

# *SoRoCAD 3.0: A Cross-Platform Approach to Soft Robotics Design*

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

*by*  
*Philip Le*

Thesis advisor:  
Prof. Dr. Jan Borchers

Second examiner:  
Prof. Dr. Ulrik Schroeder

Registration date: 09.01.2024  
Submission date: 10.05.2024



# 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

<b>Abstract</b>	<b>ix</b>
<b>Überblick</b>	<b>xi</b>
<b>Acknowledgments</b>	<b>xiii</b>
<b>Conventions</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis Outline . . . . .	2
<b>2 Related Work</b>	<b>3</b>
2.1 Soft Robotics Applications . . . . .	3
2.2 Design Tool kits . . . . .	5
2.2.1 Prior Theses . . . . .	5
2.2.2 Tool kits . . . . .	7
2.3 Simulation Tools . . . . .	9
<b>3 SoRoCAD 3.0</b>	<b>11</b>

---

3.1	Motivation and Goals . . . . .	11
3.2	Previous Design and Issues . . . . .	12
3.3	Framework . . . . .	13
3.4	Actuator Design . . . . .	15
3.5	Mold . . . . .	17
3.6	Simulation Feature . . . . .	20
<b>4</b>	<b>Evaluation</b>	<b>25</b>
4.1	Technical Evaluation . . . . .	25
4.1.1	Distribution and Installation . . . . .	25
4.1.2	Simulation . . . . .	26
4.2	User Feedback . . . . .	27
4.2.1	Setup . . . . .	27
4.2.2	Participants . . . . .	27
4.2.3	Procedure . . . . .	28
4.2.4	Tasks . . . . .	28
4.2.5	Results . . . . .	29
4.3	Discussion . . . . .	30
4.4	Limitations . . . . .	31
<b>5</b>	<b>Summary and Future Work</b>	<b>33</b>
5.1	Summary and Contributions . . . . .	33
5.2	Future Work . . . . .	34

<b>A Study Materials</b>	<b>35</b>
<b>B SoRoCAD Files and Resources</b>	<b>39</b>
<b>Bibliography</b>	<b>41</b>
<b>Index</b>	<b>45</b>



---

# List of Figures and Tables

2.1	<i>AirPinch</i> : A pneumotactile feedback interface . . . . .	4
2.2	<i>A soft pneumatic grasping system</i> . . . . .	5
2.3	<i>Inflatibits</i> : Example of a simple gripper . . . . .	8
2.4	<i>MorpheusPlug</i> : A View of the Autodesk Fusion 360 add-on. . . . .	9
2.5	<i>VoxCAD</i> : Simulating soft bodies with voxels. . . . .	10
3.1	<i>Template Selection</i> : The first window that appears after starting SoRoCAD. . . . .	15
3.2	<i>Actuator Design</i> : Interface of the <i>Air channel</i> tab and the <i>Silicone Exterior</i> tab . . . . .	16
3.3	<i>Mold View</i> : Example of a mold for an actuator with uneven silicone exterior . . . . .	18
3.4	<i>Actuator</i> : An example of an actuator that cannot be demolded with our molding technique. . . . .	20
3.5	<i>Simulation Results</i> : Example of a simulation of an bending actuator. . . . .	21
3.6	<i>Overview</i> : Tested libraries with volumetric mesh generation capabilities . . . . .	23
A.1	<i>Study materials</i> : Informed Consent Form . . . . .	36
A.2	<i>Study materials</i> : Questionnaire . . . . .	37

A.3 <i>Study materials: Task 1</i> . . . . .	38
B.1 <i>Actuator: Twisting motion</i> . . . . .	39

# Abstract

The goal of this thesis is improving the accessibility and usability of SoRoCAD, a CAD tool which specializes in teaching soft robotics and fabricating soft pneumatic actuator. SoRoCAD tries to lower the barrier of entry for novice users for learning soft robots while not limiting the design options for pneumatic actuators. In this work, the design process and the simulation feature are adjusted. This tool also enables users to build emotion and mood conveying actuators which can be used to explore shape changing interfaces.





# Überblick

Das Ziel dieser Abschlussarbeit ist die Verbesserung der Verfügbarkeit und Nutzbarkeit von SoRoCAD, ein CAD Tool, das darauf spezialisiert ist Soft Robotics beizubringen und pneumatische Aktuatoren herzustellen. SoRoCAD versucht den Einstieg für Anfänger, die Soft Robots lernen möchten, zu erleichtern, ohne dabei die Designoptionen für pneumatische Aktuatoren einzuschränken. In dieser Arbeit wird der Designprozess und die Simulationfunktion angepasst. Dieses Werkzeug ermöglicht außerdem auch Aktuatoren zu erstellen, die bestimmte Emotionen oder Stimmungen vermitteln, welche auch genutzt werden können um formverändernde Interfaces zu erforschen.



# Acknowledgments

First, i would like to express my deepest gratitude to my advisor Anke Brocker for all her efforts, guidance and valuable advice to improve this thesis. My sincere thanks should also go to Sarah Sahabi who also guided me as my second advisor.

I also would like to thank every study participant for their time and feedback.

I am also grateful for everyone at the i10 for assisting me by providing advice or feedback.



# Conventions

Throughout this thesis we use the following conventions:

- The thesis is written in American English.

Short excursuses are set off in colored boxes.

**EXCURSUS:**  
Excursuses are set off in orange boxes.

Where appropriate, paragraphs are summarized by one or two sentences that are positioned at the margin of the page.

This is a summary of a paragraph.



# Chapter 1

## Introduction

*“Conventional robots are made of rigid parts that are vulnerable to bumps, scrapes, twists and falls. In contrast, researchers worldwide are increasingly developing robots made from soft, elastic plastic and rubber that are inspired by worms, starfish and octopuses. These soft robots can resist many of the kinds of damage, and can squirm past many of the obstacles, that can impede hard robots.”*

—Charles Q. Choi

The field of Soft Robotics is rapidly evolving with robots that are partially or entirely made of soft and compliant materials like silicone. This makes them not only more flexible, affordable to maintain and build, and safer for humans to work with, but they also adapt better in unstructured environments compared to their rigid counterparts Zhao et al. [2021]. Their design is often bio-inspired which makes them ideal for applications such as gripping, locomotion and biomedical devices Xavier et al. [2022]. Soft Robots are also used for research in human-computer interaction, for example soft robots as shape changing interfaces. Although there are many advantages, fabricating soft robots is not an trivial task. Knowledge in different areas is required, for example embedded system, material science or experience with computer-aided design (CAD) tools. Gen-

Soft Robots: Inspiration  
and Composition

eral purpose CAD tools have a high barrier of entry, especially for beginners who just want to fabricate and learn about their first soft robot. Although using a CAD tool is not the only way to design an actuator, it is recommended since designing and fabricating an actuator requires high precision. SoRoCAD stands for Soft Robotics Computer-Aided Design and is a specialized CAD tool for teaching soft robotics by aiding novice users in designing, simulating and fabricating an pneumatic actuator.

Excursus:  
*Actuator*

**ACTUATOR:**

The part of a soft robot which converts energy into motion is called an actuator. We will mostly focus on soft pneumatic actuators which convert air pressure into a desired movement.

Soft robots are commonly actuated pneumatically, hydraulically or by using cables or strings.

## 1.1 Thesis Outline

Following the brief introduction of soft robotics and SoRoCAD, related work will be presented in the next chapter. Chapter 3 presents the main focus of this work, the latest version of SoRoCAD by describing current issues and how they are addressed. The fourth chapter provides a technical evaluation and user feedback. The last chapter summarizes this work and provides ideas for future work.



## Chapter 2

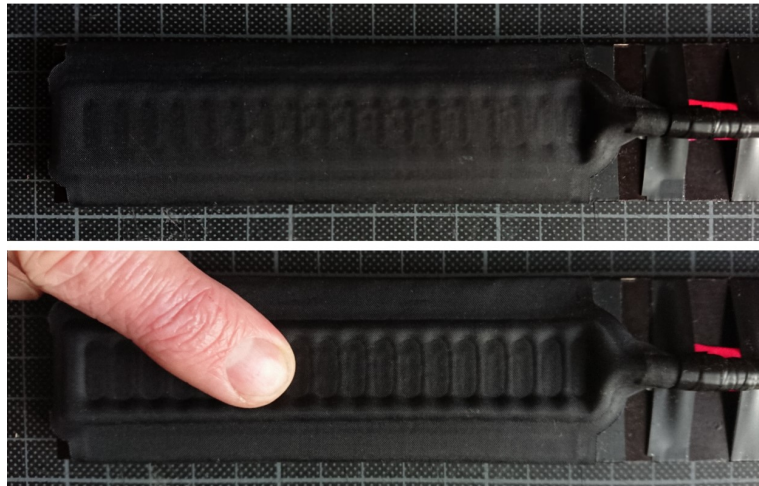
# Related Work

In the following chapter, we will present some work related to this thesis. The first section introduces applications for soft robots, the second section shows different design tool kits and their approaches to teach soft robotics and lastly simulation tools are presented.

### 2.1 Soft Robotics Applications

Gohlke et al. [2022] present a prototype of a textile-pneumatic force feedback system that can serve as a shape-changing interface, called AirPinch. AirPinch uses the natural deformation properties of textiles, like TPU-coated nylon fabrics, to create an interface that gently pinches a user's fingertips. This pneumotactile feedback can be used as non-visual interaction for smart home appliances or as tangible interaction with computer games. Two textile air chambers are placed laterally on both sides of the user's fingertips. Sliding the finger along the touch fader actuates the air chambers and depending on the finger position a pinch or tactile pulses can be felt. The interface consists of a pressure sensitive strip and the described air channels which are mounted together on the rigid base plate. There is also the option to design the fader with different tangible patterns. In Figure 2.1, we can see one of the nine pre-

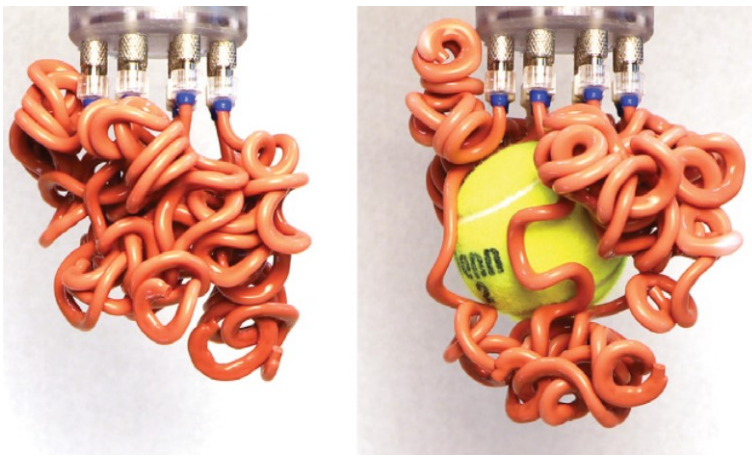
AirPinch: Application of soft robots as an interface



**Figure 2.1:** Source:Gohlke et al. [2022]. *Top:* AirPinch's not actuated fader. *Bottom:* The same fader, actuated.

A soft pneumatic grasping system

sented pattern which appears to be similar to tangible tick marks on a slider. AirPinch's fader is actuated by a stepper motor that can modulate the air pressure continuously. Another application is using soft robots not as interface between human and machine but as medium between human and environment. A gripping soft robot could be used in an environment that is unfriendly to humans. Becker et al. [2022] present a nondeterministic grasping system consisting of multiple, highly compliant filaments that are pneumatically actuated. This grasping system moves away from common gripper with tentacle or finger actuators. Instead long and slender, pneumatic actuated filaments are used. These filaments highly curve and adapt to the object they are grasping. Multiple filaments are distributed around two concentric circles and are actuated by a single pump. The grasping ability depends on the positioning of the attached filaments and the air pressure. This soft robot does not need any feedback control but instead relies on the mechanical properties of the filaments. Contrary to this grasping tool, rigid gripper robots need some kind of feedback control to not damage the object or to even lift the object. In Figure 2.2 an array of actuated filaments can be seen. The filaments hang straight down when not actuated.



**Figure 2.2:** Source:[Becker et al., 2022]. *Left:* 12 actuated filaments *Right:* The same filaments grasping a tennis ball.

## 2.2 Design Tool kits

### 2.2.1 Prior Theses

SoRoCAD is a project project of the Media Computing Group at RWTH Aachen University. This work builds on the work of different bachelor and master theses which are briefly introduced in this section.

The first implementation of SoRoCAD was made in Swift by [Timchenko, 2019]. Swift is Apple’s own programming language and has a better native performance than Python and Java. Furthermore it has a more readable syntax than Objective-C, the other prominent programming language on macOS. SoRoCAD follows a WYSIWYG approach for designing an pneumatic actuator and its suitable mold. The blueprint of SoRoCAD’s user interface is based on industry-standard 2D and 3D graphics software like Fusion 360 and AutoCAD from [Autodesk](https://www.autodesk.com/)<sup>1</sup> and [Adobe Photoshop](https://www.adobe.com/products/photoshop.html)<sup>2</sup>.

Following the first implementation a redesign based on user feedback was done by Strüver [2020]. In this version, the workflow and user interface was addressed. While

SoRoCAD’s first version

SoRoCAD’s redesign

<sup>1</sup> <https://www.autodesk.com/>

<sup>2</sup> <https://www.adobe.com/products/photoshop.html>

there was less restriction in placements of air channel blocks in the first version, the mold for the corresponding actuator were difficult to 3D-print due to high verticality. Furthermore multiple air channels could be created in a single actuator which can be confusing and overwhelming for users. The decision to limit the actuator in favor of a less overwhelming learning experience was made. Air channel blocks are now placed next to each other along the X-axis which leads to an finger or tentacle-like actuator. These kind of actuators are often used in grippers and are able to produce different kind of motions. Since SoRoCAD is geared towards beginners with little to no experience with soft robotics, a lot of domain knowledge like mold generation has to be embedded to support the user. The revised workflow which splits the fabrication of the actuator design in the three main tabs, *Air Channel*, *Silicone Exterior* and *Mold* is still present in the newest version of SoRoCAD.

Extending SoRoCAD

The most recent work before SoRoCAD 3 was an extension of SoRoCAD, done by Sahabi [2021]. This work improved the usability by adding features for better orientation in the 3D scene and revising the user interface. This includes centering the actuator in the 3D scene and adding a button to reset the camera focus on the actuator. This version also made it possible to select and edit multiple blocks at once which was not possible in the previous version. Additionally, decimal numbers were changed to only show one fraction digit and units were added. Also, the amount of provided information was adjusted, for example there was not enough explanation what the parameters for the simulation mean or the hint pop up for the silicone exterior were simply ignored. Further changes were made, like adding an air channel block at a chosen position, instead of only on the far right side, next to the last block. Various bug fixes were implemented, like crashes when switching immediately from Air Channel tab to the Mold tab or not synchronized slider and their corresponding number input fields. This last version of SoRoCAD made in Swift is also presented by Bocker et al. [2022].

### 2.2.2 Tool kits

Although there are various tool kits with different target groups, they all share the common goal with SoRoCAD to teach and make soft robotics more accessible. This section presents some of these projects.

Siloseam: Fabricating  
pneumatic bladders

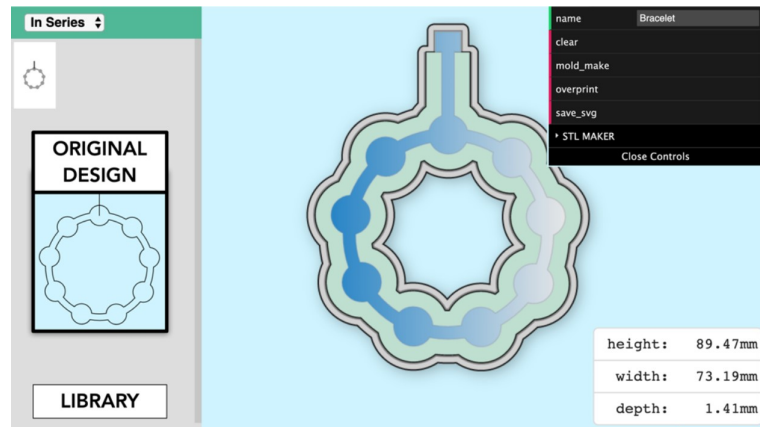
Moradi and Torres [2020] provide a framework which lets the user interact with silicone as primary material to fabricate pneumatic actuated bladders. Makers can explore silicone's potential to take on new forms. Silicone has a non-stick property which makes it a forgiving material for demolding and being a rubber makes it suitable for airtight applications. Although, it is affordable to acquire, working with silicone takes time, mainly due to long cure times. The design process begins with creating a scalable vector graphic (SVG) of the bladder shape that is then loaded in Siloseam's design tool. In this CAD tool the image is extruded to a 2.5D model for the mold. After 3D-printing the mold, casting and curing the first layer, a separator is added before casting the next layer. This method takes advantage of silicone's non-stick property to create an air chamber by removing the separator after the second layer is cured. Technically any two dimensional shape can be used as base for a bladder which can be augmented with lights, but the movement is limited to inflation of these bladders. The interface of Siloseam's design tool is seen in Figure 2.3.

Contrary to Siloseam, Inflatibits lets the user avoid having to fabricate silicone actuators themselves by providing a modular construction kit. Kopic and Gohlke [2016] approach teaching pneumatic systems in a playful manner, targeting a younger audience. Although, users are not required to have access to a 3D-printer, silicone casting equipment or any CAD experience, they still have to buy the tool kit itself. Siloseam's building blocks are compatible with Lego building blocks. The two main parts for movement of the actuators are the expanding air chambers and rigid or bending restrictors. With this tool kit not only straight inflation but also bending movements can be produced.

Inflatibits: Learning soft  
robotics in a playful  
manner

Kim et al. [2021] created a tool kit named MorpheesPlug

MorpheesPlug:  
3D-printing shape  
changing interfaces

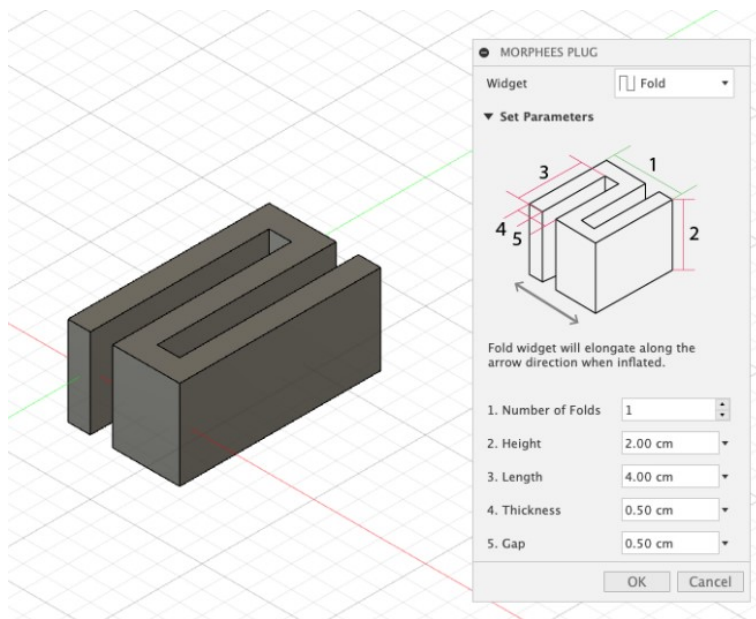


**Figure 2.3:** Source: Moradi and Torres [2020]. A SVG (original design), is loaded into Siloseam’s design tool. The tool generates a mold and the corresponding separator.

which lets the user design 3D-printable, inflatable shapes. There are six basic shapes which are called widgets. Although, fabricating these widgets requires a 3D-printer with a specific, flexible filament such as thermoplastic polyurethane (TPU), no silicone casting and curing is required. The software is available as add-on for Autodesk Fusion 360, a paid, general purpose CAD tool. In this add-on, the user can set parameters like size, thickness of the walls and frequency of the widget. The focus lies on exploring shape-changing interfaces. The interface of MorpheesPlug add on can be seen in Figure 2.4.

MiuraKit:  
Origami-inspired soft  
robots

Cui et al. [2023] developed a hands-on construction kit for shape changing interfaces called MiuraKit. MiuraKit consists of a design software with simulation feature, tube-shaped origami actuators and connectors. The web-based design tool’s intended use is designing more complex shape changing interfaces or soft robots. For exploring and learning about pneumatic actuation, a hands-on configuration is sufficient. A construction made of these actuators can be disassembled and reassembled into new designs. Supported movements are elongation and bending, which can be freely combined by using the provided connectors. Although the user can buy the kit, Cui et al. [2023] provide instructions to build the kit.



**Figure 2.4:** Source: Kim et al. [2021]. The snake shaped widget can be adjusted by setting the values for different parameters. Afterwards the widget itself can be 3D-printed instead of a mold.

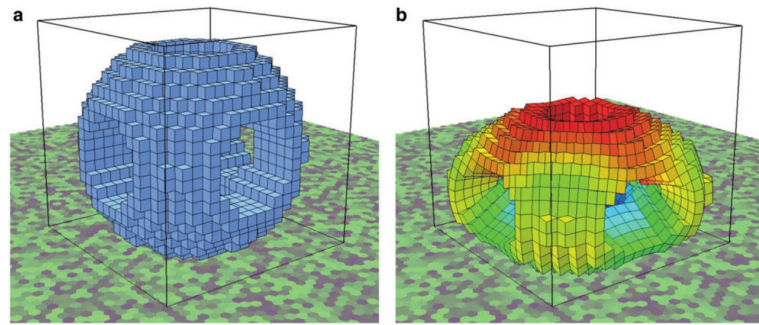
SoRoCAD is not only a pneumatic actuator design software but also tries to teach the basic principles of pneumatic actuation. The presented design tool kits also provide templates for various actuator or the actuators themselves, but only SoRoCAD allows users to design an actuator based on a emotion or mood. Also, most of the knowledge is conveyed in SoRoCAD's software itself by designing and simulating an actuator or exploring the different templates. For this, no physical actuators are required, but the user still has the option to fabricate an actuator with the help of SoRoCAD's mold generation. Only then a 3D-printer is required.

The gap SoRoCAD fills

## 2.3 Simulation Tools

Hiller and Lipson [2014] present a simulation tool which discretizes a three dimensionall space into cubes, so called

VoxCAD: Simulating soft bodies with voxels



**Figure 2.5:** Source: Hiller and Lipson [2014]. A simulation of a bouncing hollow ball under gravity, (a) before the impact and (b) at the highest deformation.

voxels. Compared to freeform meshes, voxels in lattices, do not have a high local accuracy of a simulated mesh. However, some advantages of simulations that use voxels are faster computation, individual stiffness between voxels and more efficient handling of different materials in a simulation. Soft materials are simulated by applying spring forces between blocks, such that they are kept together until a force greater than the spring force is applied. Additionally, each block has a mass and move and rotate in all directions. A simulation of a model starts with voxelizing the model, then internal forces are calculated and the positions of the blocks are updated. Although it is possible to use VoxCAD in Python, for example with [evosoro](https://github.com/skriegman/evosoro)<sup>3</sup>, this kind of simulation is not suitable for SoRoCAD. There are three reasons for this, the actuator in SoRoCAD are made of a single material and high accuracy is needed. Also exporting the simulation data would leave us with voxels which we cannot use to map deformations of our actuator. An example simulation of a hollow ball can be seen in Figure 2.5.

The tool we use for our simulation feature is SOFA and will be introduced in Chapter 3.6 “Simulation Feature”.

<sup>3</sup> <https://github.com/skriegman/evosoro>



## Chapter 3

# SoRoCAD 3.0

### 3.1 Motivation and Goals

The following Chapter presents SoRoCAD by giving motivation and goals, describing the previous design, the general workflow and then the changes in the current version. The workflow of SoRoCAD tries to omit multiple silicone casting and curing iterations by providing the user with a simulation feature. After a simulation, the design can be reviewed and adjusted which saves the user time, effort and the frustration of a failing actuator design.

The work in this thesis builds upon the previous version of SoRoCAD by [Sahabi, 2021]. This version, in the following referred as SoRoCAD 2, is implemented in Swift and is only available on macOS. In order to make SoRoCAD more accessible, an implementation which can run on multiple platforms is needed. As mentioned in Chapter 2 “Related Work” the previous work had their focus on improvements in user interface and user interaction. This thesis focuses more on making SoRoCAD more accessible as a learning tool while maintaining the overall workflow, but also uses the opportunity to address some issues that occur when working with SoRoCAD.

## 3.2 Previous Design and Issues

SoRoCAD 2.0:  
Template Selection

SoRoCAD 2 was already capable of designing and simulating soft pneumatic actuators and even creating a mold for actuators with an even surface and simple air channels was possible. When starting SoRoCAD, the first window the user can interact with is the *Template Selection* window. Here are four options the user can choose from, templates for three basic movements and the option to start with an empty template. After selecting one of the basic movements, elongation, bending and enlargement, the user can then specify the length, height and width of the finished actuator.

Issue: Disconnected air  
channel when rotating  
blocks

Afterwards a finished actuator is displayed in the *Silicone Exterior* tab. When choosing the *Empty Template* the workflow begins in the *Air Channel* tab, where multiple of the five different basic building blocks can be placed, rotated and scaled. These blocks are initially placed in a line, but they can be moved in Y and Z direction independently. Generally, each block should intersect or have contact with its neighbors, so that the air can flow through the entire air channel. In SoRoCAD 2, an air channel block only has enough contact to its neighbor if the rotation is a integer multiple of 90°. Furthermore, some block placements lead to an air channel consisting of disconnected parts.

Issue: Air channel block  
is outside of silicone  
exterior

After constructing the air channel, the workflow continues in the before mentioned *Silicone Exterior* tab. This tab shows the silicone exterior which determines the movement behaviour of the actuator. With the controls, the thickness of each silicone exterior side can be adjusted. For example, an actuator that bends upwards would have a thick top side, like 0.5 cm and the opposite side, the bottom side, would be thin, like 0.1 cm. The actuator expands where the least resistance is, here the relative to the top, thin, bottom side.

Sometimes the issue of having an air channel block outside or partially outside of the silicone exterior would occur. This would result in an undesirable hole in the actuator. Also in this tab, the simulation can be initiated.

SoRoCAD pipes the required meshes to [SOFA](#)<sup>1</sup> and receives 20 meshes of the deformation of the actuator back. These are then loaded and displayed back in SoRoCAD. The user can skip through each deformed mesh and see how the actuator would behave when built. After going back and forth between the *Air Channel* and *Silicone Exterior* tab to revise and simulate the actuator design, the user can prepare a mold in the last tab, the *Mold* tab.

### 3.3 Framework

The goal of this third version of SoRoCAD is to be more accessible than its predecessors, which are limited to macOS. For this goal [Python](#)<sup>2</sup> is a suitable choice. It only requires an interpreter which is available on the most used desktop computer and laptop operating systems, Windows, macOS and Linux. Furthermore modules are easily installed with Python's package manager pip, which is included with every Python installation. There is also the option to use [PyInstaller](#)<sup>3</sup> to bundle SoRoCAD and its Python dependencies to a single packaged app that can be run without requiring the user to install a Python interpreter themselves. Although this work implements SoRoCAD as a Python standalone application, a future web application version could be made by using this Python implementation. The core package of SoRoCAD 3 is [Open3D](#)<sup>4</sup> by Zhou et al. [2018], a modern library for 3D data processing. This package is used to store, edit and export 3D data. Open3D also provides functions to build a custom scenes which includes camera controls, lighting and displaying the model. Additionally functions for Constructive Solid Geometry and user interface building are included.

---

<sup>1</sup> <https://www.sofa-framework.org/>

<sup>2</sup> <https://www.python.org/>

<sup>3</sup> <https://pyinstaller.org/>

<sup>4</sup> <https://www.open3d.org/>

Excursus:  
*Constructive Solid  
Geometry*

**CONSTRUCTIVE SOLID GEOMETRY:**

Constructive Solid Geometry (CSG) is a 3D modeling technique for surfaces and objects. Binary operations like union, intersection and subtraction are used on simpler objects to combine them into a more complex model.

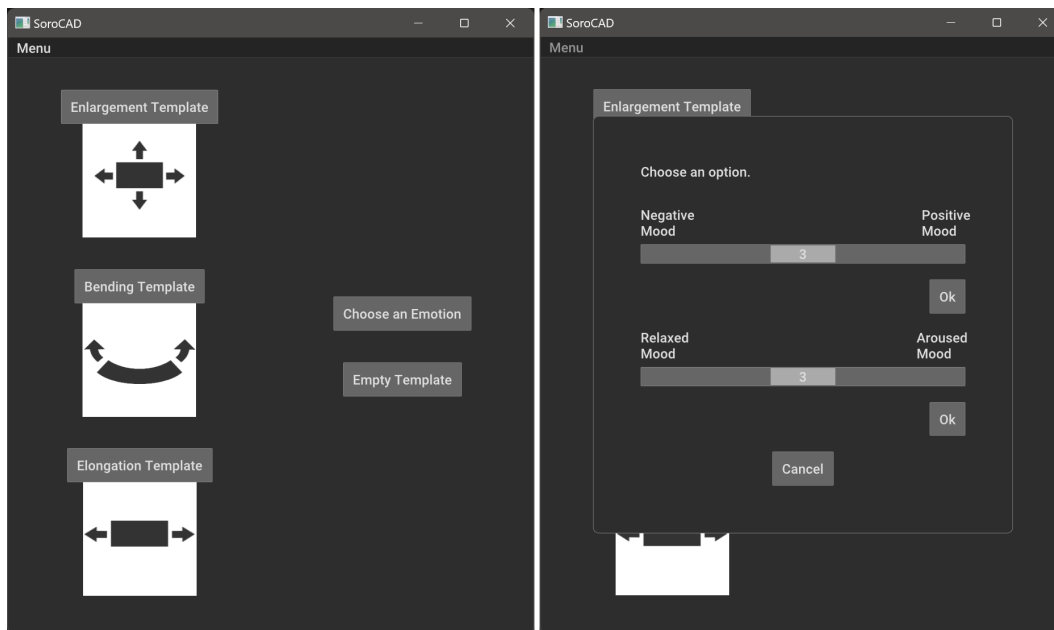
Desired mesh  
properties

Furthermore, a WebRTC function, which allows the user to run their Open3D-based code in a web browser. Unfortunately, this makes the application very unresponsive which leads to an unpleasant user experience. Another important aspect is that this package is still maintained.

The second package is [Gpytoolbox](https://gpytoolbox.org/0.2.0/)<sup>5</sup> by Sellán et al. [2023], which replaces the some CSG functions from Open3D, since some of the boolean functions in Open3D v0.18 are not working as intended. To be precise, after calling a boolean function on two meshes without holes, we sometimes get a mesh with holes. Each basic building block in SoRoCAD consists of a watertight air channel mesh and a watertight silicone exterior. A triangle mesh is watertight if and only if it is vertex manifold, edge manifold and not self intersecting. A triangle mesh is edge manifold if each edge is only bounding exact one or two other triangles. If for each vertex the edges that are connecting in this vertex are edge manifold and all edges are connected which means two or more edges can only be connected in a vertex, not by an edge, the mesh is called vertex manifold. A mesh is self intersecting if it has a triangle that intersects another mesh. Ideally, we want to maintain these properties after the boolean operations for CSG. As mentioned Open3D does not always maintain the edge manifold property, because the edges at a hole bound to no other edges. The dependency for SoRoCAD's simulation feature is SOFA, which is later introduced in Chapter 3.6 "Simulation Feature".

SoRoCAD's back-end structure is modular, which means switching to a different simulation tool or to a different library for CSG in future is not a difficult task.

<sup>5</sup> <https://gpytoolbox.org/0.2.0/>



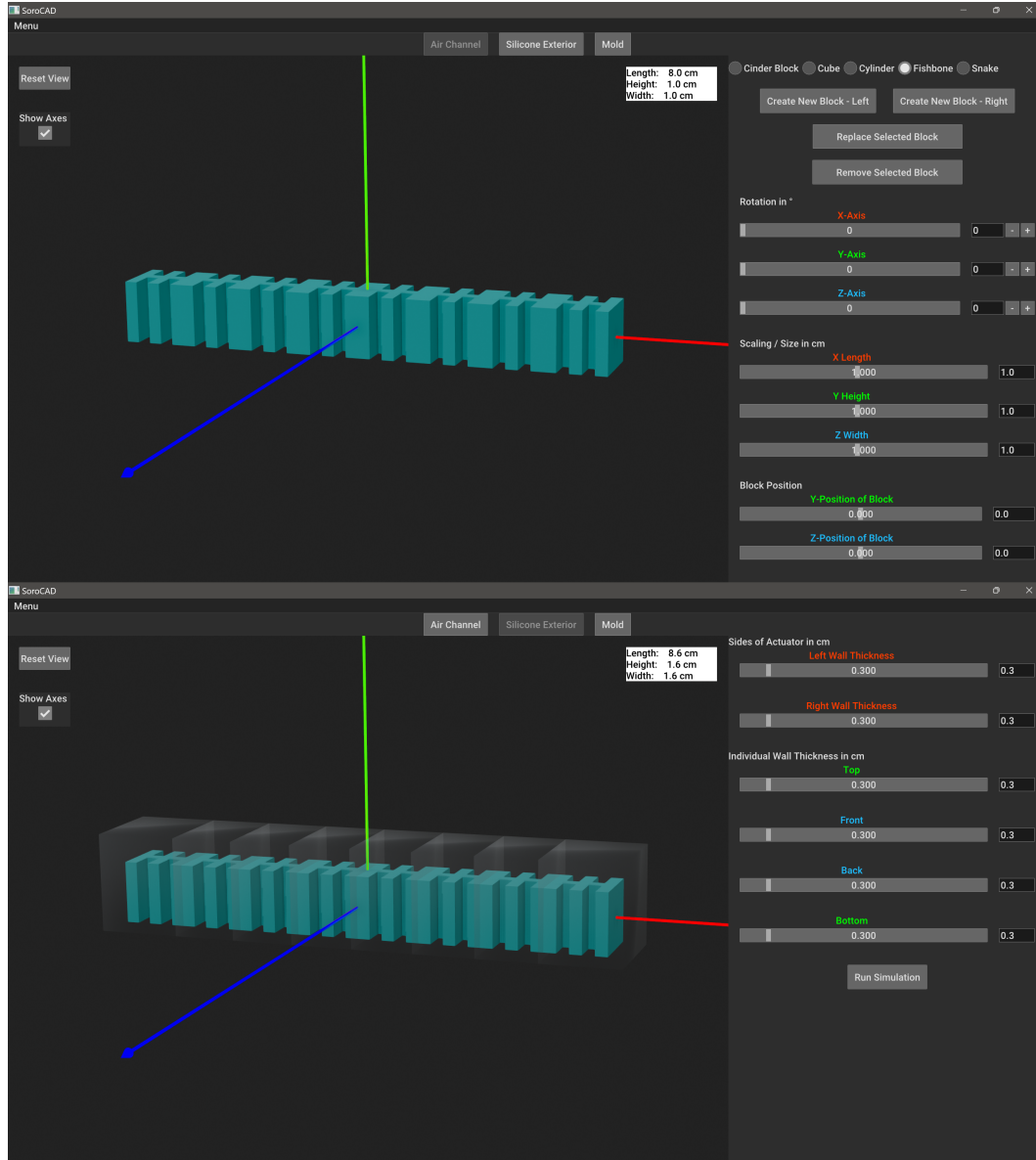
**Figure 3.1:** *Left:* The first window after starting SoRoCAD. *Right:* Pop up after selecting *Choose an emotion*. The user can choose a template based on one of the sliders.

### 3.4 Actuator Design

The general workflow, user interface and user interaction will mostly be maintained in SoRoCAD 3. The objective here is to use this opportunity of re-implementing SoRoCAD to fix the mentioned issues and make using this tool more convenient.

Firstly, the *Template Selection* tab not only provides templates for the before mentioned, three basic movements, but also enables the user to design actuators that convey specific emotions. Upon selecting the *Choose an Emotion* button, a window pops up with two different sliders. The first slider ranges from a negative emotion to a positive emotion, while the second slider ranges from a relaxed to a aroused mood. Each slider has five levels. The data for the emotion and their corresponding movements are provided by Schröder [2022]. The user interface for the template selection can be seen in Figure 3.1

New template selection  
Window



**Figure 3.2:** *Top:* Air channel tab, with fishbone air channel blocks. The controls on the right can be used to create and manipulate air channel blocks. *Bottom:* Silicone Exterior tab with an actuator provided by the bending template. The thickness of the silicone exterior can be set with sliders and number fields on the right. A simulation of the seen actuator can be initiated in this tab by clicking on the *Run Simulation* button.

The actuator design happens mostly in the *Air Channel* tab and *Silicone Exterior* tab, see Figure 3.2. The first few changes are related to user interaction, to be more precise the behaviour of the air channel blocks. While rotating and scaling multiple block was already possible in the previous version, where new blocks are added when multiple block are marked was not clear. For more consistency a new block is now created directly next to every marked block, depending on which side the user chooses. Next, the behaviour of the air channel blocks, when rotating is changed. Only a multiple of a 90° rotation in every axis lead to a connected air channel, because each block had its own coordinate frame which could not intersect with others. Having the air channel blocks intersect each other enables more design options for an air channel. All blocks are kept in a single coordinate frame and the distance between two neighboring blocks, now depends only on the X-scaling of these blocks. Nonetheless, gaps between air channel blocks can still occur. However, to prevent faulty actuator, a connectivity check is run when the user tries to switch to the next tab. If not all air channel blocks are not connected, the user is prompted to adjust their air channel design before continuing.

Addressing the issue in the air channel tab

Avoid disconnected air channels

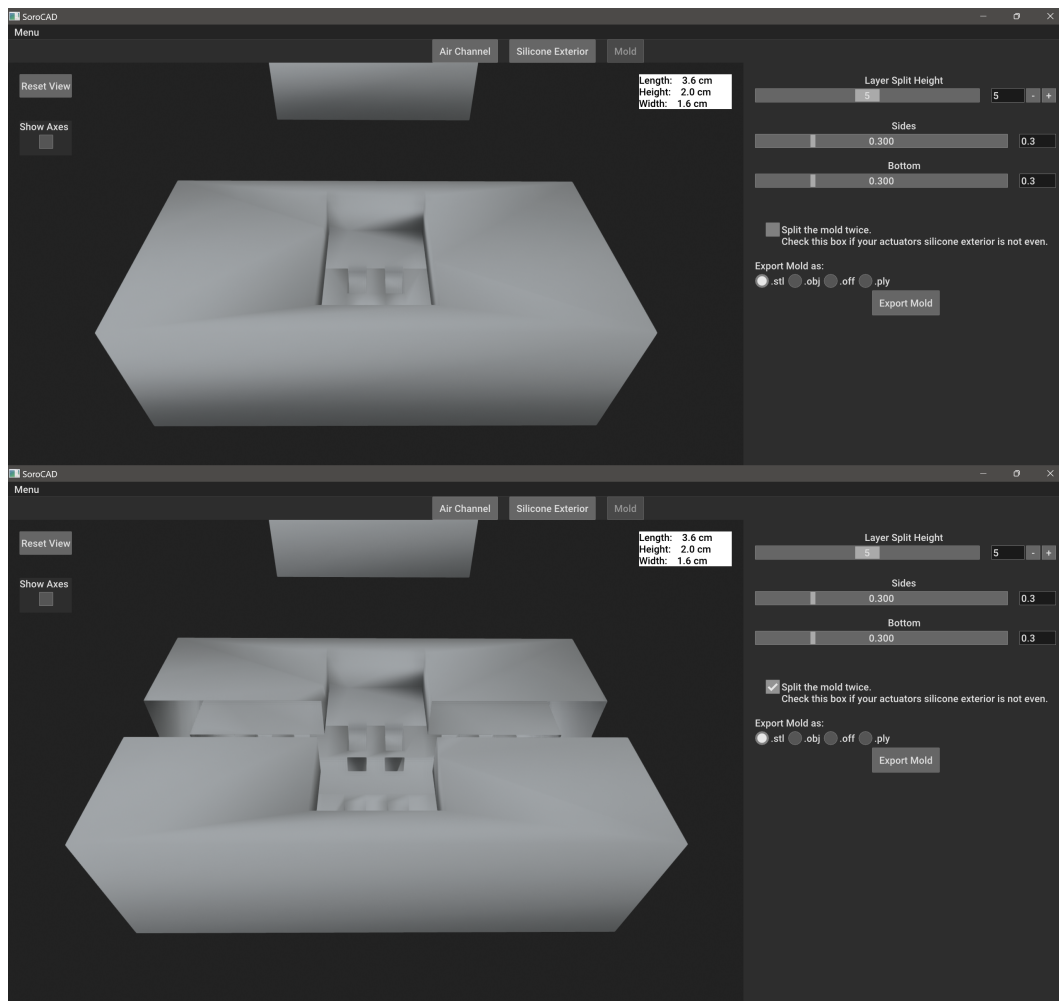
The next issue occurred in the second tab when setting the silicone exterior. To prevent having air channel blocks outside of the silicone exterior, adding the thickness of silicone exterior depends now on the axis aligned bounding box of the air channel block. This ensures an actuator without holes. The axis aligned bounding box is the smallest box that contains the air channel block, but the edges of the box are also parallel to the axis of the space.

Always have the air channel block inside the silicone exterior

### 3.5 Mold

The third and last tab in SoRoCAD is dedicated to creating a suitable mold of the actuator which then can be 3D-printed. While a floating air channel in a virtual environment is no issue, in reality supports are needed to print the air channel in the correct position. This would lead two issues. Firstly, the supports which hold the air channel in the correct po-

Molding method



**Figure 3.3:** *Top:* The mold of an actuator with not perfectly even silicone exterior. *Bottom:* The same mold, after additional splitting for easier demolding.

sition would each leave a hole in the silicone exterior. Secondly, the solid air channel has to be removed from the actuator through one of the holes left by the supports. So, we decided to keep using the method used in SoRoCAD 2, that is splitting the actuator in two parts and creating a mold for each part separately.

Mold generation  
process

The first step for creating a mold of a given actuator is splitting the actuator in the XZ-plane at a user chosen height. Then for each half of the actuator a cuboid which encompasses the entire actuator part, here top and bottom part,



is created. Boolean subtraction in Constructive Solid Geometry is then used to obtain a negative mesh, this means everything that was hollow is now solid, i.e. the air channel is now solid and the silicone exterior is represented by the hollow part of the mold. Casting silicone in these mold parts and letting it cure would leave us with one top and one bottom part of the actuator. The user now can glue these parts together with the same silicone used for casting the parts.

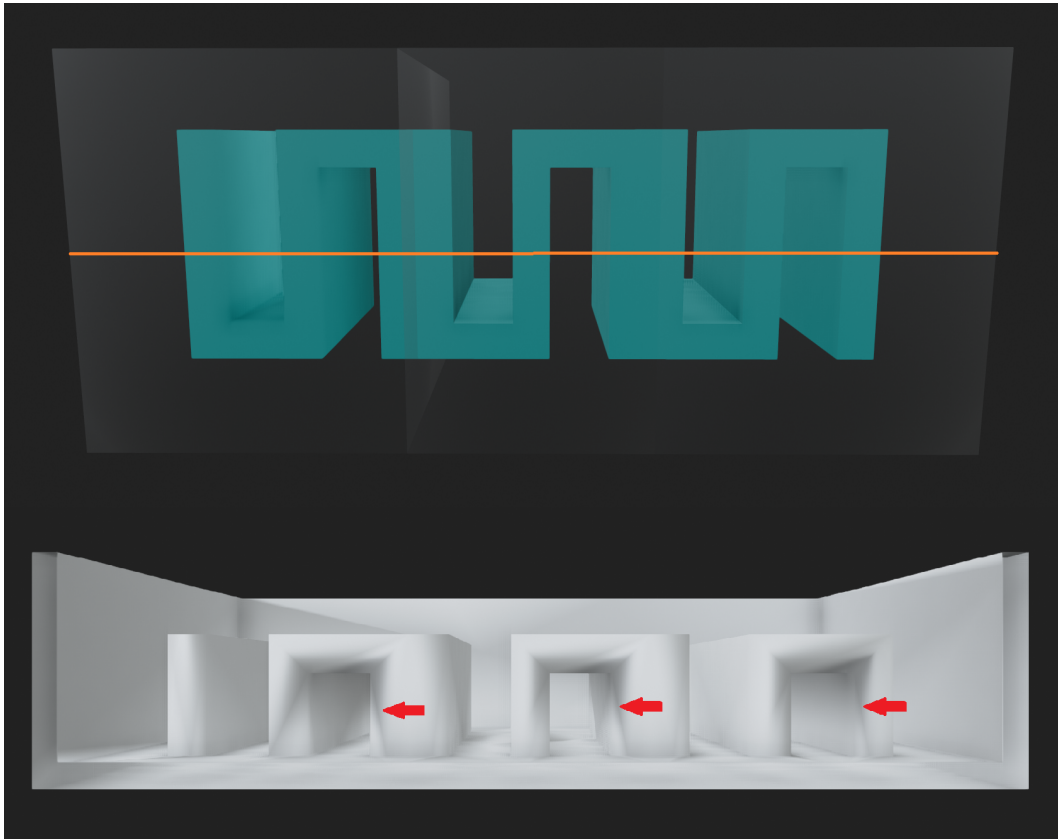
While this method works for actuators with an even silicone exterior at the top and bottom, demolding issues occur when the top or bottom walls are of different heights. A solution for these actuators would be splitting the top and bottom molds of the actuator an additional time along the XY-plane. Each of the top and bottom part of the mold consists of two parts which are used as single mold for each the top or bottom. After the silicone is cured, these quarter mold parts can be gently pulled apart. An example for such a mold can be seen in Figure 3.3.

Demolding actuator  
with uneven silicone  
exterior

With this method molds for actuator with irregular silicone exterior can be 3D-printed but actuators with complex air channels still cannot be demolded without damaging the mold or the actuator itself. We added this feature to ensure that fabrication of all actuators provided by our templates is possible. Generally, the more complex an air channel is, the more mold parts are needed. An example for an actuator that cannot be built with our mold technique is shown in Figure 3.4.

Lastly, SoRoCAD now provides a file manager. The user now can save the mold in separate files for the top and bot part which makes them more convenient to 3D-print. Also, the feature to choose a save location and name the files makes handling multiple molds easier.

Adding a file manager

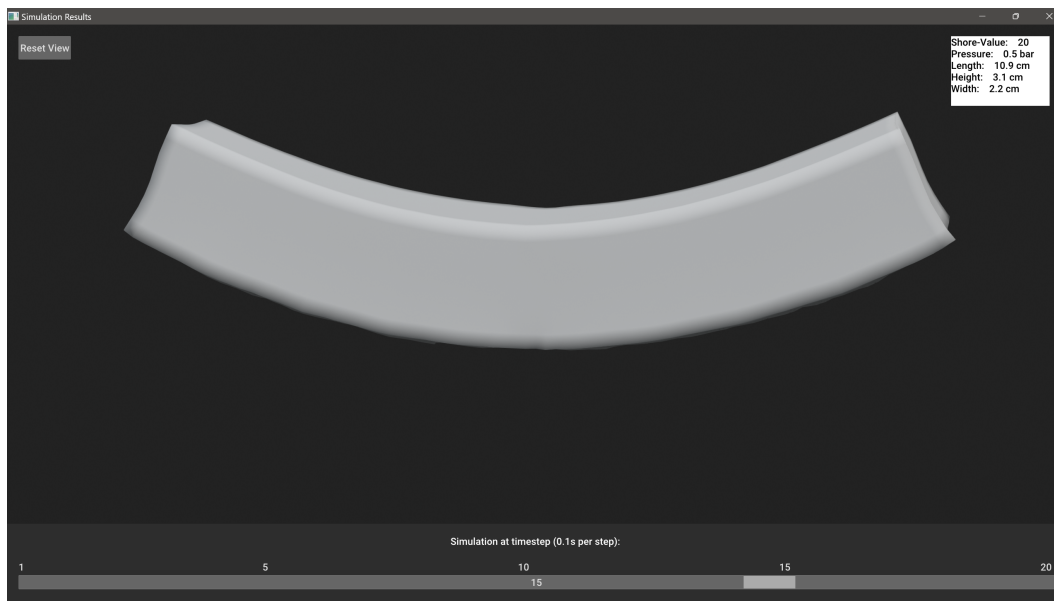


**Figure 3.4:** *Top:* An actuator with snake shaped air channel blocks. The orange line indicates the layer split height. *Bottom:* The corresponding mold for the upper half of the actuator. The red arrows indicate tunnels that make demolding impossible without damaging the actuator or mold.

### 3.6 Simulation Feature

SoRoCAD's simulation feature continues to rely on [SOFA](https://www.sofa-framework.org/)<sup>6</sup>. SOFA stands for Simulation Open Framework Architecture and is an efficient framework dedicated to research, prototyping and development of physics simulation [Faure et al., 2012]. Although the main application area was the medical field, the rise of interest in soft robotics made the developers include the Soft Robots plugin [Coevoet et al., 2017] in SOFA's main release. Additionally the SofaPython3 plugin is used to create a SOFA scene and simu-

<sup>6</sup> <https://www.sofa-framework.org/>



**Figure 3.5:** *Simulation Results:* In this window 20 models of the given actuator are shown. The user can use the slider at the bottom to switch between time steps.

late an actuator directly from a Python environment. For Windows and Linux operating systems an installer, and for macOS ready to use binaries are provided. Although for machines with macOS and a M1 processor those files did not work. For these machines the user has to build SOFA themself by compiling SOFA with CMake. A detailed instruction is provided on SoRoCAD's git page or [SoRoCAD's project page](#)<sup>7</sup>.

A finite element method (FEM) simulation of a given actuator requires two different meshes, a volumetric mesh of the finished, hollow actuator and a surface mesh of the air channel.

#### **SURFACE AND VOLUMETRIC MESHES:**

Surface meshes describe an object with a set of two dimensional shapes, for example triangles while volumetric meshes not only describe the surface and shape but also the inner volume of an object. The FEM simulation in SOFA uses a tetrahedral volume mesh.

Excursus:

*Surface and Volumetric Meshes*

<sup>7</sup> <https://hci.rwth-aachen.de/sorocad>

Simulating air pressure

Generating a volumetric mesh from a surface mesh

The latter mesh is provided by SoRoCAD itself since we mainly work with triangle surface meshes. The surface mesh of the air channel is used to simulate the air pressure by applying forces along each vertex normal of the air channel mesh. The force is then exerted on the volumetric mesh of the actuator and a deformation is visible. However, obtaining an volumetric mesh of the actuator is not trivial task. In order to create a tetrahedral volume mesh from a surface mesh, we use the Computational Geometry Algorithms Library, abbreviated CGAL. This library is available in C++, but also as SOFA plugin which connects the C++ library to SOFA. After installing the library with Homebrew on macOS or with [vcpkg](https://vcpkg.io/en/)<sup>8</sup> on Windows, the plugin can be installed by setting the corresponding variable in CMake when installing SOFA. An alternative way to use CGAL is pygalmesh by Schlömer, a python front-end to CGAL's 2D and 3D mesh generation capabilities. Unfortunately this package is no longer maintained and the installation with pip, Python's package installer, does not work on every system without modifying the setup file in python. The third option for mesh generation is Gmsh by Geuzaine, Christophe and Remacle, Jean-Francois, which has stricter requirements on the provided surface mesh than CGAL, i.e. a vertex manifold and edge manifold mesh.

Using boolean union with the Gpytoolbox on all air channel blocks and silicone exterior blocks to obtain a single air channel and a single silicone exterior can lead to surface meshes with overlapping triangle facets. These overlapping facets cannot be handled by Gmsh's 3D volume generation while CGAL's implementation can ignore these facets. Although the Gmsh's installation is more convenient with pip, it is not as robust as CGAL's mesh generation. This also means, if we get more reliable boolean functions for CSG, we can switch to using Gmsh. The reason Gpytoolbox is used although we get overlapping facets is that Open3D's own boolean functions for CSG sometimes return surface meshes with holes in it. These faulty surface meshes are not usable for any volumetric mesh generation. There are a few alternative Python packages with 3D mesh generation capabilities, for example [PyMesh](https://pypi.org/project/pymesh/)<sup>9</sup> which

<sup>8</sup> <https://vcpkg.io/en/>

<sup>9</sup> <https://pypi.org/project/pymesh/>

was last updated in 2015 and also requires CGAL for 3D mesh generation. These packages have a similar difficult installation process, but the last update was longer ago, so compatibility issues are added. We also tested [CGAL's Python bindings](#)<sup>10</sup> which are still in an experimental phase. Although we could obtain an volumetric mesh, the output mesh has to be converted into a file format that is accepted by SOFA. For larger meshes converting these files with Gmsh, sometimes lead to a loss of vertices or triangles. We also found that [TetGen](#)<sup>11</sup> uses the same mesh generation algorithm as Gmsh. However, TetGen also requires us to convert the output mesh, such that it can be used in SOFA which is a source of error.

	Installation	Dependencies	Last updated	Issue
<b>CGAL Plugin</b>	CMake	CGAL, Eigen	February 2024	Installation
<b>pygalmesh</b>	pip	CGAL, Eigen	September 2022	Installation
<b>Gmsh</b>	pip	-	May 2024	Unreliable
<b>PyMesh</b>	<a href="#">Docker</a> <sup>a</sup>	CGAL, Eigen	July 2020	Compatibility
<b>TetGen</b>	pip	-	February 2024*	Unreliable

**Table 3.6:** Overview over tested Python packages with volumetric mesh generation capabilities. \*Although the last update was in February 2024, the Python package relies on the C++ files from August 2020.

<sup>a</sup> <https://www.docker.com/>

Having an actuator with smaller and more tetrahedra, effectively a higher resolution, makes the simulation more accurate but at a computational cost Navarro et al. [2023]. The general size of the tetrahedra is specified in the CGAL plugin or pygalmesh in the backend of SoRoCAD. The values are adjusted such that each edge and all features of the air channel are included in the mesh. The most coarse volumetric mesh that can describe the inner volume of the actuator is obtained. The simulation runs with parameters for the actuator provided by the user, the air pressure in bar and the stiffness of the material of the actuator measured as Shor A value. Although SOFA uses young modulus to describe the stiffness of a material, SoRoCAD converts the user provided inputs before passing them over to SOFA.

Mesh resolution

<sup>10</sup> <https://github.com/CGAL/cgal-swig-bindings/wiki>

<sup>11</sup> <https://pypi.org/project/tetgen/>

Excursus:  
*Shor A hardness*

**SHOR A HARDNESS:**

The Shor A hardness scale describes the hardness of a material with a value between 0 and 100. The harder the material the higher is the Shor A value.

Piping simulation  
results to SoRoCAD

Then SOFA writes the deformed actuator for each time step as triangle mesh in an OBJ file. Afterwards these files are loaded and displayed in SoRoCAD. The user can use a slider to see how the actuator would behave with the provided parameters. Although any amount of steps can be displayed, 20 files are enough to show the entire movement of almost every actuator.

# Chapter 4

## Evaluation

This chapter evaluates SoRoCAD 3 by discussing the technical aspect, such as distribution and installation, the accuracy of the simulation and by presenting user feedback.

### 4.1 Technical Evaluation

#### 4.1.1 Distribution and Installation

All dependencies of SoRoCAD 3 can be divided into three parts. The first part which includes Python and all Python packages, namely Python 3.10, Open3D and Gpytoolbox. This part is very easily and quickly installed on any operating system by using an Python installer or using [Homebrew](#)<sup>1</sup> for macOS. The required packages are also easily installed by executing the requirements file which is also provided on the git page of this project. There is also the option to install these packages manually with Python's package installer pip by running two command lines in the terminal. Only this first part is required to design and fabricate an actuator in SoRoCAD. The more complicated parts are the second and third part which are the dependencies for creating a volumetric mesh and enabling the simulation fea-

---

<sup>1</sup> <https://brew.sh/>

ture, respectively. While the second part that includes pygalmesh and CGAL, requires a few steps to install, the third part, the SOFA Framework, is only difficult to install on macOS with M1 processors. Pygalmesh was last updated in September 2022 and the installer assumes the path for its dependencies, CGAL and Eigen, is the same on all systems. For Windows and Linux SOFA provides installer which work like any other installer for commonly used software. For macOS ready to use binaries are provided, which unfortunately do not work for processors with arm64 architecture. These systems requires the user to install SOFA manually by compiling SOFA with CMake themselves. Although this could change in future, since there is a growing interest in 3D physics simulation and the SOFA team tries to accommodate this interest.

#### 4.1.2 Simulation

Actuator shape in  
simulation does not  
match fabricated  
actuator

In the previous version, there were two parameters the user could set before starting a simulation, the stiffness of the silicone as Shor-A value and the pressure of the air pump in bar. The value for the stiffness ranges from 1 to 100 and the air pressure ranges from 0 bar to 2 bar. Sahabi [2021] tested the accuracy of SoRoCAD's simulation feature by fabricating several actuators and compared their behaviour with the simulation of the actuator in SoRoCAD. Sahabi [2021] found that silicone actuator with a Shor value over 70 had small holes in them and most actuators with a Shor value over 50 could not be demolded without damaging them. Especially actuators with fish bone shaped air channel blocks, demolding without irreparably damaging them was not possible. We now limit the range of the parameter values for the stiffness of the material to 1 to 50 and the air pressure to 0.1 bar to 1 bar since 0 bar would represent no actuation which is the default state of every actuator. However, the actuators movement in the simulation with the new ranges of the parameters is too weak compared to the actual movement of the built actuator. To compensate this, the user inputs are scaled with values for stiffness and air pressure, such that the simulation approximates the actual behaviour. Although the simulated ac-



tuators now bend and move to a similar extent compared to their fabricated version, the shape of the actuator itself does not match exactly. For example, the circumference of the fabricated bending actuator is much bigger than the circumference at the same part of the actuator in the simulation. The simulation only provides a prediction of the movement, not of the actuator's exact shape.

## 4.2 User Feedback

As second part of the evaluation, user feedback was collected in a rather informal study. The following section describes the setup, procedure and results. Also some information about the participants are presented.

### 4.2.1 Setup

SoRoCAD was provided on a MacBook 5 2012, with an i7 2.6 Ghz processor, 16GB DDR3 RAM and a NVIDIA GT650M graphics card. Additionally a computer mouse was provided. And in order to easier evaluate qualitative data the screen and sound was recorded. The study was conducted in a think aloud format.

### 4.2.2 Participants

The two participants were male and between the age of 20 and 28, with an average age of 24, and students in computer science and mathematics. Both of them had no prior knowledge of any CAD tools or soft robotics. Although they have used a machine with macOS before, they prefer using Windows. They also prefer dark mode for applications. This question was asked because we also had a light mode version of SoRoCAD's main window.

### 4.2.3 Procedure

Participants were given a consent form and a questionnaire with questions about themselves to fill out. Afterwards the procedure was explained and a short introduction to Soft Robotics was given. The introduction included a short explanation about what soft robots are, what actuators are made of, as well as the the most common actuation methods. After the short explanation, participants were shown two silicone finger actuator, one with an bending movement and one with an S-shape movement to give them an impression of what they are designing with SoRoCAD.

### 4.2.4 Tasks

Following the introduction, the first of the five task was presented. In this task two pictures of an bending actuator were shown. The first picture of the actuator in its rest state and the second one while actuated. Participants were now asked to build the actuator in SoRoCAD, while they are allowed to use everything they can see.

Similar to the first task, using the templates would also solve the second task. Here the participants are asked to design an actuator which conveys a specific emotion, that is a very negative emotion.

As a third task participants were asked to open an empty template and design an actuator that has a bending first half while the second half does not move when actuated.

For the fourth task SoRoCAD was not required. In this task participants were asked how they would design an actuator with a twisting motion. They were given pen and paper to describe how they would design the requested actuator.

For the last task, participants were given 15 minutes time to build whatever they like and test the limits of SoRoCAD. Afterwards a few question about their experience with SoRoCAD and their preferred platform for designing actuator or soft robots were asked. Finally, they were asked if they still had any feedback to share.

### 4.2.5 Results

The screen and voice recording were analyzed and the most interesting findings are presented in this section. We observed that most pop-ups with instructions, for example the controls for the 3D scene, were mostly ignored. After explaining the controls, participants tested them on a block with the sliders. For the first task all participants identified the displayed motion as bending motion and used the matching template. However, they first tried clicking on the picture instead of the button right above it. Making the picture clickable seems like a reasonable change. For this task, both participants used the number fields for a precise input.

The second task was also about using a template but this time for an actuator that conveys an emotion. While solving the task P1 asked "How can an actuator convey an emotion?" and after simulating the actuator design with the template P1 realized "Ok, this reminds me of a sad smiley - Makes sense." Using the simulation feature to see how the actuator behaves can validate their guess. It also was the first time for the participants navigating in a 3D scene. They also found the *Show Axes* and *Reset Camera* feature helpful, although P2 said that the buttons for these feature were not visible enough at first. Both features are located in the top left corner in the user interface, as seen in Figure 3.2.

In the third task, participants used their knowledge from the first task and build the bending half first. P2 created the not bending half by only increasing the wall thickness on the right of the actuator while P1 added blocks with a wall thickness on all sides such that these parts do not move when actuated. P1's solution for the fourth task was an actuator with four blocks and each block had one wall which was slightly thinner than the others. The thinner wall for the first block was the wall on the back of the first block, the top wall on the second block, the front wall on the third block and the front wall on the fourth block. P2 solution was to arrange the air channel blocks in a spiral pattern. In the fifth task, participants tried to implement their ideas from the previous task in SoRoCAD. P1 could produce an actuator that only slightly reminds of an twisting motion while P2 found the controls for shift-

ing an air channel block in Y and Z direction difficult to use. They also asked the interviewer how an actuator that shows a twisting shape would look like. The proposed solution would look like the twisting actuator described by Schröder [2022] or Figure B.1.

After the fifth task, the interviews started. Both participants described their experience as "fun". When asked if designing an actuator with SoRoCAD was comfortable with mouse and keyboard they said that there were no issues, but they definitely would not want to use the laptop touch pad. They also said that they could imagine using SoRoCAD on a tablet but would avoid doing so on a smartphone. The reason for this is the screen size. "The screen should be at least this size.", said P1 while pointing on the 13-inch MacBook Pro. Also, the general workflow of SoRoCAD was not clear at first and P2 suggested a video tutorial which explains SoRoCAD's workflow.

### 4.3 Discussion

The results and findings of the technical evaluation and the user interviews will be discussed in this section. Although, SoRoCAD is now accessible on the most common operating systems, the installation remains difficult due to the dependencies for the simulation feature. As mentioned in Chapter 3.3 "Framework", we could bundle SoRoCAD's Python part with PyInstaller to an executable application. However, for each operating system SoRoCAD has to be bundled separately, e.g. PyInstaller has to bundle SoRoCAD on a windows system for a packaged app for windows. Although there are other 3D physics simulation software which can simulate soft objects, SOFA is still the most suitable alternative since it can export the deformed mesh as .obj-file which we can easily pipe into SoRoCAD. Alternatives like [Panda3D](https://docs.panda3d.org/1.10/python/index)<sup>2</sup>, are also often used, but lack the ability to export results.

Boolean functions and  
volumetric mesh  
generation

The issue with creating a volumetric is that the input meshes are not watertight after the boolean function. We

<sup>2</sup> <https://docs.panda3d.org/1.10/python/index>

either need better boolean function, i.e. boolean functions that can retain the watertight property of our building blocks, or a volumetric mesh generation algorithm that is equally reliable as CGAL's algorithm. Unfortunately, most Python packages we tested relied on CGAL or TetGen. We also tested different libraries for boolean functions, for example Gmsh's boolean function, but most of the libraries require the geometries, i.e. vertices and triangles, to be created in their own framework. There is also the option to have SoRoCAD run as web application. For laptop and desktop computer user this would be a convenient way to use SoRoCAD but we have not found a way to use SoRoCAD including the simulation feature in a web browser. Also, the lack of responsiveness makes SoRoCAD not usable as a learning tool in a web browser. Although, this would make SoRoCAD accessible from a tablet or smartphone, the question if there exists an interest for designing or learning soft robots on such devices, remains. Both users in the interview reported that they were hesitant to use SoRoCAD on devices with smaller screens, but only having two users does not provide us with representative data. Following a participants request to show a short introduction video, would make SoRoCAD's workflow more clear, especially for users who have no experience with soft pneumatic actuators.

SoRoCAD as web  
application

## 4.4 Limitations

SoRoCAD in its current version is only capable of designing soft pneumatic finger actuators with a silicone exterior that is combined of cuboids. Also the simulation can only simulate positive air pressure, i.e. a simulation of vacuuming air out is currently not implemented. This could be used for the template of the mood conveying actuator when selecting level 3 on the negative-positive mood slider which describes a twisting motion. Also, the only actuation methods we can simulate are pressure based, e.g. pneumatic and hydraulic. Although pneumatic actuation can be realized with a simple syringe, it is the only actuation method that SoRoCAD teaches about. Pneumatic actuation is only a part of the various actuation methods which include pneu-

matic artificial muscles (PAM), shape memory alloy (SMA) or electro-active polymers (EAP) Li et al. [2023].

## Chapter 5

# Summary and Future Work

In this last chapter, the work and results of this thesis are summarized. Furthermore, ideas for future work to improve SoRoCAD are presented.

### 5.1 Summary and Contributions

This thesis introduced SoRoCAD and described how issues of the previous version were addressed. We improved the overall design process of actuator and provided templates for emotion conveying actuators, enabling users to explore shape change design. With these new templates, the mold generation has to be updated, such that the user can fabricate all these newly added models. Exporting and handling multiple files is now more convenient, thanks to the file manager. We also updated and adjusted the SoRoCAD's simulation feature. Additionally, the modular back-end makes SoRoCAD easy to maintain and used libraries or dependencies can be switched in the future if more suitable alternatives are found. We also evaluated the technical aspect of SoRoCAD regarding the installation and simulation accuracy, and collected user feedback for future improvements.

## 5.2 Future Work

As the results of Chapter 4 “Evaluation” point out, there are still some improvements to be made in SoRoCAD. For example, the user interface can still be improved by mapping control elements in a natural way, e.g. the slider for shifting an air channel block up and down can be arranged vertically or tick marks can be added on sliders. Although, SoRoCAD detects disconnected air channel blocks and asks the user to correct the air channel, the blocks in question are not highlighted. For accessibility, adapting SoRoCAD for a web browser could work, so it would also run on mobile devices. Before doing so, a survey to check if there exists a reasonable interest in designing pneumatic actuators on mobile devices, could be conducted. The highest priority would still be replacing the 3D mesh generation or the libraries for CSG such that the distribution of the current version is easier. Further extensions in SoRoCAD could include combining multiple finger actuators to produce a gripper, adding more basic air channel blocks or provide more options to design the silicone exterior. Additional templates for emotion or mood conveying actuators could be included after this field is more explored.



## Appendix A

# Study Materials

## Informed Consent Form

### Evaluating a tool for designing and simulating Soft Pneumatic Actuators

**Principal investigator:** Philip Le  
Email: [philip.Le@rwth-aachen.de](mailto:philip.Le@rwth-aachen.de)

**Purpose:** The goal of this study is to evaluate the current state of a tool for designing and simulating Soft Pneumatic Actuators. Therefore, participants will be asked to use the tool and answer questions about their experience while doing so.

**Procedure:** Firstly, participants are asked to provide information about themselves by filling out a questionnaire. After a brief introduction about Soft Pneumatic Actuators, participants are tasked to design different actuators in SoRoCAD, the tool which is tested.

The experimenter will capture the screen and record the audio of the entire session. All information will be confidential. (See 'Confidentiality' below for details.)

**Risks/Discomfort:** Even though the study is expected to last no longer than one hour, you may become fatigued during the course of your participation in the study. Feel free to take as many breaks as necessary during the study. There are no risks associated with participation in the study. Should completion of the task becomes distressing to you, it will be terminated immediately.

**Confidentiality:** All information collected during the study period will be kept strictly confidential. You will be identified only through identification numbers and background information you divulge in publications or reports. If you agree to join this study, please sign your name below.

**Addendums:** Participation in this study is voluntary. You are free to withdraw or discontinue the participation. Participation in this study will involve no cost to you.

I have read and understood the information on this form.

I have had the information on this form explained to me.

_____	_____	_____
Participant's Name	Participant's Signature	Date
	_____	_____
	Principal Investigator	Date

**Figure A.1:** The Informed Consent Form study participants were are asked to fill out.

---

## Questionnaire

1. How old are you?

---

2. What is your profession?

---

3. Which operating system do you prefer?

---

4. Do you have experience with CAD (Computer Aided Design)?

---

5. If you have experience with CAD tools, which tools have you used before?  
And how often do you use these tools?

---

6. How much experience do you have with Soft Robotics?

Very  
little

A lot

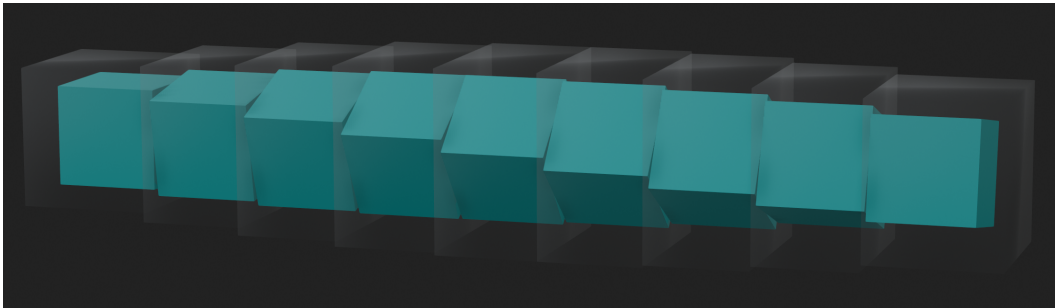
7. If you have experience with Soft Robotics, please describe them.

---

**Figure A.2:** The questionnaire study participants were asked to fill out.



**Figure A.3:** These images were shown for the first task in the study. *Top:* An actuator (8 cm x 1.5 cm x 1.5 cm) before actuation. *Bottom:* The same actuator after pumping air inside.



**Figure B.1:** The actuator that creates a twisting motion after vacuuming air out of the actuator.

## Appendix B

# SoRoCAD Files and Resources

[SoRoCAD's Project Page](https://hci.rwth-aachen.de/sorocad)<sup>1</sup>

[SOFA Framework's Homepage](https://www.sofa-framework.org/)<sup>2</sup>

---

<sup>1</sup> <https://hci.rwth-aachen.de/sorocad>

<sup>2</sup> <https://www.sofa-framework.org/>



## Bibliography

- [1] Kaitlyn Becker, Clark Teeple, Nicholas Charles, Yeonsu Jung, Daniel Baum, James C. Weaver, L. Mahadevan, and Robert Wood. Active entanglement enables stochastic, topological grasping. *Proceedings of the National Academy of Sciences*, 119(42):e2209819119, 2022. doi.org/10.1073/pnas.2209819119.
- [2] Anke Brocker, Jakob Strüver, Simon Voelker, and Jan Borchers. SoRoCAD: A Design Tool for the Building Blocks of Pneumatic Soft Robotics. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts, CHI EA '22*, New York, NY, USA, April 2022. Association for Computing Machinery. doi.org/10.1145/3491101.3519770.
- [3] Eulalie Coevoet, Thor Morales Bieze, Frederick Largilliere, Zhongkai Zhang, Maxime Thieffry, Mario Sanz Lopez, Bruno Carrez, Damien Marchal, Olivier Goury, Jeremie Dequidt, and Christian Duriez. Software toolkit for modeling, simulation and control of soft robots. *Advanced Robotics*, 31:1208–1224, November 2017. doi.org/10.1080/01691864.2017.1395362.
- [4] Zhitong Cui, Shuhong Wang, Junxian Li, Shijian Luo, and Alexandra Ion. MiuraKit: A Modular Hands-On Construction Kit For Pneumatic Shape-Changing And Robotic Interfaces. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference, DIS '23*, page 2066–2078, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3563657.3596108.
- [5] François Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, and Stéphane Cotin. SOFA: A Multi-Model Framework for Interactive Physical Simulation. In Yohan Payan, editor, *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, volume 11 of *Studies in Mechanobiology, Tissue Engineering and Biomaterials*, pages 283–321. Springer, June 2012. doi.org/10.1007/8415\_2012\_125.
- [6] Geuzaine, Christophe and Remacle, Jean-Francois. Gmsh. URL <http://http://gmsh.info/>.

- 
- [7] Kristian Gohlke, Wolfgang Sattler, and Eva Hornecker. AirPinch – An Inflatable Touch Fader with Pneumatic Tactile Feedback. In *Proceedings of the Sixteenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '22, New York, NY, USA, 2022. Association for Computing Machinery. doi.org/10.1145/3490149.3505568.
- [8] Jonathan Hiller and Hod Lipson. Dynamic Simulation of Soft Multimaterial 3D-Printed Objects. *Soft Robotics*, 1(1):88–101, 2014. doi.org/10.1089/soro.2013.0010.
- [9] Hyunyoung Kim, Aluna Everitt, Carlos Tejada, Mengyu Zhong, and Daniel Ashbrook. MorpheesPlug: A Toolkit for Prototyping Shape-Changing Interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. doi.org/10.1145/3411764.3445786.
- [10] Christopher Kopic and Kristian Gohlke. InflatiBits: A Modular Soft Robotic Construction Kit for Children. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '16, page 723–728, New York, NY, USA, 2016. Association for Computing Machinery. doi.org/10.1145/2839462.2872962.
- [11] Weidong Li, Diangang Hu, and Lei Yang. Actuation Mechanisms and Applications for Soft Robots: A Comprehensive Review. *Applied Sciences*, 13(16), 2023. doi.org/10.3390/app13169255.
- [12] Hedieh Moradi and César Torres. Siloseam: A Morphogenetic Workflow for the Design and Fabrication of Inflatable Silicone Bladders. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, DIS '20, page 1995–2006, New York, NY, USA, 2020. Association for Computing Machinery. doi.org/10.1145/3357236.3395473.
- [13] Stefan Escaida Navarro, Tanguy Navez, Olivier Goury, Luis Molina, and Christian Duriez. An Open Source Design Optimization Toolbox Evaluated on a Soft Finger, 2023.
- [14] Sarah Sahabi. SoRoCAD 2.0: Extending a CAD Tool for Soft Robotics. Bachelor's thesis, RWTH Aachen University, Aachen, April 2021.
- [15] Nico Schlömer. pygalmesh: Python interface for CGAL's meshing tools. <https://github.com/nschloe/pygalmesh>.
- [16] Sören Schröder. Conveying Emotions through Movements and Materials of Soft Robots in the Context of Smart Jewelry. Master's thesis, RWTH Aachen University, Aachen, January 2022.



- 
- [17] Silvia Sellán, Oded Stein, and others. *gpytoolbox: A Python Geometry Processing Toolbox*, 2023. <https://gpytoolbox.org/>.
- [18] Jakob Strüver. *Evaluating SoRoCAD: Enabling Users to Design Custom Soft Robotics*. Master's thesis, RWTH Aachen University, Aachen, October 2020.
- [19] Kirill Timchenko. *SoRoCAD: A Tool to Design Shape Changes in Soft Robotics*. Bachelor's thesis, RWTH Aachen University, Aachen, October 2019.
- [20] Matheus S. Xavier, Charbel D. Tawk, Ali Zolfagharian, Joshua Pinskier, David Howard, Taylor Young, Jiewen Lai, Simon M. Harrison, Yuen K. Yong, Mahdi Bodaghi, and Andrew J. Fleming. *Soft Pneumatic Actuators: A Review of Design, Fabrication, Modeling, Sensing, Control and Applications*. *IEEE Access*, 10:59442–59485, 2022. [doi.org/10.1109/ACCESS.2022.3179589](https://doi.org/10.1109/ACCESS.2022.3179589).
- [21] Wenchuan Zhao, Yu Zhang, and Ning Wang. *Soft Robotics: Research, Challenges, and Prospects*. *Journal of Robotics and Mechatronics*, 33(1):45–68, 2021. [doi.org/10.20965/jrm.2021.p0045](https://doi.org/10.20965/jrm.2021.p0045).
- [22] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. *Open3D: A Modern Library for 3D Data Processing*. *arXiv:1801.09847*, 2018.



---

# Index

Actuator .....	2
Air channel tab .....	17
AirPinch .....	3
Constructive Solid Geometry .....	14
Evaluation .....	25–32
Future work .....	34
Gmsh .....	22
Inflatibits .....	7
MiuraKit .....	8
Mold .....	17
MorpheesPlug .....	7
Shor A hardness scale .....	24
Siloseam .....	7
Simulation feature .....	20–24
Soft robots applications .....	3–4
SoRoCAD's framework .....	13
Surface Mesh .....	21
Template Selection .....	15
Volumetric mesh .....	21
VoxCad .....	9



