Figure 2.4: The Magic Ball before and after the vacuum is applied, showing the tactile and proprioceptive sensors.

ments, such as for disaster control [Martinez et al., 2012], [Shepherd et al., 2011].

Since another widely spread application field of Soft Robotics is in the industry, serving as a gripper for example for packaging or recycling, there is a considerable amount of research going on to create a ‘universal gripper’. This gripper can sense and identify the objects it is grasping and enables finer controls according to the item’s individual properties, such as its weight, shape, size and fragility [Hughes et al., 2020]. Moreover, to improve accuracy when grasping items, another goal for the ideal gripper is precise proprioceptive sensing, i.e. the robot is aware of the position and movements of its own body [She et al., 2019].

The universal gripper with tactile and proprioceptive sensing

Hughes et al. [2020] and She et al. [2019] each presented an approach for creating the universal gripper with both tactile and proprioceptive sensing. Hughes et al. [2020] introduce the ‘Magic Ball’, a dome-shaped skin made of flexible latex and an origami-inspired skeleton, that collapses around the object and conforms to its shape when a vacuum is applied inside. Therefore, a wide range of objects of different sizes and shapes can be gripped. Additionally, the Magic Ball is extended by sensors to first, measure and remember changes in the air pressure of the gripper to classify objects accordingly, and second, for proprioceptive sensing (see Fig. 2.4). An algorithm uses the classification data to adjust the grasping behaviour appropriately. She

et al. [2019], on the other hand, created ‘GelFlex’, a vision-based proprioceptive and tactile sensing gripper consisting of exoskeleton-covered soft fingers. This gripper contains two cameras and trains neural networks to estimate the shape of the finger and to classify the size and shape of the object, based on the camera-data obtained.

The vast potential of extending the physics of soft robots and adding features to their body catches the attention of many researchers. Adapting the shape and the internal structure offers new categories of shape change and locomotion, inter alia Shepherd et al. [2011]’s multigait robot or Martinez et al. [2012]’s origami-determined deformations. Further possibilities are opened up by creating soft robots with tactile and proprioceptive sensing abilities, either through sensors as described by Hughes et al. [2020] or through cameras, as presented by She et al. [2019]. Similar to the SoRoCAD tool, which is subject of this work, the papers introduced in this section all deal with exploring how changes to the physics of a soft robot can affect its behaviour and enable additional possibilities. However, in contrast to SoRoCAD, these works focus on realising defined objectives, such as Shepherd et al. [2011]’s research on enabling multiple gaits, or Hughes et al. [2020] and She et al. [2019]’s approaches on developing the universal gripper. SoRoCAD, on the other hand, works the other way around, by allowing the user to adapt the actuator’s design and to observe the resulting deformations. The main focus lies, here, in the learning and exploring, other than accomplishing specific tasks.

Contrasting the tools
with SoRoCAD

2.3 Tools for Designing, Fabricating and Simulating

The usage of Soft Robotics is without a doubt indispensable in tomorrow’s world, due to its huge number of benefits, some of which were mentioned in chapter 1 ‘Introduction’. However, as it is a relatively new field and it is based on different principles than rigid robots with joints, existing tools and algorithms for designing, fabricating and

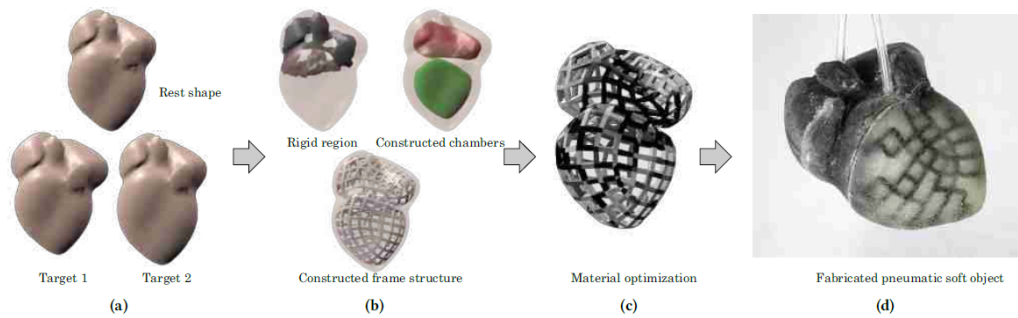


Figure 2.5: Algorithm for designing soft pneumatic objects with optimal chamber structure and material distribution (b-d) for desired rest and deformed shapes (a).

simulating robots cannot be applied to soft robots [Rus and Tolley, 2015]. For example, considering simple finger actuators made of silicone, as the ones that will be looked at throughout this work, designing them through trial and error, without the help of additional tools, is not trivial, either, and takes a lot of time, assuming that the designed and fabricated soft robot will not turn out as desired on the first try. As a result, multiple iterations of the same time-consuming process of modelling the moulds, 3D printing them, silicone casting, and curing must be executed [Kopic and Gohlke, 2016]. Therefore, it makes sense to develop tools that assist the user with the design and fabrication process or to provide accurate simulations so, in experimental uses, the actual fabrication can even be omitted.

Ma et al. [2017] present an algorithm that requires the models of an object as well as its deformed condition as input and then computes the optimal geometric layout of the chambers within the given object, the object's frame structure, material properties, and the needed air pressure (see Fig. 2.5). The user can then immediately 3D print the respective model and actuate it to obtain the desired deformation. Similarly to SoRoCAD, this tool aims at simplifying the costly fabrication process of an actuator by reducing the number of frequent errors. However, its main objective is to support professional designers of soft robots for particular uses with well-known deformations by providing optimised designs regarding fabrication materials and manufacturing cost. SoRoCAD, in contrast, is a learning tool for absolute Soft Robotics beginners and is intended to

Algorithm for
designing soft
pneumatic objects
with desired
deformations

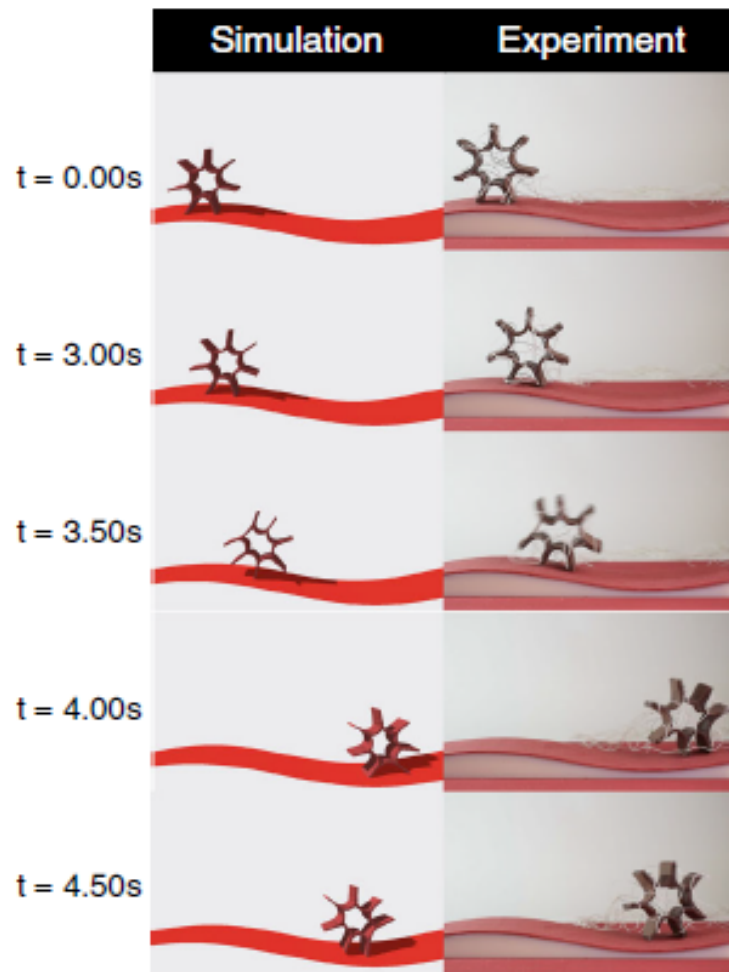


Figure 2.6: Huang et al. [2020]’s tool for simulating the locomotion of a limbed soft robot. The simulation in time steps is shown on the left, the real position and deformation for the same times are depicted on the right.

be used for experimenting with different actuators to get a feel for the parameters and their effects on the resulting deformation.

Numerical framework
simulates the
locomotion of the soft
robot

Another computational framework for quick experimental analysis or to help to avoid trial and error when searching for the right design was introduced by Huang et al. [2020]. This framework simulates the locomotion of limbed

soft robots while considering realistic environmental conditions, such as the involved materials' elasticity, actuation parameters, friction, (in)elastic collisions, forces, and energies. Other than for Ma et al. [2017]'s algorithm, this framework does not require knowledge of the resulting deformation and motion in advance. Still, due to a large number of parameters that need to be declared by the user beforehand, some of which extensively measured, it is not well suited as a Soft Robotics learning program for beginners. In contrast to SoRoCAD, this tool is, hence, considered as decision-support for long-term projects for real or experimental uses. Besides that, the precision of the simulation by far exceeds a learner's needs by not only predicting the robot's deformation when inflated but also considering its locomotion in accordance with realistic physical effects. Even though this is useful to relate the observations to the underlying physical principles [Huang et al., 2020], the absolute beginner might easily be overwhelmed.

However, the Inflatibits Soft Robotics Toolkit presented by Kopic and Gohlke [2016] and SoRoCAD share the same intention of teaching a wider audience the basic principles of Soft Robotics and their actuation behaviour. The toolkit contains air chambers, air connectors, restrictors and other useful components for building simple soft robots without the need for additional tools or prior knowledge (see Fig. 2.7). Having all the necessary elements readily available, the casting and curing process can again be skipped and testing around with different constellations to gain experience is faster and less frustrating since the user gets instant feedback and the robot can easily be adapted. Still, as there is only a certain number of components involved, the set of constructible robots is rather limited in both the number and the complexity, wherefore the learning effect does not go much beyond a beginner's level. In SoRoCAD, on the other hand, the actuators can be varied in many parameters, some of which can be newly combined and modified in their sizes, orientation, and more. It, hence, allows for more freely composed and complex actuators while also providing assistance in the form of templates and tutorials to improve the learning.

Inflatibits Soft
Robotics Toolkit

Moreover, the difference between hardware and software

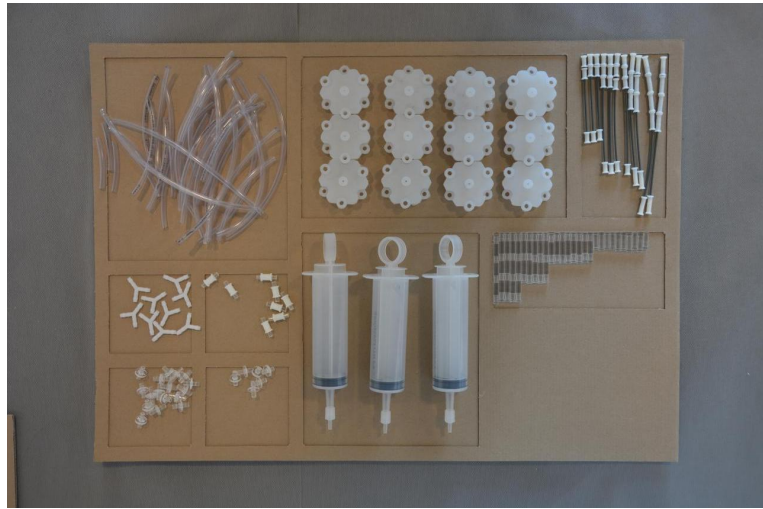


Figure 2.7: Components contained in the Inflatibits Soft Robotics Toolkit

must be taken into account when comparing previously mentioned tools and Kopic and Gohlke [2016]’s Inflatibits Toolkit. Whereas the SoRoCAD tool, as well as Ma et al. [2017]’s and Huang et al. [2020]’s tools require at least basic computer skills and might still take the user some time to learn, Inflatibits requires nothing but the toolkit itself and is therefore even suitable for children. Nevertheless, a software tool has benefits, too, like the fact that there are no limited or sensible resources and the robot can be modified quickly and easily.

Chapter 3

SoRoCAD

After presenting some related work that is relevant for this thesis, SoRoCAD and its motivation and intentions, the workflow, its user interaction elements and current limitations, will be introduced, as well as several approaches to extend the tool.

3.1 General

The subject of this thesis is the SoRoCAD tool and follows up on Strüver [2020]’s work. SoRoCAD is an abbreviation for Soft Robotics computer-aided design and is the name of a tool for modelling and simulating simple soft robotic actuators of silicone and their deformations. Additionally, the tool generates ready to use moulds of the designed models, which simplifies a typically complex task.

3.1.1 Motivation & Intentions

In contrast to traditional, rigid robots that need to be programmed to perform specific operations, the movement directions of soft robots are specified through their physical characteristics, which allows them to move dynamically. Despite being one of the main benefits of soft robots, this

The problems that
Soft Robotics
novices often face

poses a problem, especially for the novices at Soft Robotics, since the exact deformation is not easy to predict. The only way for these novices to become acquainted with Soft Robotics is through trial and error. However, the process of building a fully functional soft robot from silicone, which is the type that will be considered throughout this thesis, requires a high expenditure of time and material, as well as a lot of effort, primarily for beginners. The common procedure involves first the generation or modelling of the mould, which needs to be 3D printed afterwards. For this, depending on the complexity of the actuator, at least basic CAD-knowledge is required, as well as access to a 3D printer, and several hours must be taken into account for the printing. The expense and effort of the following casting and curing with silicone vary according to the geometry of the mould in which the actuator is being cast and the viscosity of the uncured silicone used. Indeed, several problems might occur during this step, including flaws in the resulting actuator that were caused by little bubbles, too thin walls or too hard silicones. Last, the cured halves of the actuator can be assembled by carefully coating them with the same silicone. After the time needed to cure again, the soft robot is finally ready to be tested, having taken the user up to two days in total for the production. In case the user has a specific deformation in mind that he wants to achieve, he has to iterate through the whole process all over again until he succeeds. Therefore, the user cannot realise all ideas, as the effort needed for the fabrication of each robot, possibly including some that might be not functional, is too much to be worthwhile. Particularly the novices who lack the experience to assess the significance of an idea might need numerous attempts until the desired deformation is achieved.

The motivation
behind SoRoCAD is
to support Soft
Robotics novices

To help users to get started with Soft Robotics, SoRoCAD was created. This tool allows the modelling of simple silicone actuators, the simulation of their deformations and the auto-generation of the respective moulds that can be used after 3D printed. Hence, the user can quickly and easily try around and explore how the parameters affect the actuator's deformation while being spared from the time- and cost-intensive process of fabricating a soft robot. Thus, besides the fact that the user is no longer limited in the num-

ber of ideas that can be modelled and simulated to investigate different deformations, he is not limited to creating actuators that are realisable in their size or hardness, either. Moreover, in case the user wants to fabricate a soft robot with a desired deformation, he has to go through the time-consuming process only once, as miscalculations of the resulting behaviour are fully avoided by pre-checking the simulation. Additionally, neither prior CAD- nor Soft Robotics-knowledge is required any longer, since SoRoCAD replaces the step of manually designing and generating models of the mould via CAD software by providing automatically generated ready to use models with reasonable designs.

To retain the target group of novices in both Soft Robotics and CAD-modelling, the user interface, as well as the available functionalities in the tool, are kept simple. The buildable actuators are therefore limited to elementary designs, resulting in the effects of the parameters on the deformation, such as the relation between different wall thicknesses, the design and volume of the air channels, the material's Shore Hardness and the applied air pressure, becoming apparent. Besides the learning effect that can be achieved through trial and error using the tool, additional assisting cues are provided by SoRoCAD, imparting basic Soft Robotics principles, such as how a bending motion can be attained.

The goals of
SoRoCAD

Having considered all these points throughout the implementation, SoRoCAD incorporates solutions to many of the earlier mentioned, common problems that especially Soft Robotics novices are facing. Its target user is the beginner in both Soft Robotics and CAD-modelling and it aims at helping these users to overcome their inhibitions to confront themselves with the new field of robotics by providing a simple user interface and subtle hints.

Target users of the
tool

3.1.2 The Basic Workflow

Before identifying the common problems that occurred using the SoRoCAD tool during the user studies, conducted

by Strüver [2020], the workflow that a user goes through to build and simulate a soft robot will be briefly presented in the following.

The *air channels* tab

The tool comprises three main tabs in total, namely the *air channels*, *silicone exterior*, and *mold* tabs, each representing one of the three elementary steps in the workflow of designing an actuator with its inner air chamber structure and silicone exterior. When starting SoRoCAD, the user can either select a template for predefined motions or he can start modelling the interior of the actuator, consisting of air chambers. Next, he changes to the *silicone exterior* tab, which is the same tab he is immediately prompted to when selecting a template in the beginning.

The *silicone exterior* tab

In the *silicone exterior* tab, possibly containing the respective soft robot that was automatically built from the template, the silicone exterior, i.e. the silicone wall thicknesses of the actuator, can be adapted. The soft robot can then immediately be simulated for a user-defined material's hardness and air pressure, to show the actuated behaviour. After that, the user can still make modifications to adapt the deformation while keeping the inner structure.

The *mold* tab

If the user wants to fabricate the soft robot, the third tab provides a fully generated mould of the previously designed actuator that can be 3D printed and cast.

3.2 User Interaction Elements

This section offers a deeper insight into SoRoCAD's user interaction elements and further functionalities that have not been presented previously.

The template selection window

As mentioned earlier, when starting SoRoCAD, the first window to open up is the template selection window (see Fig. 3.1), where the user can either select one of the three provided templates for immediately obtaining a ready for use actuator that behaves in a predefined way, or he can open an empty project to realise his ideas. The three different motions, that can be attained through the templates,

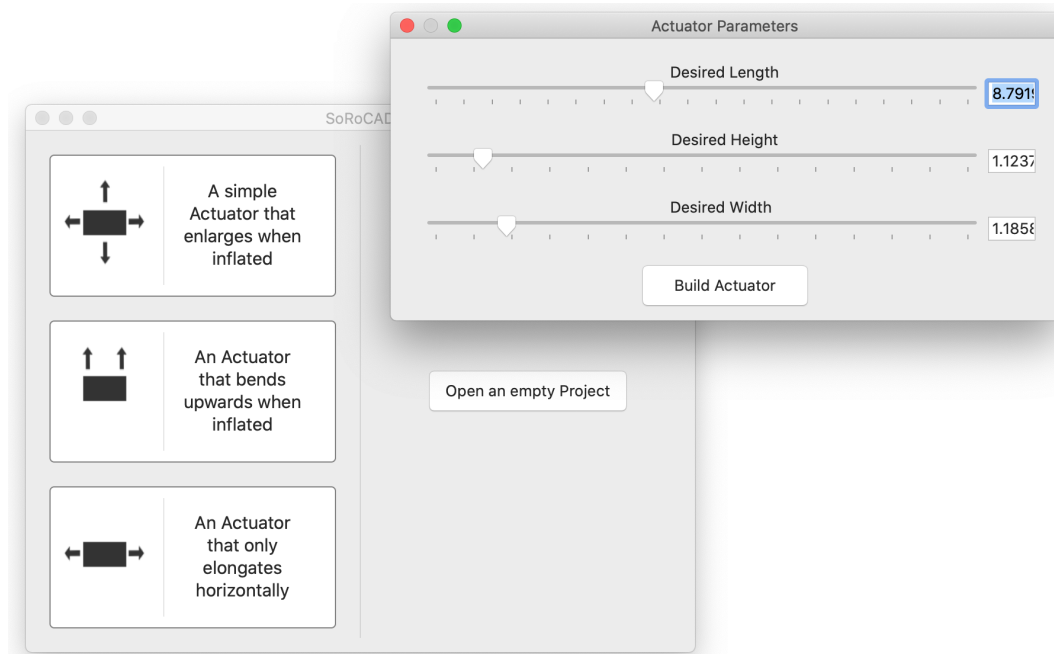


Figure 3.1: The template selection window opens up when starting SoRoCAD. When a template was selected, the user is asked to define the actuator’s dimensions.

cover the enlargement motion, where the actuator enlarges equally in all directions, as well as the bending, and the horizontal elongation. If one of the templates is selected, the user is asked to enter values for the parameters, such as the desired length, height, and width of the final actuator (see Fig. 3.1).

In the *air channels* tab, where the inner air channels can be designed, air channel blocks of different types can be created and assembled from the cell library on the upper right corner, and custom sizes and orientations can be applied to each selected item through sliders, text fields or by dragging the viewport interaction axes that can be enabled via the respective buttons on the upper left corner of the view (see Fig. 3.2). The dimension box on the top continuously displays the current dimensions of the model.

The *air channels* tab

The second main part of the workflow takes place in the *silicone exterior* tab, where the silicone exterior of the created air channel is designed or, more precisely, the wall thick-

The *silicone exterior* tab

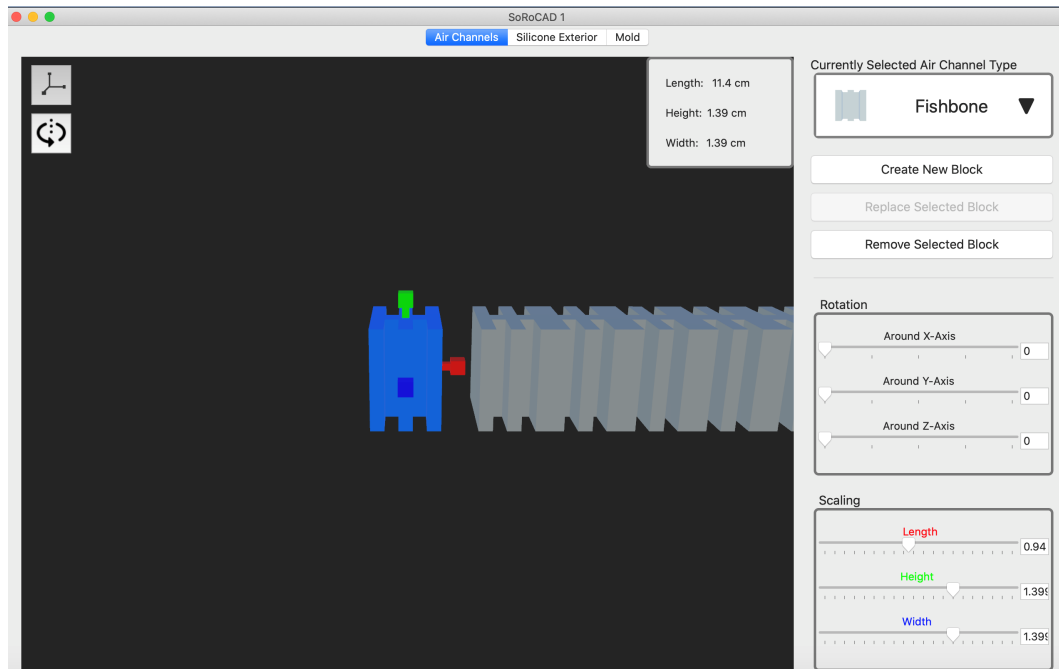


Figure 3.2: In the *air channels* tab the inner structure of the actuator can be designed. The scaling axes are enabled to rescale the selected block through dragging.

The tutorial view

GIFs indicating
current motion
tendencies

Simulation of the
actuated soft robot

nesses per air channel block for each of its sides are defined. This is, again, achieved by using sliders, text fields or the viewport interaction axes for scaling (see Fig. 3.3). When changing to this tab for the first time, a tutorial view opens up, providing cues on how to manipulate the actuator by changing the wall thicknesses to achieve each of the three motions that are already known from the template selection, i.e. the bending, elongation or enlargement motions (see Fig. 3.4). For additional help, SoRoCAD provides mini real-time simulations on the lower right corner, whenever one of the wall thicknesses is changed. Although these simulations are not supposed to accurately demonstrate the exact behaviour, they can be helpful to indicate the tendencies of the deformations caused by the last change made. Also within this tab, the user can run a simulation of the current soft robot when actuated with user-defined parameter values for the Shore Hardness of the silicone and the applied air pressure, both of which can be defined within the simulation parameters window that opens up. When the values are entered, the simulation starts and the user can

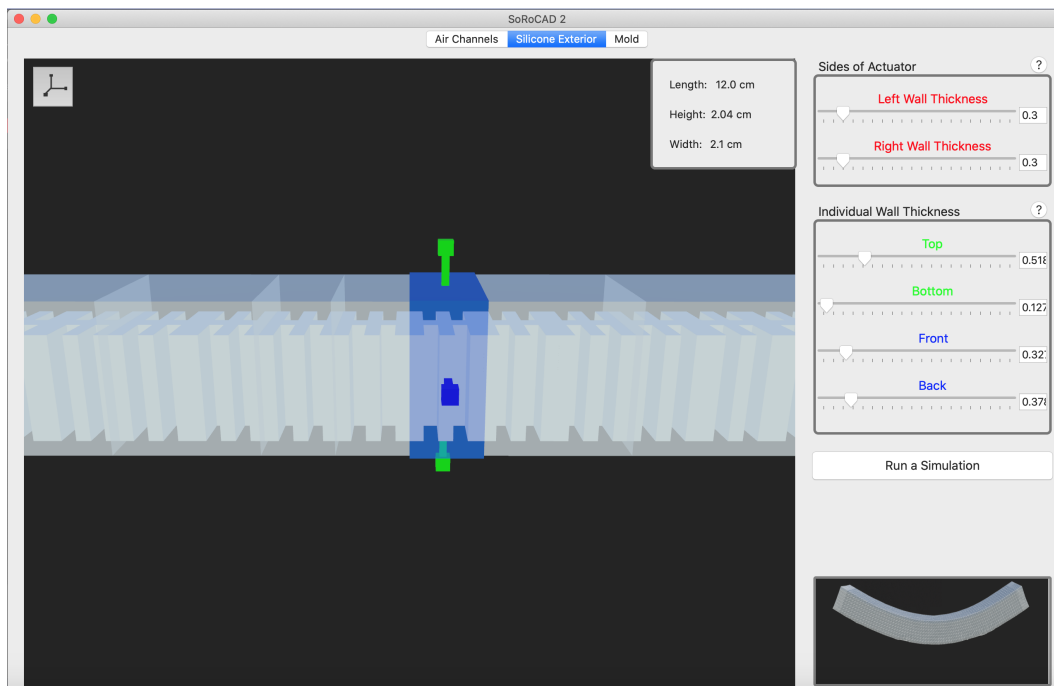


Figure 3.3: In the *silicone exterior* tab the wall thicknesses of the actuator can be designed. The shown actuator bends upwards when actuated. To make changes to the whole actuator, the blocks need to be selected and modified one after the other.

slide through the time steps, given in seconds, to see the inflated actuator after each amount of time (see Fig. 3.5).

Finally, the *mold* tab does not offer as many user interaction elements as the previously described tabs. Here, the auto-generated moulds of the current actuator can easily be exported as .stl, .obj, or .dae files to be 3D printed and cast (see Fig. 3.6). Moreover, the mould's wall thicknesses, as well as the layer split height, can be changed if necessary.

The *mold* tab

3.3 Identifying Problems of the Current State

Striving for providing an easy solution for the novices to get to know the basic principles of Soft Robotics with no prior knowledge required, SoRoCAD incorporates many

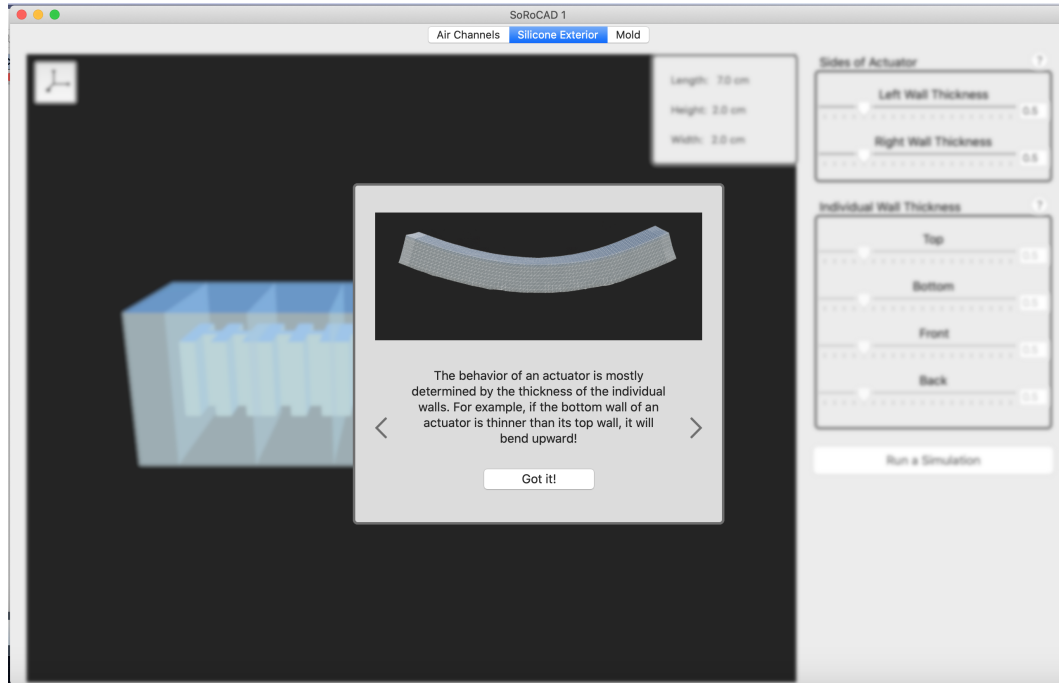


Figure 3.4: The tutorial view opens up when changing to the *silicone exterior* tab for the first time and gives hints on how to reach a certain behaviour, such as a bending motion.

helpful features, such as offering ready to use templates, tips, and simulations. Still, the tool shows problems in the field of user interaction and deployment. Additionally, SoRoCAD, especially the simulation and mould generation, needs to be technically evaluated to analyse its validity and practicability.

3.3.1 User Interface & User Interaction Problems

To evaluate the user interface and the workflow of the SoRoCAD tool, Strüver [2020] conducted a qualitative user study, in which the users were, amongst other tasks, asked to recreate the given actuators with their actuated behaviour, using the tool only. After having analysed the study results, the predominant issues will be discussed hereafter.

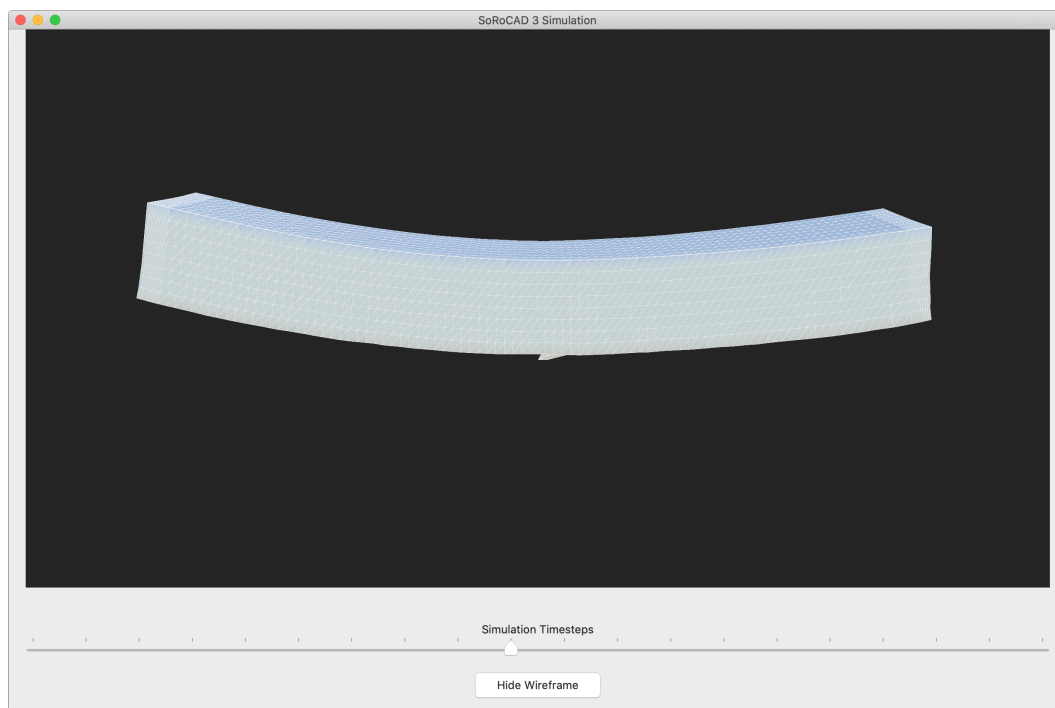


Figure 3.5: The simulation shows that the designed actuator bends slightly when actuated.

The problems that occurred the most were related to the tool's user interaction. For instance, the users were mostly confused by the orientation when dealing with the model in the *air channels* and *silicone exterior* tabs, as there were no indicators to help to orientate themselves. The user would often rotate the camera, sometimes by accident, and afterwards experience difficulties distinguishing the upper and lower side of the model. As there was no option to reset the camera, this problem was not easy to solve and usually cost the user a lot of time. This happened especially because the default position of the string of air chambers was not centred, or because the camera zoomed in on the actuator too much and would therefore hide half of the blocks, which forced the user to change the camera perspective to apply changes to the whole model (see Fig. 3.2 and Fig. 3.3).

Orientation problems

Another interaction that caused many problems was related to selecting and deselecting blocks. While for some users it was unclear that the blocks had to be selected before

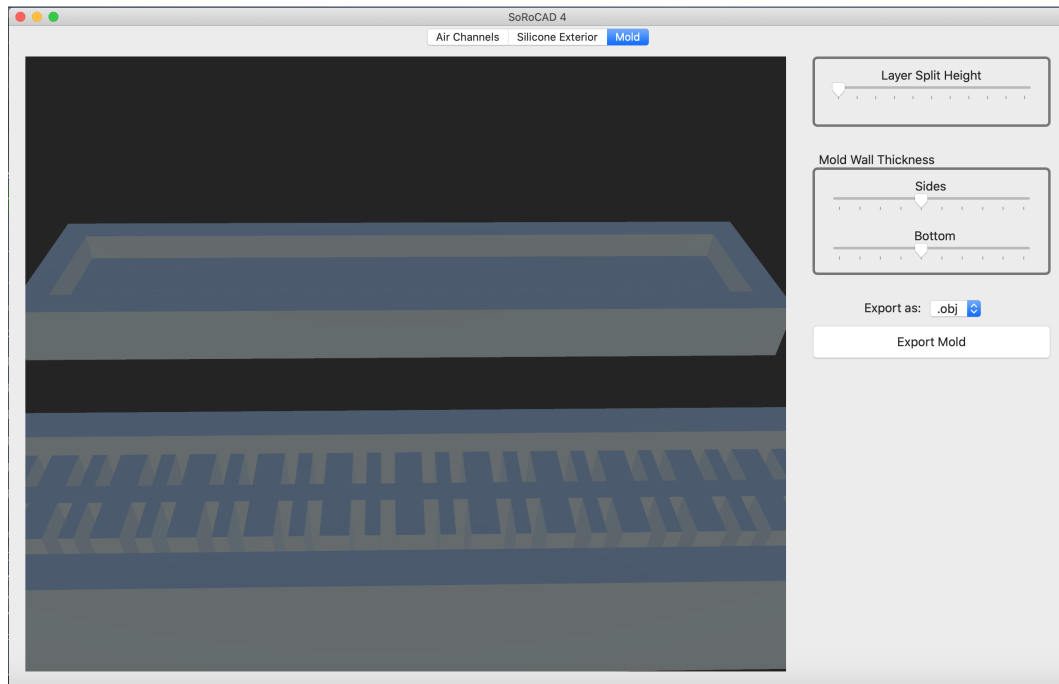


Figure 3.6: The automatically generated ready to use moulds of the freshly designed actuator can be exported in the *mold* tab.

Input problems

changes could be applied to them, the majority was struggling with the fact that only one block could be selected at a time, which made it a tedious task to make changes to them all. This was even aggravated by the fact that most users faced problems handling the input options since they felt that the sliders were imprecise and the buttons that enabled the viewport interactions were hardly recognised as such and therefore rarely touched. Additionally, the text fields contained too many floating point digits, many of them invisible because the fields were too small (see Fig. 3.3), and some users discovered an unexpected behaviour when entering comma-separated decimal numbers like they were used to from the German notation. Furthermore, the lack of shortcuts and not having the possibility to undo changes was notably restricting the speed of the workflow.

Misinterpretation of user interaction

Besides having problems identifying the viewport interaction buttons or misinterpreting them and, as a consequence, not using the respective axes properly, other issues occurred that fall into the category of interpretation prob-

lems. Most notably, users were missing units throughout the tool, for example for the dimensions at the template parameter window in the beginning, for the Shore and the pressure value when starting the simulation and for the time steps in the simulation window.

Also, in many cases, the template selection window caused confusion because either the buttons for selecting the respective templates were not identified, or the presented actuation behaviours from the templates were misinterpreted to be different to the motions the users had been looking for. Thus, the offered templates were only rarely used. Some more parts of the software confused the users due to a lack of explanations, too, such as the meaning of the Shore value they were asked to set before the simulation could start.

Some other parts, on the other hand, were unclear because too much information was given. The tutorial (see Fig. 3.4) which, as an example, opens up when changing to the *silicone exterior* tab and aims at giving hints for designing actuators with a specific deformation behaviour, was mostly ignored and immediately clicked away before reading. Some other participants, however, read the hints but they could not see their relevance to the tasks, as a result of being offered the information at an unexpected moment and therefore misinterpreting them.

Regarding the complexity of the realisable actuators, the tool is rather limited. Although this is, to some extent, intentional to constrain the number of potential sources of confusion, having the option to create slightly more complex designs would be helpful. Also, in the current state, it is not possible to select the position of newly created air channel blocks, as they are always appended at the outer right of the existing air channel string.

Limited complexity of the actuators

Furthermore, the SoRoCAD tool contains various bugs. The more severe ones include an erroneous selection and deselection behaviour, unsynchronised sliders and text fields, partially incorrect mini simulations and a wrong passing on of user-defined parameter values when selecting the elongation template. Besides that, the tool crashes whenever the user directly changes from the *air channels* to

Bugs and UI issues

the *mold* tab and all applied modifications to the silicone exterior are reverted when changing back to the *air channels* tab from anywhere else. Additionally, the split height of the generated moulds in the *mold* tab causes slight fragments on the top layer which cannot be corrected by the user, as the range of the respective slider is not wide enough. Moreover, regarding the design of the tool, some major issues stand out. Examples include missing alignments of the UI elements, inconsistent font sizes and flashy colours, as well as distorted and pixelated images and icons that were partially misleading, such as the icon of the bending template (see Fig. 3.1).

3.3.2 Distribution Issues

SOFA issues

Apart from user interaction-relevant issues, one present and particularly severe problem has to do with distributing the SoRoCAD tool. While at the moment, the tool only runs properly on the one device it was developed on, the next step would be to provide the possibility for any macOS user to easily install and use it together with all included features. More precisely, the problem lies in the SOFA framework that was integrated into SoRoCAD to enable the simulation feature. As there is no pre-compiled SOFA binary available that contains all the plugins required for the simulation, the framework needs to be manually compiled by the user, which is not an easy task. More details on the compilation of SOFA will be discussed later in this chapter, in section 3.5 'Distribution of SoRoCAD'. What is missing is a solution for automating the compilation to be executed when SoRoCAD gets installed, so the user will not have to worry about any additional challenges.

Technical evaluation

Moreover, before distributing the tool, it is essential to assess the accuracy of the simulations, as well as the realisability of fabricating the same actuator as previously designed with the provided moulds. Thus, while Strüver [2020] already conducted a user study to evaluate the tool's ease of use, SoRoCAD still needs to be technically evaluated to analyse its validity and practicability. This will be discussed later in chapter 4 'Technical Evaluation'.

3.4 Extending SoRoCAD

One focus of this thesis was to extend the SoRoCAD tool to solve present problems, in particular regarding the earlier mentioned results of the user study conducted by Strüver [2020], as well as other issues that we noted throughout working with the tool.

3.4.1 User Interface & User Interaction Improvements

While earlier in this chapter, the most common problems of the users during the study were presented, we will now take a look at the approaches we made to solve them. One of the major issues that were stated by all participants was that the given input methods for changing the actuator's design, such as the wall thicknesses, were frustrating and somehow awkward. This held for the sliders, as well as the text fields and the viewport interactions. To resolve this, we discretised the sliders to snap on tick marks which speeds up the action when the user has no precise number in mind and only wants to significantly increase or decrease the value. In case more specific values are required, the user can also use the text fields to enter them. Those, we adjusted to display the values after being rounded to the second or first decimals, since their differences to the more precise values are, in this dimension, negligibly small. At the same time, we thereby reduce the overpowering of too long and partially invisible numbers as can be seen in Fig. 3.3. Additionally, we added a function to equivalently accept comma-separated and dot-separated decimal numbers as input for the text fields, because this misconception is expected in a German tool. Moreover, as a consequence of the participants frequently mixing up the mapping of the sliders and the directions, we renamed and permanently coloured the labels of the sliders in accordance with the standard direction colours that are visible at several points in the view and we further changed their orders so that the 'Bottom' slider is positioned at the bottom (see Fig. 3.7 and Fig. 3.8).

Resolving input
issues

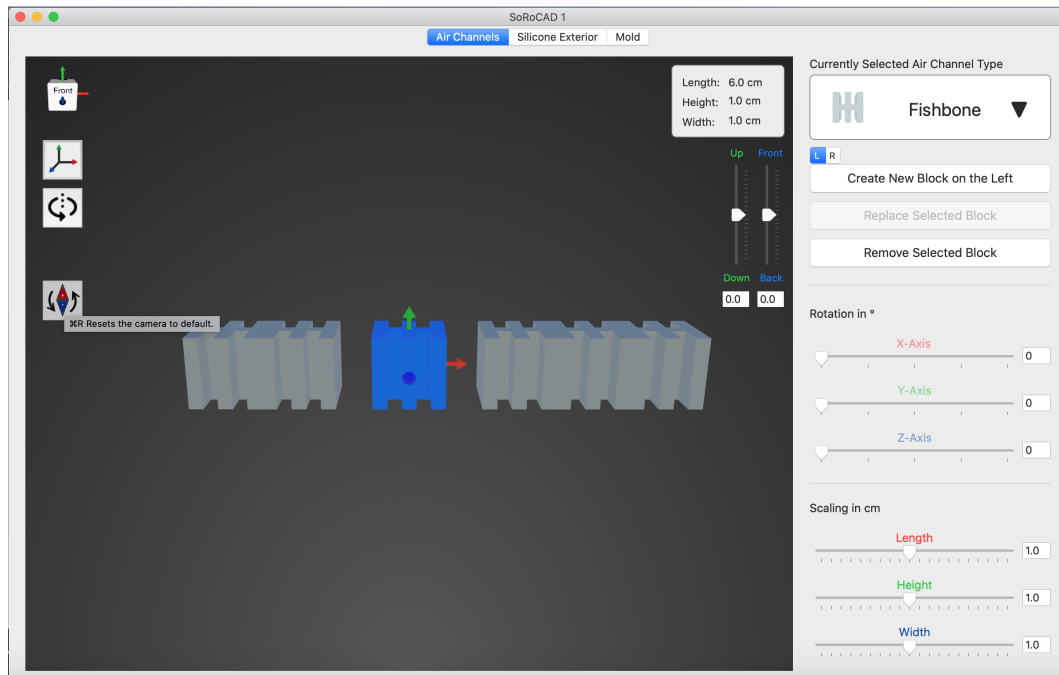


Figure 3.7: In the redesigned *air channels* tab the model is always centred on a gradient background. The tooltips help to identify the buttons, such as the button to reset the camera when the user loses orientation.

Resolving selection
issues

Multiple selection

Although disabling the sliders and text fields, to some participants, it was unclear that blocks had to be selected before applying changes. We, therefore, lowered the opacity of the label colours when no block was selected and, additionally, preselect one block when changing to the *silicone exterior* tab to indicate this principle. However, this was not the only issue in the context of block selection and deselection. At some points, the implementation of its behaviour immensely slowed down the workflow, most notably when wanting to apply changes to many sides of many blocks. In that case, one after the other, the block needed to be clicked on, then the modifications could be made for that specific element only, after that, the next one needed to be selected, and so on. To avoid this laborious work, we enabled the function to select multiple blocks and make changes to them all at once by simply holding the command or shift key and simultaneously selecting the respective blocks. To even extend this functionality and increase the speed, we also added the Command-A shortcut to se-

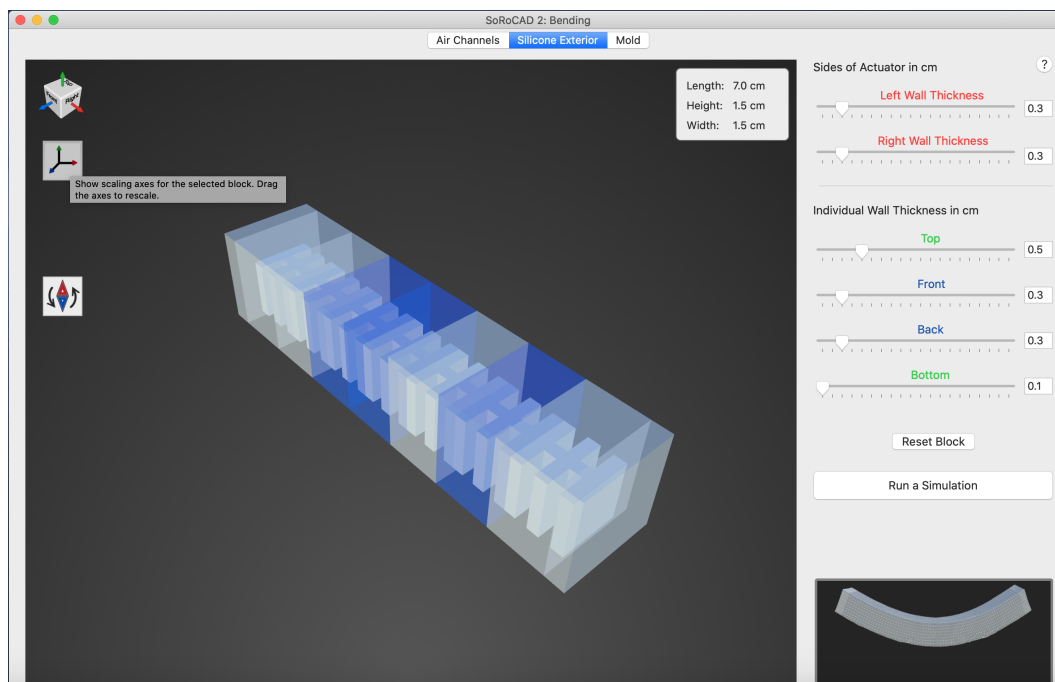


Figure 3.8: The redesigned *silicone exterior* tab allows to select and modify multiple blocks at once. The orientation indicator cube on the upper left corner moves simultaneously when the camera is changed.

lect all available blocks at once. Besides that, other blocks can be selected through the left and right arrow keys and, when being removed, the previous block gets automatically selected to reduce the number of interactions for the sake of time. More approaches we made to speed up the process include a variety of other common key equivalents, amongst others the Return key to confirm the input, the Command-C, Command-V, and Command-Z shortcuts to copy, paste, and undo, as well as the Delete key to remove blocks and the space bar to create them.

Apart from time-consuming processes when using the tool, another factor that undoubtedly held up the workflow was the misinterpretation of user interaction elements. Most importantly, as mentioned earlier, all participants were confused by the absence of units at relevant places, for instance when they were asked to enter custom dimensions of the actuator to generate a soft robot with the selected template (see Fig. 3.1), or when choosing the air pressure that is ap-

Shortcuts

Resolving
misinterpretations

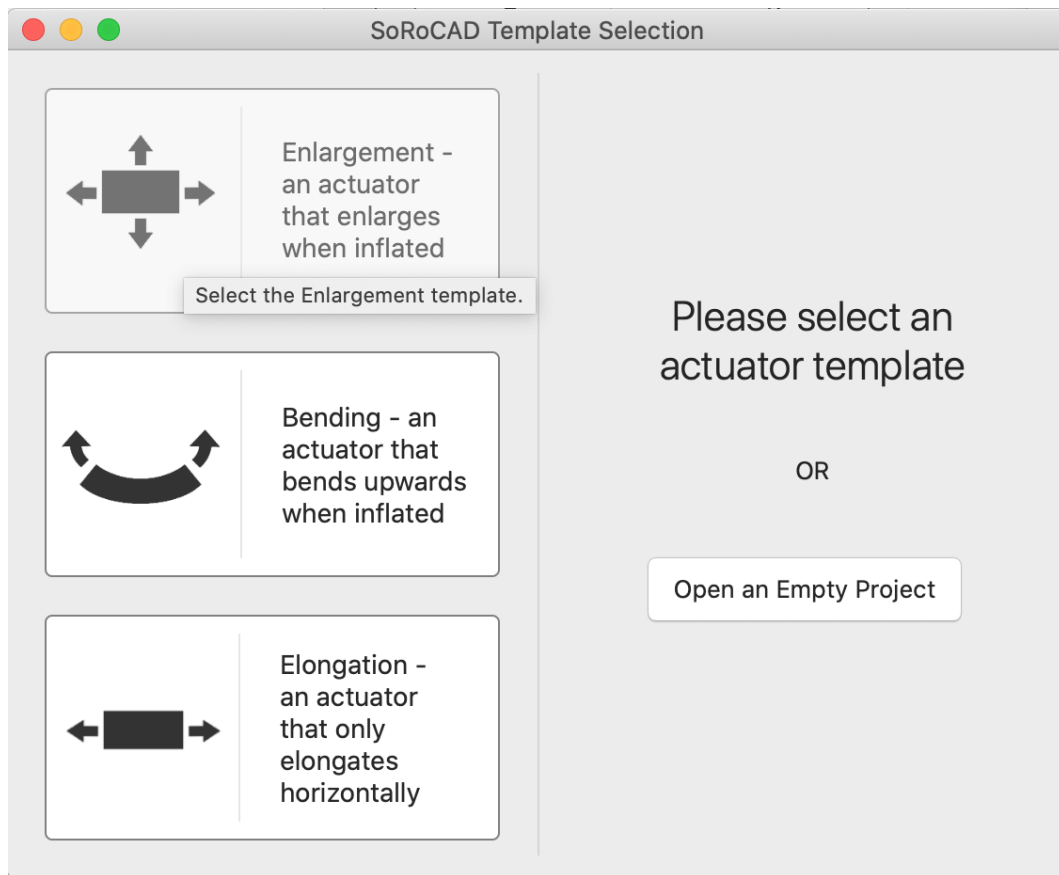


Figure 3.9: The template selection window was extended by meaningful instructions, window titles, tooltips and redesigned icons.

plied to actuate the robot. This was easily fixed by adding the missing units (see Fig. 3.8).

Template selection icons

Another major user interaction issue occurred in the first window where the user can select a template or alternatively open an empty project. Several participants misunderstood the behaviours that were indicated by the template icons or did not even realise that they had a choice since they did not interpret the icons as clickable buttons and the window was lacking any instruction (see Fig. 3.1). Thus, we slightly redesigned the window (see Fig. 3.9), by first adding a noticeable instruction to either select a template or to open an empty project, and by changing the window title to 'SoRoCAD Template Selection'. The next hint is given by highlighting the template buttons and display-

ing informational expansion tooltips when hovering over them with the mouse, so they can be identified as clickable items. To further solve recurrent problems of comprehension, we redesigned the icons to be more straightforward and, besides that, look less distorted and pixelated and, additionally, we reformulated the explanations, also expanding them by the respective template title.

We then continued the idea of extending the tool by adding instructive window titles, reformulating expressions, or enabling hover effects, such as highlights and tooltips, at many places, most notably to emphasise the function of the viewport interaction, which posed a huge problem to most participants. Here, again, many users did not recognise the viewport interaction buttons, some did not even notice that they were there. Hence, we once more enabled hover effects with explanatory tooltips and redesigned the icons to be more eye-catching by integrating the same colours that are associated with each of the three coordinate axes (see Fig. 3.2). More problems of comprehension were engaged by adding help signs which display a short explanation when hovered over, for example for the Shore value in the simulation parameters window.

Viewport interaction
icons

The last major problem, caused by misunderstanding certain user interaction elements, is related to the tutorial window that opens up when changing to the *silicone exterior* tab for the first time. Although it is typical behaviour to ignore and immediately close any help window that opens up without being asked for it, we attempted to improve this by first changing the order of the three available tips to always start with the most relevant, according to the template the user has selected. For example, if the elongation template was chosen, then the tutorial view would present the hints for the elongation behaviour in the first place. In addition to that, we once again reformulated the explanation of the three deformations and added template titles to each tip to make its purposes more obvious (see Fig. 3.10). Lastly, we shortened the GIFs, which are shown to describe each motion, to get to the heart of their behaviours.

The tutorials

To further enhance the ease of use of SoRoCAD, we expanded the tool by permanently displaying relevant infor-

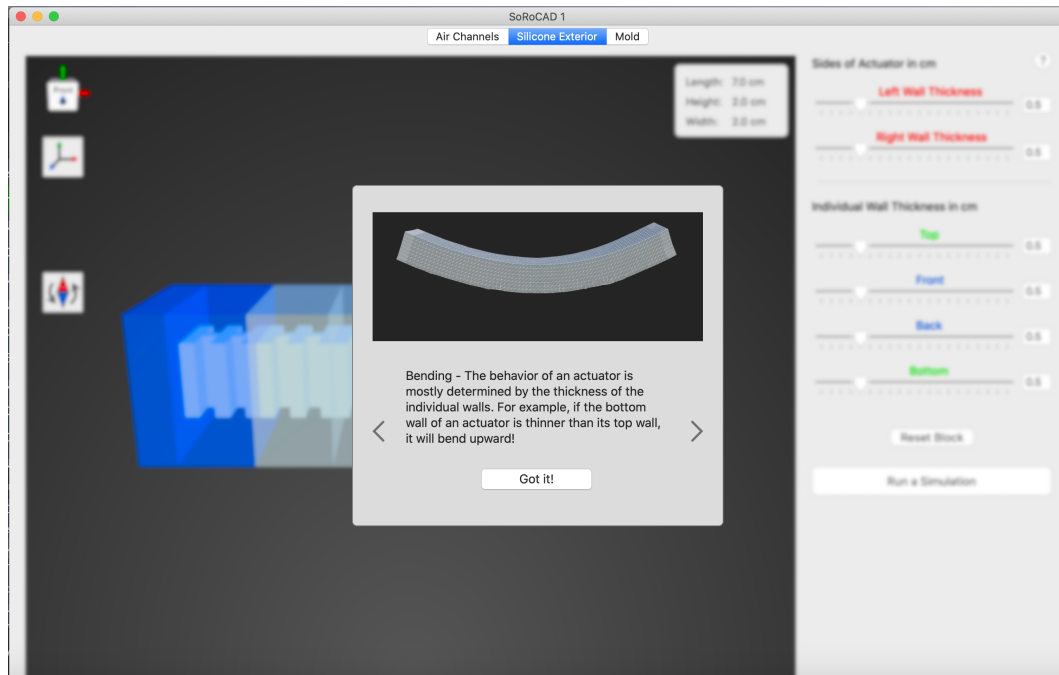


Figure 3.10: The redesigned tutorial view was reformulated and contains titles for each motion.

mation, such as extending the window titles by the selected template name. To give another example, we added an information box to the upper right corner of the simulation window, containing the dimensions of the actuator at each time step, so that even slight effects of the enlargement become apparent and can be precisely determined, as well as the selected Shore and air pressure value, to allow a direct comparison of the impacts of the different parameters (see Fig. 3.11).

Resolving bugs and UI issues

Besides the changes that were directly derived from the user studies, we resolved all the bugs we came across and which were mentioned earlier in this chapter, mostly related to the selecting and deselecting behaviour of blocks and an incorrect handing over of the user input. Another troublesome problem we solved was that all wall thicknesses of the silicone exterior of the designed actuator were set back to default, clearing all modifications the user worked on, whenever the user changed back to the *air channels* tab. Similarly irritating, when immediately chang-

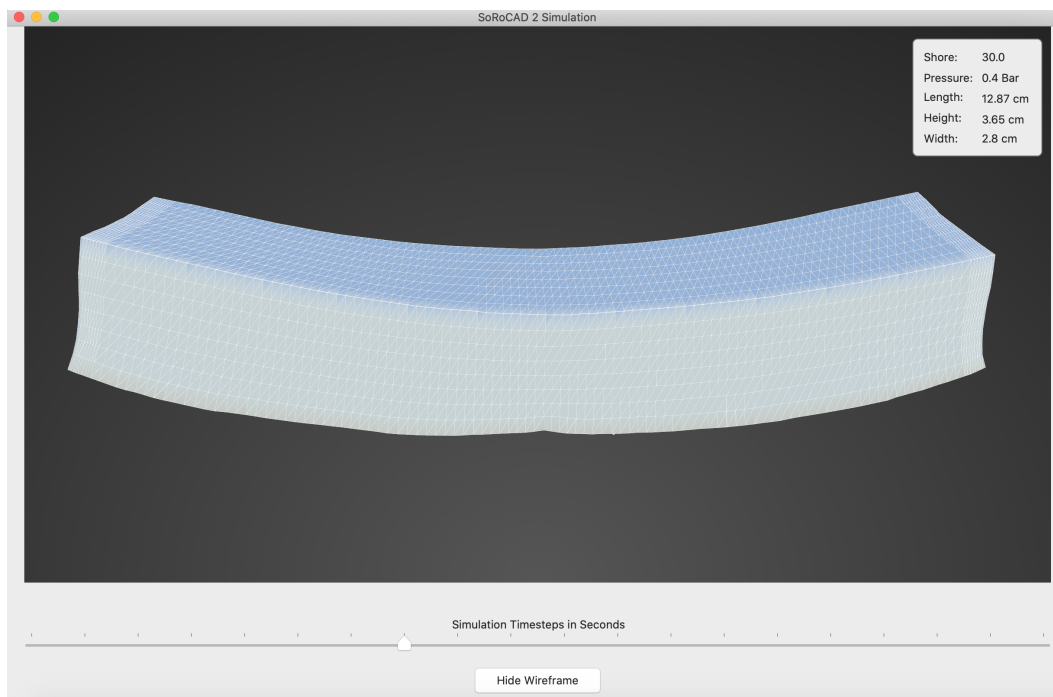


Figure 3.11: The redesigned simulation window contains a box on the upper right corner that displays the actuator’s dimensions at the current actuated state, as well as the user-defined Shore value and air pressure.

ing from the *air channels* to the *mold* tab, the tool would crash, again erasing all modifications, which we now avoid by opening an alert to prevent the user from changing. In addition to that, we implemented a solution to the problem in the *mold* tab, where the generated top layer of the mould would exhibit slight fragments, which, with a closer look, can be seen in Fig. 3.6. We expanded the layer split height slider to cover a wider range of possible values, which allows the user to manually correct possible fragments (see Fig. 3.12). Furthermore, we made several adjustments to create a more pleasing and neat user interface, which we will now briefly sum up: after we fixed missing alignments, spelling mistakes and inconsistent font and button sizes, we adapted the viewport interaction axes in their shape, modified the orientation of most air channel types in the list to look more meaningful, and changed all occurrences of too flashy colours to smoother ones. We, additionally, replaced the dark and dull background with a gradient background, to give it a more dynamic look, and significantly

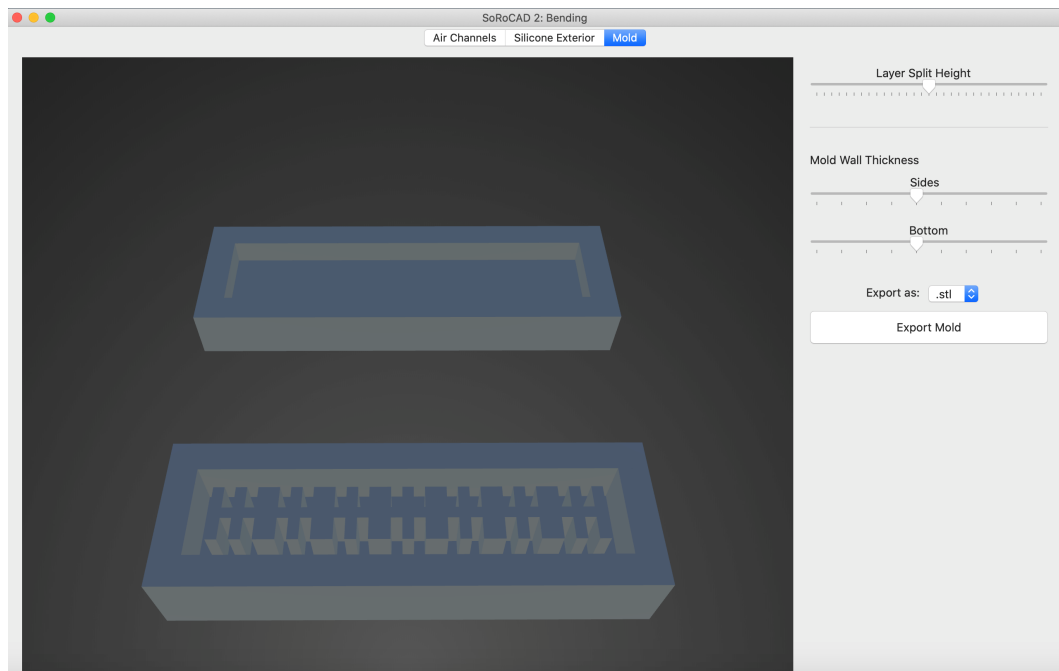


Figure 3.12: The *mold* tab was modified to allow a wider range of the value for the layer split height to correct fragments.

reduced the size of the very present dimension box in the upper right corner of the *air channels* and *silicone exterior* view. Finally, we replaced the bold grey boxes enclosing the control elements, on the right of the three tabs, with subtle horizontal lines and repositioned some elements to clean up the overall appearance.

3.4.2 Additional User Interaction

Besides the modifications we made to fit the users' needs, we extended SoRoCAD with additional user interaction functionalities to solve the remaining issues. Since one of the major problems of all participants in Strüver [2020]'s study - the orientation problem - could not be solved by adapting already existing interactions, we first added a button to reset the camera to the viewport interactions of both the *air channels* and the *silicone exterior* tab (see Fig. 3.7). This allows the user to easily return to the initial camera position whenever he loses his bearings. To, in the first

place, avoid that users play around with the camera too much and reach the point where they can no longer distinguish the bottom and the top of the actuator, we added a function to calculate the centre of the view whenever a new block is added in the *air channels* tab. All existing blocks are then repositioned accordingly to be always centred, instead of statically appending the air channels on the right side, beyond the borders of the view frame.

However, the most significant expansion to reduce the disorientation when using the tool was the addition of the orientation indicator cube on the upper left corner of the *air channels* and *silicone exterior* tabs (see Fig. 3.8). The cube, which is labelled on each of its sides with the side's name and which involves the three coordinate axes in the repetitive colours red, green, and blue, moves synchronously with the designed air channels or the actuator, respectively. When moving the camera, the user is therefore constantly given an overview of the momentary orientation to support the interaction with the model from all sides.

Orientation Indicator

The next functionality we added was the button to reset the wall thicknesses of the selected blocks in the *silicone exterior* tab. The user is, thus, free to experiment with the sliders and the other user interaction elements to get to know the tool, for example after having selected a template, without having to worry about making irreparable changes to the preset wall thicknesses. Moreover, this button speeds up the workflow when the user wants to create many different actuators that share the same inner structure.

Reset wall
thicknesses

Another interaction we included in the tool to save time if the user changes his mind on the already designed air channels, was the option to decide where newly created blocks will be positioned. Other than before, the new blocks will no longer be automatically positioned at the outer right position of the air channel. Instead, the user can interfere with this behaviour by selecting blocks to determine the position of the new component. In addition to that, we added a segmented control to specify whether new blocks will be positioned on the left or the right of the selected cell or, in case no cell was selected, of the actuator (see Fig. 3.7).

Repositioning of
blocks

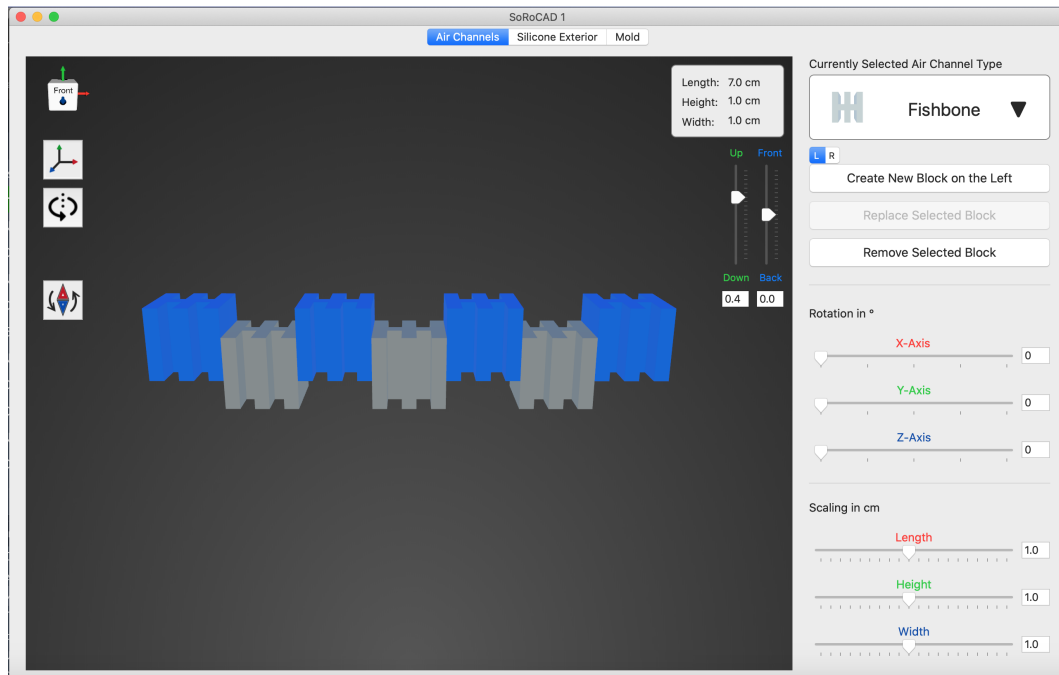


Figure 3.13: The new sliders on the right side of the *air channels* view allow moving the selected blocks up, down, back, and front. Also, the user can now decide where new blocks will be positioned.

Lastly, to enable more customisable and complex designs, we decided to enable the option to modify the horizontal and vertical position of the air channels blocks. Hence, we included two sliders in the *air channels* view to adapt the front/back, as well as the up/down positions of the selected items. A possible remodelling can be seen in Fig. 3.13.

3.5 Distribution of SoRoCAD

The next goal of this thesis was to make SoRoCAD ready for distribution. We, therefore, had to properly integrate the SOFA framework into the tool, which, at the moment caused severe issues and prevented the tool from being deployed. In the following, we will first elucidate the deployment issues of SoRoCAD in more detail, and we will briefly present our solution to the problem afterwards.

3.5.1 Distribution Issues

The SoRoCAD tool, which is the subject of this work, was completely written in Swift and is currently only a Mac application. At the moment, however, the only device the tool would properly run on is the device it was implemented on. While on any other Mac it can be installed and used to some extent, as soon as the user starts the simulation, the tool would crash. This issue is caused by the SOFA framework which is used by SoRoCAD through the command line to handle the simulation.

The [SOFA framework](#)¹ is an open-source physics-based simulation platform. Although it is primarily used for medical simulations, there are plenty of plugins that support the use for other purposes, in our case for the simulation of soft robots. To make the simulation of the actuators work in our tool, we used three plugins: first, we used the [Soft Robots](#)² plugin to simulate and control soft robots. Besides that, we integrated the [SofaPython](#)³ plugin to define the scenes including the actuator and to set the values for the needed parameters, such as the gravity, as well as the dimensions. For the computation of the simulation, however, SOFA requires surface meshes, as well as volumetric meshes of the actuator. Since in SoRoCAD, we only work with surface meshes, we still have to transform them into volumetric meshes to hand them both over to the framework, which is not a trivial task. For this computation, we finally appended the framework by the [CGAL](#)⁴ plugin, whose underlying C++ written library can be used for geometric computations of many kinds, including the one we need.

The SOFA
framework

Our first idea, to integrate a pre-compiled version of the SOFA framework, was defeated early in the process, as we could not find a pre-compiled version with all three plugins we needed, only one without the CGAL plugin. We then continued searching for an alternative plugin with similar

¹<https://www.sofa-framework.org/> Accessed: April 8, 2021

²<https://project.inria.fr/softrobot/> Accessed: April 8, 2021

³<https://www.sofa-framework.org/community/doc/plugins/usual-plugins/python-scripting/> Accessed: April 8, 2021

⁴<https://www.sofa-framework.org/applications/marketplace/mesh-generation-with-cgal/> Accessed: April 8, 2021

abilities to CGAL, but with a pre-compiled version, as a replacement but did not make a find, either. The last option was to separately build SOFA, including the plugins, before starting SoRoCAD. As the compilation of SOFA still depended on absolute file paths, it could not be made portable between different devices, hence each user would have to build SOFA himself when installing SoRoCAD. However, the process is not straightforward and takes time and effort to complete successfully, wherefore we did not expect the potential SoRoCAD users to autonomously undergo the procedure. Thus, we finally decided to automate the building of SOFA and its plugins through a script to facilitate the user's work and reduce his responsibilities.

3.5.2 The Run Script Build Phase Solution

Xcode, where SoRoCAD was developed, incorporates the possibility to run custom code within the build process. This run script build phase works through the given command-line script and executes each instruction one after the other. To avoid the cumbersome process for each user to manually build SOFA and its plugins before use, we integrated such a run script build phase into SoRoCAD that handles all necessary tasks to successfully install the framework and the respective plugins.

The script that compiles SOFA

Our shell script, which we will briefly introduce in the following, was composed of the step-by-step instructions from the online [SOFA documentation](https://www.sofa-framework.org/community/doc/getting-started/build/macos/)⁵, as well as the documentation of the [Soft Robots](https://project.inria.fr/softrobot/install-get-started-2/download/)⁶, [SofaPython](https://github.com/sofa-framework/doc/)⁷ and [CGAL](https://www.sofa-framework.org/community/doc/plugins/usual-plugins/cgal-library/)⁸ plugins. In the beginning, the script checks for the required preconditions, most importantly, whether a fully operational version of SOFA and all needed plugins was already installed. In that case, the script immediately terminates and the remaining build phases are continued. Otherwise,

⁵<https://www.sofa-framework.org/community/doc/getting-started/build/macos/> Accessed: April 8, 2021

⁶<https://project.inria.fr/softrobot/install-get-started-2/download/> Accessed: April 8, 2021

⁷<https://github.com/sofa-framework/doc/> Accessed: April 8, 2021

⁸<https://www.sofa-framework.org/community/doc/plugins/usual-plugins/cgal-library/> Accessed: April 8, 2021

the script continues testing for prerequisites, which we will discuss later on, and installs missing libraries. After all necessary sources are downloaded and the makefiles are generated, the compilation of SOFA starts. Eventually, all three plugins are compiled analogously. As soon as the build was completed, the script terminates without errors and SoRoCAD is ready to use, including its simulation feature.

While our initial draft of the script did not contain any special prerequisite's checks, we performed various tests on different machines and one by one appended our instructions with additional checks for the occurring errors along with their respective solutions. For example, the findings of the first two tests were related to improperly installed Command Line Tools, which was solved by reinstalling them. After the extensions were done, we tested the script further on and achieved solely successful results, i.e. the SoRoCAD tool was built without errors and, finally, all of its features could be used on another user's Mac.

Reworking the script

3.6 Summary

In this chapter, the SoRoCAD tool was introduced in more detail, explaining its motivation and intentions and briefly going through the standard workflow, as well as analysing its limitations at the time of the start of this thesis. Then, our approaches to extend the tool and rework the user interface to solve these and other problems that hindered the user's work, were presented. The main problems included the misinterpretations of user interactions and other ease of use limitations, that considerably slowed down the workflow, such as input methods or orientation issues. Moreover, the tool was limited in the complexity of the designable actuators, wherefore we added the possibility to move the air channel blocks along the vertical and horizontal axes. Finally, the at the time existing distribution issues that prevented using SoRoCAD on other devices were discussed and it was described how our approach to including a run script build phase resolved those.

Chapter 4

Technical Evaluation

In this chapter, the technical evaluation we conducted for the next main focus of this thesis, will be presented. The goal of this technical evaluation was to assess the tool, regarding accuracy and feasibility. First, the goals and the setup will be introduced, as well as the procedure of the evaluation. Then, our results will be pointed out and interpreted and, finally, possible approaches for improving them will be discussed.

4.1 Setup

To technically evaluate the SoRoCAD tool, i.e. testing the accuracy of SoRoCAD's predictions and the practicability of fabricating soft robots from the suggested moulds, we recreated a setup to actuate self-made, SoRoCAD-based soft robots. We then compared the real-life measured values of their deformations to the predicted measurements and investigated the discrepancy between them.

The independent variables of our experimental setup include the applied air pressure, which we varied within the same range as available in SoRoCAD, 0-2 bar, and the design and material's hardness of the soft robot. On the other hand, the actuated behaviour and the maximum ca-

The dependent and independent variables

capacity of applied air pressure before exploding represent the independent variables. For this technical evaluation, we analysed mostly template-based actuator designs and their moulds, both of which were generated by SoRoCAD. The expected behaviours for the designs we chose include simple bending, elongation, enlargement and bending into an S-curve. For each of the four designs, the overall dimensions amount to 10x1.5x1.5 cm for the actuators whose inner structures are composed of fishbone-shaped air channel types, and 14x2x2 cm for the cubic air channel actuators. More information on the designs can be taken from Tab. 4.1, which lists the wall thicknesses for the fishbone-based soft robots. When designing the cube-based ones, we added 0.2 cm to each value in the table. After 3D printing

Wall Thicknesses in cm					Expected Behaviour
left/right	top	front	back	bottom	
0.3	0.5	0.3	0.3	0.1	Bending
0.1	0.4	0.4	0.4	0.4	Elongation
0.2	0.2	0.2	0.2	0.2	Enlargement
0.3	0.3	0.1/0.5	0.1/0.5	0.3	S-Curve

Table 4.1: The designs for the the fishbone-based actuators that were generated from the templates and used for evaluating SoRoCAD.

the SoRoCAD-generated moulds for the top and the bottom layers of each actuator, we cast silicone of different Shore Hardnesses and let it cure before glueing both parts together. To cover the whole scale of Shore A values, we examined silicones with 10, 20, 35, 50, 70, 85, and 95 Shore.

The experimental setup

For our setup to evaluate the behaviour of the actuated soft robots, we connected an airbrush compressor, a pneumatic regulator, a manometer to indicate the air pressure, and the prepared soft robots of the previously presented types and materials, with tubes, as shown in Fig. 4.1 . To measure the deformations, we placed the soft robots on millimetre paper and used a protractor, plus, we recorded the actuation with a camera that was mounted on a tripod. We then started the evaluation by activating the compressor and slowly intensified the air pressure that was applied to the actuator. Once within every 0.1 bar, we measured the deformation and finally compared our outcomes to the simulation results.

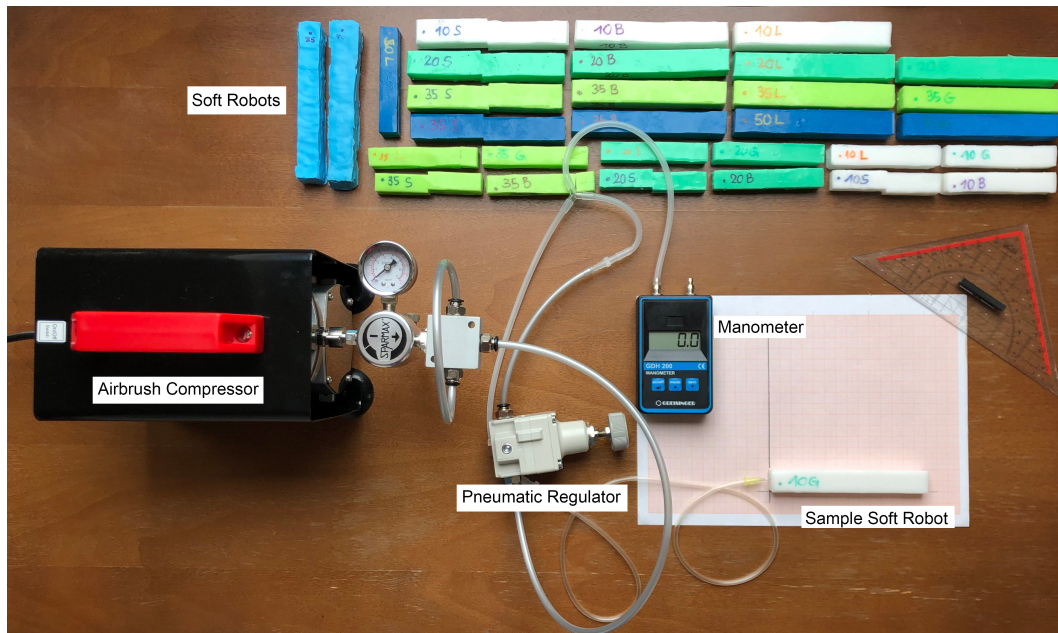


Figure 4.1: The experimental setup for the technical evaluation. The airbrush compressor, pneumatic regulator, manometer and soft robot are connected through pipes.

4.2 Results

Although we planned to measure the deformations of all designs with all available silicones, we ended up with significantly fewer functional soft robots that would deliver usable results. This was mainly due to the difficulties that arose after the curing of the silicone when we tried to demould the main bodies of the actuators. While the larger cube-based moulds only posed a problem with the 95 Shore silicone, demoulding 50 or higher Shore silicones from the smaller moulds was hardly possible without irreparably destroying them. For the fishbone-based actuators, we, therefore, decided to omit the evaluation of the soft robots with 50 or higher Shore, as well as for the 95 Shore cube-based ones.

Observations during the fabrication of the Soft Robots

Apart from these actuators, others raised a problem when actuated, as they did not show any deformation, because of small holes in the silicone exterior. This occurred for all actuators with 70 or higher Shore, as well as commonly for

the small, fishbone-based actuators. Another observation we made was that the small actuators would often let out the air early on since the air inlets would tear open, which, too, inhibited the measuring of the deformations of these samples. Additionally, the actuators that were composed of 50 Shore silicones only showed relatively small deformations.

Actuation results:
Enlargement

In the following, we will present our results for the remaining soft robots, which contain mostly 10, 20, and 35 Shore actuators, classified by their expected behaviours. Those soft robots that were constructed with the enlargement template did indeed show an enlargement behaviour which turned out stronger for the softer materials. Tab. 4.2 portrays the measurements for the elongation and the expansion of the soft robot's circumference for both the real and the simulated actuators when applying the maximal air pressure immediately before the real actuator's explosion. The simulations, however, did not forecast a uniform enlargement but the inflation into an eight-shaped state, which can be seen in Fig. 4.2 and sometimes predicted explosions for the same air pressure values for which their real counterparts remained intact.

Type	Reality			SoRoCAD	
	p_{max}	l_{max}/l_0	c_{max}/c_0	l_{max}/l_0	c_{max}/c_0
10 C	0.44 bar	171%	250%	114%	150%
10 F	0.42 bar	200%	200%	explosion	
20 C	0.71 bar	342%	450%	121%	147%
20 F	0.58 bar	111%	120%	explosion	
35 C	1.0 bar	114%	150%	121%	157%
35 F	faulty				
50 C	1.0 bar	100%	100%	112%	145%
50 F	faulty				

Table 4.2: Comparing the percentual elongation, given by l_{max}/l_0 , and the expansion of the circumference c_{max}/c_0 of the real and the simulated enlargement actuators at the time of applying the maximal air pressure p_{max} before exploding. l_{max} denotes the maximal length immediately before the explosion and l_0 the initial length before the inflation. The values for the circumference changes are given analogously by c_{max} and C_0 . The type is given by the Shore value and the cube (C) or fishbone (F) air channel type.

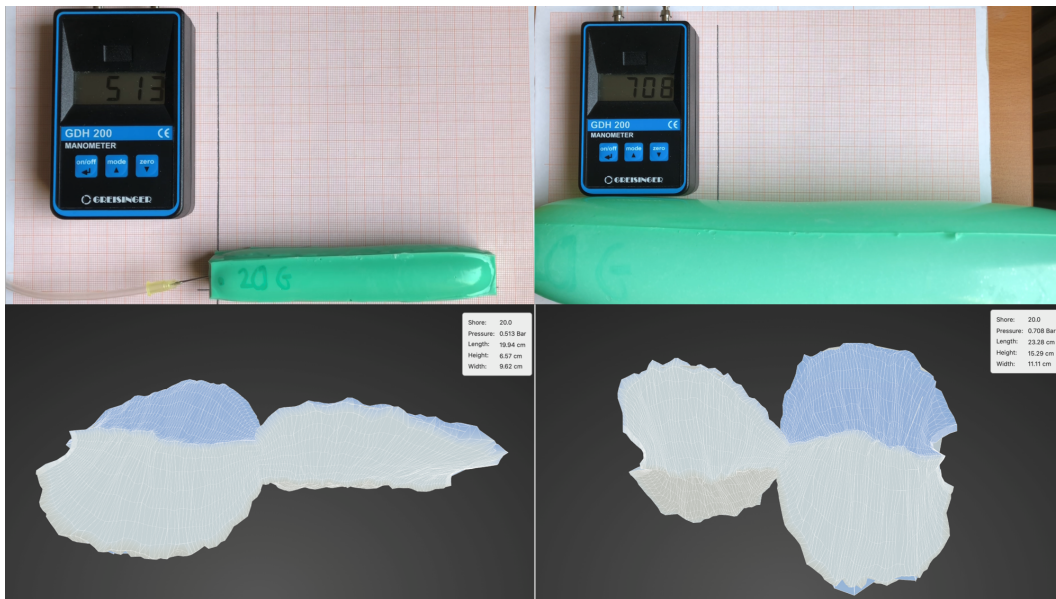


Figure 4.2: The deformations of the real (top) and the simulated (bottom) actuator in contrast. Enlargement template, 20 Shore, cubic air channel.

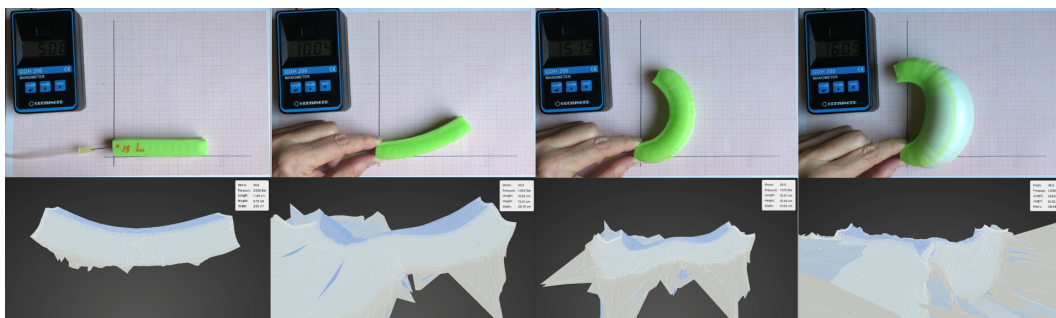


Figure 4.3: The deformations of the real (top) and the simulated (bottom) actuator in contrast. Bending template, 35 Shore, fishbone air channel.

The bending template's soft robots all performed a bending motion while, at the same time, enlarging both horizontally and vertically, varying in their characteristics. Their exact measurements, including their curving degrees and enlargement, described through the elongation and circumference increase as used before, are summarised in Tab. 4.3. Although starting to bend, the simulated actuators mostly exploded long before the real-life air pressure capacity was reached. As an example, see Fig. 4.3.

Bending

Type	Reality				SoRoCAD's Simulation		
	p_{max}	r_{max}	l_{max}/l_0	c_{max}/c_0	r_{max}	l_{max}/l_0	c_{max}/c_0
10 C	0.51 bar	90°	164%	333%	explosion		
10 F	0.61 bar	70°	140%	200%	33°	110%	120%
20 C	0.9 bar	90°	246%	533%	explosion		
20 F	0.51 bar	65°	105%	133%	50°	110%	120%
35 C	1.08 bar	18°	103%	150%	explosion		
35 F	1.61 bar	200°	140%	333%	explosion		
50 C	>2.0 bar	23°	114%	147%	explosion		
50 F	faulty						

Table 4.3: Comparing the bending r_{max} and enlargement composed of the horizontal elongation l_{max}/l_0 and the expansion of the circumference c_{max}/c_0 as used before, of the real and the simulated bending actuators at the time of applying the maximal air pressure p_{max} before exploding. The type is given by the Shore value and the cube (C) or fishbone (F) air channel type.

Elongation The soft robots that resulted from the elongation template, on the other hand, were hardly behaving as expected. While some few samples developed a slight bending of up to 20° or a strong volume expansion, the others did not change at all. The simulations, too, predicted seemingly baseless and inconsistent deformations. Hence, the simulated actuator would sometimes deform spirally or into an eight-shape or it would often explode at relatively low pressure. An example of both the real and the forecast behaviour can be seen in Fig. 4.4. We, therefore, decided to forego contrasting the measured deformations in a table.

S-curve The last class of motion, the S-curve, will not be discussed in detail, either. Only four samples developed an S-curved shape, while the rest was either found to be faulty or, once more, did not deform at all. The SoRoCAD results were again incomparable to our real results. Even though in many cases the actuator indeed shaped up to an S-curve, the curves were so negligibly small that they were hardly measurable. One comparative picture can be seen in Fig. 4.5.

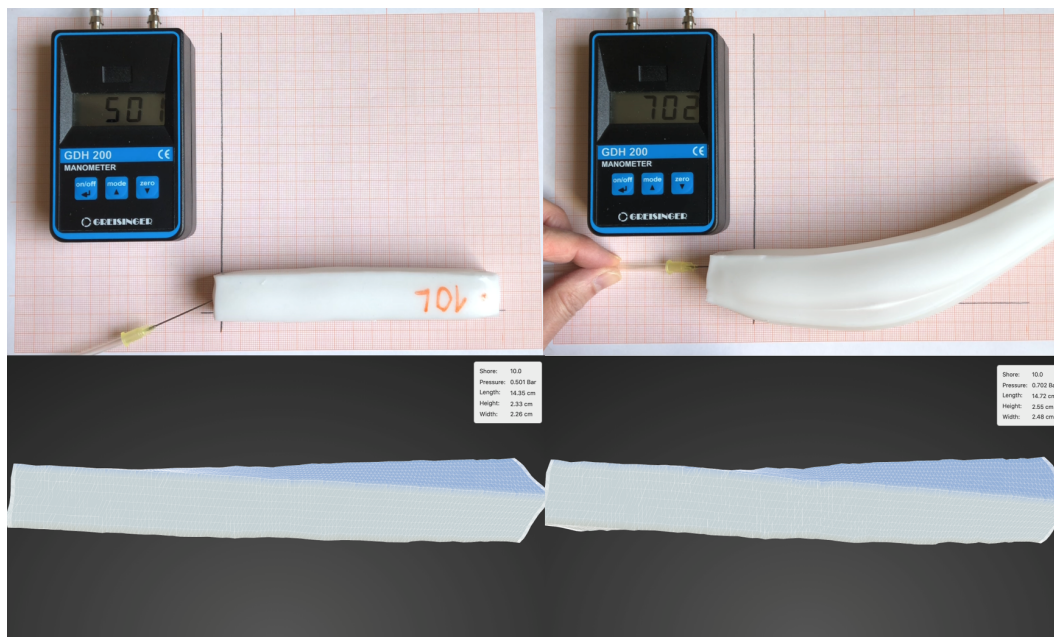


Figure 4.4: The deformations of the real (top) and the simulated (bottom) actuator in contrast. Elongation template, 10 Shore, cubic air channel.

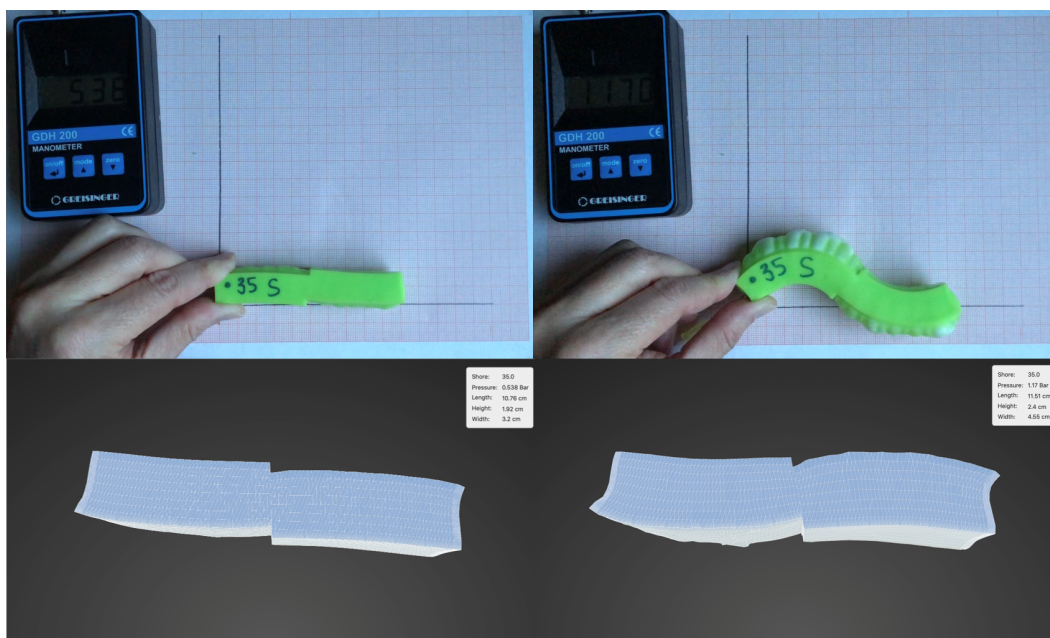


Figure 4.5: The deformations of the real (top) and the simulated (bottom) actuator in contrast. S-curve, 35 Shore, fishbone air channel.

4.3 Discussion

After presenting our results from the technical evaluation, in this section, the outcomes will be interpreted and possible solutions for improving SoRoCAD's simulations on that basis, as well as solving the problems that arose throughout the process of fabrication and actuation of the soft robots, will be discussed.

Casting issues for harder silicones, caused by thin gaps within the moulds

As explained in the results, we were not able to utilize all the soft robots we had in mind. Early in the stage of fabricating soft robots from the moulds that we generated with SoRoCAD, we faced problems when casting the harder silicones. While the uncured 50 and fewer Shore silicones had a fluid consistency, the 70 and higher Shore silicones were kneadable and could hardly be filled seamlessly into the mould. Therefore, the respective soft robots could not be actuated, as the air would stream out of little, almost invisible holes, delivering unusable results for the accuracy testing. A similar problem occurred when casting the smaller, fishbone-based moulds with not only the harder silicones but also with the softer ones. In most cases, the silicone exterior of the actuator was either holey from the start or the walls were so thin that they would easily bust, even before any deformation could be developed. Subsequent corrections often manipulated the design, wherefore this was not expedient, either.

The reason why this problem occurred only when moulding the smaller actuators was that the walls were too thin, although we used SoRoCAD's template designs, which, in many cases, caused a poor distribution of the silicone when moulding and, consequently, the emergence of small bubbles on the ground. A comparison of the wall thicknesses of the small and the large designs can be taken from Fig. 4.6. Additionally, the fine walls constitute vulnerable spots that are likely to break even at low pressure. This, we noted repeatedly.

Unfeasible moulds generated by SoRoCAD

Another problem that resulted from the small designs' thin walls was that we did not manage to fabricate soft robots of the smaller type for 50 or higher Shore silicones, since

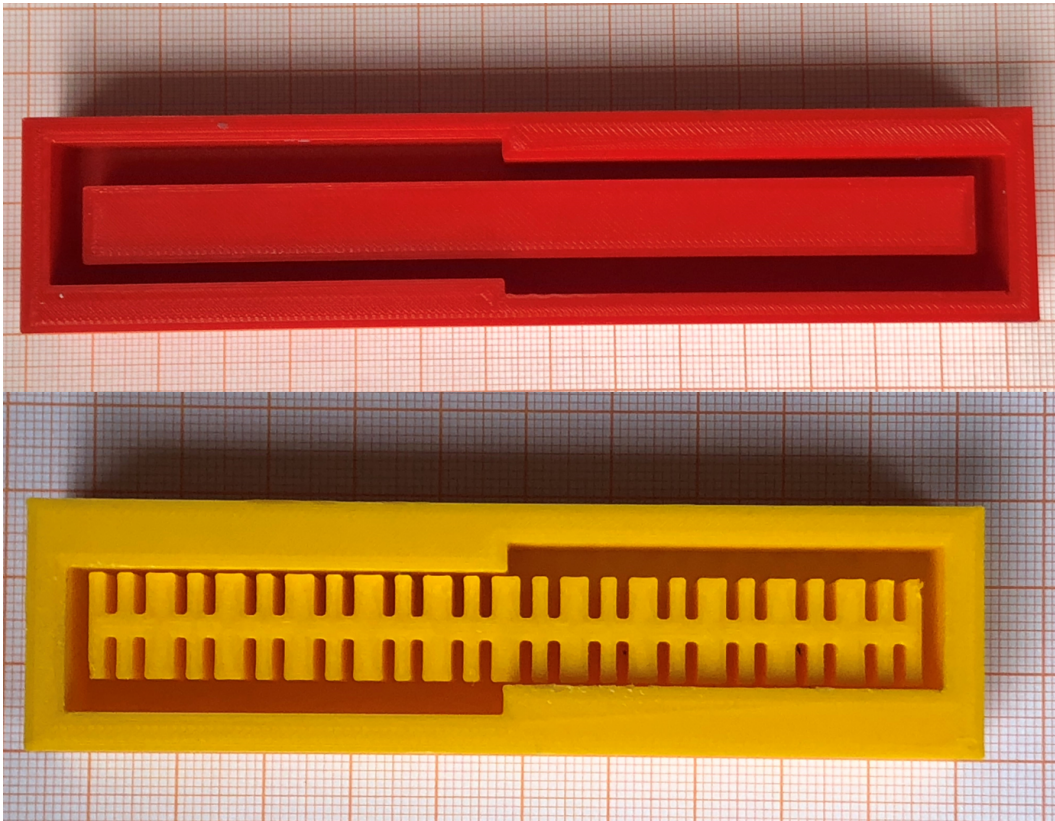


Figure 4.6: Comparison of the larger (top) and the smaller (bottom) S-curve mould designs. The small fishbone mould contains significantly narrower gaps and thin walls when compared to the cube mould.

they just could not be demoulded without being irreparably damaged. The gaps, both within the fishbone structure and for the outer walls, were significantly too narrow in relation to the air channel's width.

At one point, we also noticed that not all air channel types that are available in SoROCAD can be used when planning to generate and cast the moulds to recreate the designed actuators. Some designs, for example the cinder and the snake blocks, contain vertical holes that rule out demoulding the silicone body without rending it. An image to demonstrate the issue can be seen in Fig. 4.7. Although this issue cannot be solved through the existing sliders in the *mold* tab, it can easily be fixed by rotating the air channel blocks around the x-axis. However, this might affect

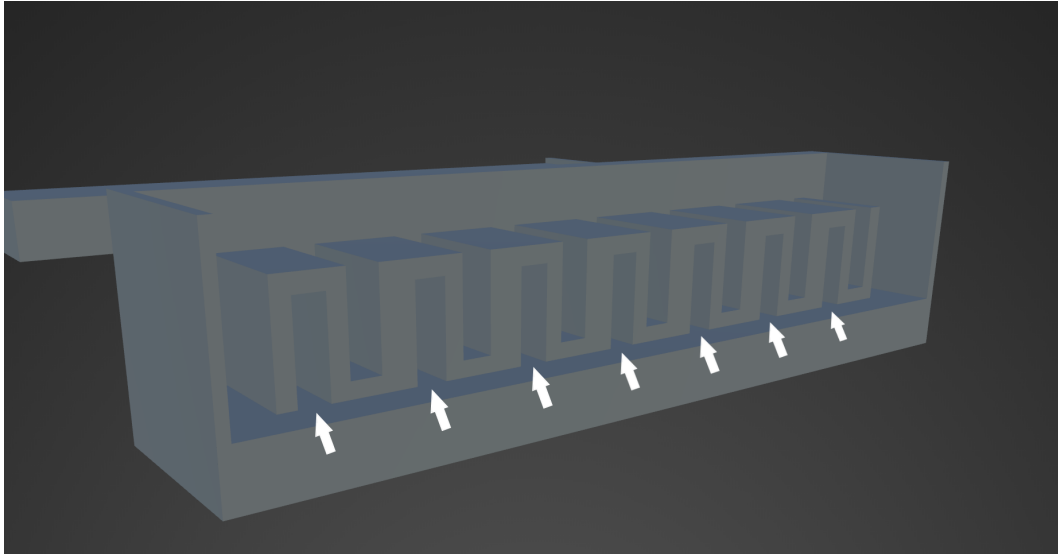


Figure 4.7: The SoRoCAD-generated mould containing snake air channels, in cross section. The white arrows indicate the tunnel-like parts which prevent the demoulding.

the need to rework the whole model to create an equivalent actuator. One possibility to avoid this extra work would be to directly orientate the respective blocks in a way that does not cause problems. Alternatively, another approach could be to add new functionalities to the *mold* tab that allow more modifications, such as rotating the whole actuator when generating the mould.

Approaches to solve moulding issues

Combining these findings, the origin of all fabrication-related problems we faced was that the SoRoCAD-generated templates and, as a result, their moulds, are hardly feasible in practice or need a lot of experience in working with silicone to fabricate the soft robots. The simplest way to solve this would be to adapt the templates by slightly increasing their default wall thicknesses while maintaining the proportions, as we did for the larger designs. Still, even with broader structures in the moulds, hard silicones are at risk to be ruined when demoulding. Hence, besides adapting the templates, the standard air channel type could be changed from fishbone to one that contains fewer gaps to reduce the grip, for example, the cube type. Moreover, the Shore value could be lowered from 50 to around 20 to comply more realistically with the

feasibility and to not raise expectations that might turn out to be false when trying to recreate the simulated actuator in real life.

Furthermore, an interesting approach would be to implement the functionality to SoRoCAD to evaluate the ratios of the current wall thicknesses in relation to the selected air channel type and Shore hardness to then display warnings in case the selected combinations are potentially problematic. This warning could either appear when the user, having designed such an actuator, changes to the *mold* tab to export and 3D print the files for use, or it could be communicated in real-time in the *silicone exterior* tab by displaying a subtle symbol to indicate the risk. The distinction could be based on calculations or the gathered data from a comprehensive evaluation. However, the user does not always use SoRoCAD to fabricate real soft robots, but also to get a feel for the influence of specific parameters or designs. Hence, to not restrict users to feasible designs only, these warnings can just be ignored and all past actuators can be designed and simulated further on.

Display warnings in real time

Although this approach would ideally prevent users from despairing of actuator halves that are stuck in their moulds, including all required restrictions would still significantly reduce the number of designs that can be fabricated with the help of SoRoCAD. Our next approach does, therefore, look at the problem from the angle of casting technology, as a redesign of the moulds that SoRoCAD generates yields improved opportunities for demoulding. Because finding an optimal design from a casting technology point of view requires extensive research and does not constitute the main focus of this work, we will only briefly point out possible approaches hereafter.

When redesigning the moulds to solve the mentioned problem, an important aspect to keep in mind is to maximise the gripping surface of the cast silicone. Our first idea, illustrated in Fig. 4.8, was to integrate the formation of a silicone handle that is flatter than the actuator's main body and can thus be separated from the mould more easily. Once demoulded, this handle can be firmly gripped to pull out the remaining part of the body and simply chopped off after-

Possible redesigns of the moulds

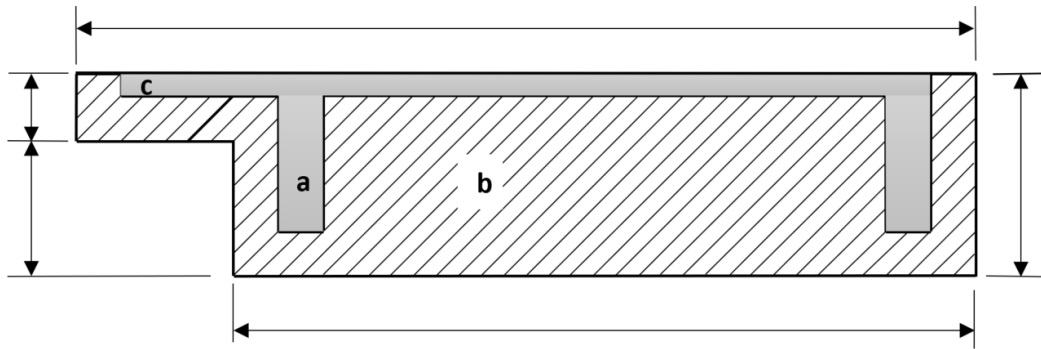


Figure 4.8: The cross section of a redesigned mould (b). The grey background (a) indicates the hollow part of the mould which is cast with silicone. The actuator design is extended by the flat, extra indent (c) to be also filled with silicone to increase the gripping surface.

wards. Another idea was to replace one of the mould's sidewalls with a hinged one. When snapping the wall shut, the process of casting and curing the silicone can be performed as usual. However, for demoulding, the wall can be opened up so that one of the actuator's sidewalls is uncovered and offers a larger gripping surface to demould the whole body. This led us to our third idea of, instead of generating a two-part mould, using moulds that consist of three parts: the top, bottom and middle layer. While the top layer does not differ from the usual one, the new bottom and middle layer result from disassembling the past bottom layer and its surrounding walls. While assembled, the actuator can again be cast and cured as usual, and for demoulding, the contouring middle layer, consisting of the mould's surrounding walls, can be taken away, leading to the maximal gripping surface. Nevertheless, it is important to emphasise that none of these ideas has been tested throughout the technical evaluation.

Discussing the results from testing the simulation

In the following, our results and problems we encountered while actuating and comparing the soft robots and the simulations, will be interpreted and argued. In section 4.2 'Results', we contrasted the real and the simulated behaviour of the class of both the enlargement and the bending actuators in a table. However, the presented values cannot be considered to be representative for evaluating the accuracy of SoRoCAD's simulations. This is because the simulations

differ from the real-life counterparts not only in the intensity of deformation but also in many other aspects. Most notably, in many cases, the deformed shapes of the real actuator and the simulated one cannot be compared, such as all actuators that were constructed with the enlargement template, as can be seen in Fig. 4.2.

Additionally, we could not find any connection between the real maximum capacity of applied air pressure before exploding and the simulated one, neither did we discover noteworthy similarities between the places of explosion. While in some cases, the simulated actuator exploded at a considerably lower pressure than the real one, in other cases the exploding state was not reached for the simulation. This can be seen especially well for the bending template in Fig. 4.3 and Tab. 4.3, which, even though the simulation did indeed perform a bending motion, did therefore not yield useable results as both parties are incomparable. Still, we noticed in both tables that the measured values for the simulation's deformation only slightly differed within the same template, however, they did neither change linearly in relation to the Shore hardness nor did we find any other correlation to explain this behaviour. Additional problems due to incomparability for the elongation and the S-curve classes were described earlier in section 4.2 'Results', wherefore we will not further discuss them.

All these issues can be summed up as inaccurate simulations. The problem lies in the implementation of SoRoCAD, more specifically, in the implementation of the python scene, which is passed to SOFA and used to define the scene that includes the actuator, as well as all required parameters to accurately calculate and present the model's deformation. The unit handling of the framework is quite complex but there is hardly any information available. In brief, as explained in the [SOFA forum](https://www.sofa-framework.org/community/forum/section/sofa/)¹, SOFA generally uses S.I. units. However, if the user explicitly adapts any of the predefined constants, all other values will conform with this adaptation, too, which results in a lot of required conversions. For example, if the gravitational constant is changed from the default 9.81 to 9810, all other length specifications will further

Simulated vs. real
explosions

Inaccuracy of the
simulation caused by
faulty parameters

¹<https://www.sofa-framework.org/community/forum/section/sofa/>
Accessed: April 8, 2021

be interpreted as millimetres, instead of the standard metres. Consequently, all other values whose units depend on metres, explicitly or implicitly, need to be adapted accordingly, including the Shore value we use as one of the main simulation parameters. The Shore value is internally defined by the Young's modulus, which is another way to describe the silicone's hardness and which, again, is defined through the units of force/size² and must therefore be converted, as well. In our tool, we normally interpret length specifications to be given in centimetres, while other units were redefined to fit the user's needs, too. In our python scene file, on the other hand, the conversion of other values was not considered, resulting in massively inconsistent units throughout the scene. As a consequence, the tool's simulations are hardly accurate.

Further sources of
error

But even if these issues were fixed, another problem remains, which distorts the simulation results. In general, SoRoCAD considers ideal soft robots only. To be more specific, the tool expects a perfectly even silicone exterior with neither bubbles nor cracks and assumes that the air pressure is applied to the soft robot seemingly out of nowhere, not requiring punctures, which, in our case, constitute predetermined breaking points. In fact in reality, apart from the actuators that were flawed in the first place, i.e. as explained earlier, they were damaged or contained too thin walls and could therefore not deform properly before exploding, there were other actuators that did not provide useful results, either: numerous actuators did not deform at all, because the air inlets tore open in the beginning and let out the air. However, some of these potential threats can be solved, as explained earlier, and the discrepancies it causes are not expected to be too grave.

4.4 Summary

In this chapter, first, our research question was presented, as well as our motivation and the setup of the technical evaluation we conducted in the course of this thesis. Our goal was to evaluate the SoRoCAD tool with particular regard to the accuracy of the simulations and the feasibil-

ity of fabricating the same actuators as designed with the help of the SoRoCAD-generated moulds. For this, we fabricated a number of soft robots based on different templates, sizes, and air channel types and actuated them by applying air pressure of varying intensities. We, eventually, reconstructed these actuators with the tool and compared the simulation results with our real measurements, which we presented later in this chapter, along with our observations we made throughout the fabrication. In the end, we discussed and interpreted these results and pointed up the underlying problems, as well as introducing possible approaches to solve them. In summary, we experienced severe problems during the fabrication process, most notably because the generated moulds were not always very suitable for the automatically designed template-based actuators, due to the narrow gaps they contained, which restrained the moulding and demoulding. Also, we found that the simulations were not only inaccurate in the intensity of their deformations, but also their overall behaviour. For example, in many cases, the deformed shapes of the simulated and the real actuator were incomparable. Primarily, we suspected inconsistencies in the units of the python scene file that is passed on to the SOFA framework to calculate the simulation.

Chapter 5

Summary and Future Work

In the last chapter, this thesis is summed up, presenting the tasks and outcomes of this work and discussing further ideas for future work to expand and enhance SoRoCAD.

5.1 Summary and Contributions

This thesis deals with the extension of SoRoCAD, a tool for modelling and simulating simple soft actuators, specialised for the novices in Soft Robotics, by adding new functionalities, resolving previous issues and technically evaluating the tool.

The field of Soft Robotics is relatively new and describes robots that are inspired by living organisms and constructed from soft, compliant materials to open up new application fields. Although there exist various actuation techniques and materials that can be used, this work only covers silicone-actuators that are actuated pneumatically. The actuation behaviour depends on a number of parameters. In the case of the simple finger actuators that are considered in this work, important parameters are the inner air chamber design, the ratio of the wall thicknesses of the ex-

Soft Robotics

terior, the silicone's hardness and the applied air pressure.

Presenting
SoRoCAD

To help the novices to start with Soft Robotics and get a feel for the influence of different parameter combinations, SoRoCAD was created. The basic workflow of designing a functional soft robot, as well as the main user interaction elements are introduced and current limitations are worked out, mostly based on the results of a previous user study, which was conducted by Strüver [2020]. Most of the problems that were identified originate from misleading or inconvenient user interface and interaction elements, such as the viewport interaction buttons that often could not be identified, or the input methods that were not easily operable. Other problems refer to impaired ease of use and a restricted complexity of the actuators that can be designed. These issues were addressed by adding new functionalities to the tool, for instance, by enabling to select and modify multiple elements at once or to change the horizontal and vertical position of air channel blocks.

Resolving user
interface and user
interaction issues

Distribution issues
due to the SOFA
framework

Moreover, at the time existing distribution problems are explained, that prevented the tool from being fully used on any other device than the Mac it was developed on since it would crash whenever a simulation was started. This issue was caused by the fact that SoRoCAD's simulation feature is being handled by the SOFA framework, which, at the time, had to be manually installed on the device prior to starting the simulation. To evade this additional expense imposed on the user, a run script build phase to manage the installation of SOFA was integrated into the project, which is briefly outlined in this work.

The setup of the
technical evaluation

Further on, the technical evaluation, that was conducted to test both the feasibility of the SoRoCAD-generated template designs and moulds, and the accuracy of the simulations is presented in this thesis. For this evaluation, various soft robots of each template type - bending, elongation and enlargement, plus, additionally the S-curve - and of different Shore hardnesses, were recreated in reality, using the moulds that were created by SoRoCAD. Their deformations when pneumatically actuated, including the length and circumference changes, the bending, and the maximal air pressure that can be applied before exploding, were

then measured and compared to their simulation's measurements. Finally, the observations from fabricating the actuators, as well as the comparison results are then depicted and possible reasons and solution approaches are discussed. As a result of the technical evaluation, it was discovered that the SoRoCAD-generated moulds for the templates are often not practicable to recreate the real counterpart and, even more importantly, that the simulation results are mostly incomparable to the deformations observed in real, presumably because of a confusion of the units in the scene-defining script.

SoRoCAD shows problems with the generated moulds and the simulation's accuracy.

5.2 Future Work

In the following, several ideas for future work on the tool are presented. Some ideas are based on the outcomes of the technical evaluation during this thesis, others follow up on the first task of this thesis, on the software's user interface and the existing user interaction elements.

5.2.1 Results of the Technical Evaluation

The technical evaluation of SoRoCAD provided two main improvable aspects from the tool, both of which were discussed in section 4.3 'Discussion'. These include the mould generation and the available templates, as well as the accuracy of the simulation.

The Simulation Accuracy

When comparing the two behaviours of both the actuated soft robot and its SoRoCAD-simulation, significant discrepancies were detected that relate to the deformed shape, the intensity of the enlargement and the maximum capacity of applied air pressure before exploding. These differences are assumed to be caused by an erroneously defined python scene, more precisely, by steadily inconsistent use of units

Inconsistent usage of units cause inaccurate simulation results.

that, besides that, do not even match with the units that are used and interpreted by the user. A thoroughly reworked python scene file is therefore crucial to maintain useful simulation results.

The Mould Design

The current mould design causes problems in many ways.

During the process of fabricating the soft robots to test the deformations on, at least half of the planned samples could not be used, because they either could not be demoulded, or the final actuator consisted of too thin walls to offer usable results, as they would immediately tear open when actuated. These issues occurred the harder the cast silicone, i.e. the higher the Shore value, was and were caused by too thin wall thicknesses and even thinner gaps within the fishbone-shaped air channels.

A real-time evaluation of selected parameters to warn the user.

As depicted in section 4.3 'Discussion', possible solutions could be to equally increase all wall thicknesses for each template design and to modify the default air channel type to contain fewer gaps, as the cube type. Besides that, the standard Shore value could be decreased to comply with the real scope that can be used to test the designs. Besides that, a warning-functionality could be implemented to SoRoCAD to, in real-time, evaluate the combination of the currently selected parameters and deliver a warning whenever the generated moulds do not conform to the conditions that are needed to successfully create a functional soft robot.

Add further controls to manually modify the moulds.

Additionally, SoRoCAD's *mold* tab could be extended by further controls that allow more manual modifications of the generated moulds. For example, adding a slider to rotate the hollow actuator-shaped inner of the mould could solve the present issues that for several actuator designs, the generated moulds are faulty and, hence, impossible to use. Those designs include cinder or snake-shaped air channel types, actuators that deform to a vertical S-curve, and non-linear shapes that were modelled by means of the newly appended sliders to reposition air channel blocks. Following up on this approach, it is worth thinking

Redesigning the moulds is necessary.

about redesigning the moulds from a casting-technology point of view. As portrayed already, there are several concepts that are yet to be tested and contain the integration of a silicone-cast handle to grasp and pull on or manipulating one or all surrounding walls to become hinged or even removable after the silicone was cast and cured, to increase the gripping surface.

5.2.2 User Interface & User Interaction

The next group that is capable of development relates to SoRoCAD's user interface and user interactions. At the moment, the tool is rather limited in the complexity of designable soft robots, which contain solely one-fingered actuators. One possibility of expanding it would be to allow the modelling of multiple actuators that can be connected through joints to be actuated simultaneously. This could also be appended by the functionality to actuate each component individually to achieve even more interesting results. Moreover, to expand SoRoCAD in its main role as a learning programme, more air channel designs could be added, all of which are introduced with short explanations on how they affect the actuator's behaviour, similarly to the tutorial view in the *silicone exterior* tab.

Finally, since, throughout this thesis, SoRoCAD was modified in several ways, a reasonable move would be to reiterate the user study, presented in Strüver [2020], to evaluate the current state of the tool.

Expand the tool to allow more complex soft robots.

Extending the list of air channel types and add explanations.

Repeat the previous user study.

Bibliography

Mads Bering Christiansen and Jonas Jørgensen. Augmenting Soft Robotics with Sound. In Tony Belpaeme, James Young, Hatice Gunes, and Laurel Riek, editors, *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 133–135, New York, NY, USA, 2020. ACM. ISBN 9781450370578. doi: 10.1145/3371382.3378328.

Kristian Gohlke. Exploring Bio-Inspired Soft Fluidic Actuators and Sensors for the Design of Shape Changing Tangible User Interfaces. In Masa Inakage, Hiroshi Ishii, Ellen Yi-Luen Do, Jürgen Steimle, Orit Shaer, Kai Kunze, and Roshan Peiris, editors, *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, pages 703–706, New York, NY, USA, 2017. ACM. ISBN 9781450346764. doi: 10.1145/3024969.3025039.

Weicheng Huang, Xiaonan Huang, Carmel Majidi, and M. Khalid Jawed. Dynamic Simulation of Articulated Soft Robots. *Nature communications*, 11(1):2233, 2020. doi: 10.1038/s41467-020-15651-9.

Josie Hughes, Shuguang Li, and Daniela Rus. Sensorization of a Continuum Body Gripper for High Force and Delicate Object Grasping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6913–6919. IEEE, 2020. ISBN 978-1-7281-7395-5. doi: 10.1109/ICRA40945.2020.9196603.

Filip Ilievski, Aaron D. Mazzeo, Robert F. Shepherd, Xin Chen, and George M. Whitesides. Soft Robotics for Chemists. *Angewandte Chemie (International ed. in English)*, 50(8):1890–1895, 2011.

- Jonas Jørgensen. Appeal and Perceived Naturalness of a Soft Robotic Tentacle. In Takayuki Kanda, Selma Šabanović, Guy Hoffman, and Adriana Tapus, editors, *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 139–140, New York, NY, USA, 2018. ACM. ISBN 9781450356152. doi: 10.1145/3173386.3176985.
- Sangbae Kim, Cecilia Laschi, and Barry Trimmer. Soft Robotics: A Bioinspired Evolution in Robotics. *Trends in biotechnology*, 31(5):287–294, 2013.
- Christopher Kopic and Kristian Gohlke. Inflatibits – A Modular Soft Robotic Construction Kit for Children. In Saskia Bakker, Caroline Hummels, Brygg Ullmer, Luc Geurts, Bart Hengeveld, Daniel Saakes, and Mendel Broekhuijsen, editors, *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 723–728, New York, NY, USA, 2016. ACM. ISBN 9781450335829. doi: 10.1145/2839462.2872962.
- Li-Ke Ma, Yizhong Zhang, Yang Liu, Kun Zhou, and Xin Tong. Computational Design and Fabrication of Soft Pneumatic Objects with Desired Deformations. *ACM Transactions on Graphics*, 36(6):1–12, 2017. ISSN 0730-0301. doi: 10.1145/3130800.3130850.
- Ramses V. Martinez, Carina R. Fish, Xin Chen, and George M. Whitesides. Elastomeric Origami: Programmable Paper-Elastomer Composites as Pneumatic Actuators. *Advanced Functional Materials*, 22(7):1376–1384, 2012. ISSN 1616301X. doi: 10.1002/adfm.201102978.
- Anna Dagmar Bille Milthers, Anne Bjerre Hammer, Jonathan Jung Johansen, Lasse Goul Jensen, Elizabeth Ann Jochum, and Markus Löchtefeld. The Helpless Soft Robot - Stimulating Human Collaboration through Robotic Movement. In Stephen Brewster, Geraldine Fitzpatrick, Anna Cox, and Vassilis Kostakos, editors, *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, New York, NY, USA, 2019. ACM. ISBN 9781450359719. doi: 10.1145/3290607.3312807.

- Daniela Rus and Michael T. Tolley. Design, Fabrication and Control of Soft Robots. *Nature*, 521(7553):467–475, 2015.
- Yu She, Sandra Q. Liu, Peiyu Yu, and Edward Adelson. Exoskeleton-Covered Soft Finger with Vision-Based Proprioception and Tactile Sensing, 2019. URL <http://arxiv.org/pdf/1910.01287v2>.
- Robert F. Shepherd, Filip Ilievski, Wonjae Choi, Stephen A. Morin, Adam A. Stokes, Aaron D. Mazzeo, Xin Chen, Michael Wang, and George M. Whitesides. Multigait Soft Robot. *Proceedings of the National Academy of Sciences of the United States of America*, 108(51):20400–20403, 2011. doi: 10.1073/pnas.1116564108.
- Robert F. Shepherd, Adam A. Stokes, Jacob Freake, Jabulani Barber, Phillip W. Snyder, Aaron D. Mazzeo, Ludovico Cademartiri, Stephen A. Morin, and George M. Whitesides. Using Explosions to Power a Soft Robot. *Angewandte Chemie (International ed. in English)*, 52(10):2892–2896, 2013.
- Adam A. Stokes, Robert F. Shepherd, Stephen A. Morin, Filip Ilievski, and George M. Whitesides. A Hybrid Combining Hard and Soft Robots. *Soft Robotics*, 1(1):70–74, 2014.
- Jakob Strüver. Evaluating SoRoCAD: Enabling Users to Design Custom Soft Robotics. Master’s thesis, RWTH Aachen University, Aachen, October 2020.

Index

Actuator	2
CGAL plugin	39
Collaborative robots.....	3
Evaluation.....	43–57
Future work	61–63
PneuNets	1–2
Run script build phase	40
Shore Hardness	1
SOFA framework.....	39–40
SofaPython plugin	39
Soft Robotics.....	1–4
Soft Robots plugin	39
SoRoCAD	3–4
Universal gripper.....	11–12

