

# Ergonomics-Driven Computational Design of Furniture and Indoor Layouts

DISSERTATION

zur Erlangung des akademischen Grades

**Doktor der Technischen Wissenschaften**

eingereicht von

**Dipl.-Ing. Kurt Leimer, BSc**

Matrikelnummer 00825842

an der Fakultät für Informatik  
der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer  
Zweitbetreuung: Associate Prof. Dipl.-Mediensys.wiss. Dr.techn. Przemyslaw Musialski

Diese Dissertation haben begutachtet:

---

Peter Wonka

---

Bedrich Benes

Wien, 10. April 2023

---

Kurt Leimer





# Ergonomics-Driven Computational Design of Furniture and Indoor Layouts

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

**Doktor der Technischen Wissenschaften**

by

**Dipl.-Ing. Kurt Leimer, BSc**

Registration Number 00825842

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer

Second advisor: Associate Prof. Dipl.-Mediensys.wiss. Dr.techn. Przemyslaw Musialski

The dissertation has been reviewed by:

---

Peter Wonka

---

Bedrich Benes

Vienna, 10<sup>th</sup> April, 2023

---

Kurt Leimer



# Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Kurt Leimer, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 10. April 2023

---

Kurt Leimer



# Danksagung

Ich danke meinem Betreuer Przemyslaw Musialski für seine Leitung während meines gesamten Doktorats; meinen Kollegen Stefan Pillwein, Michael Birsak, Christian Hafner und Bernhard Steiner mit denen ich viele gemeinsame Tage im Büro verbracht habe; meinen anderen Mitwirkenden Florian Rist, Paul Guerrero, Thomer Weiss, Lukas Gersthofer und Andreas Winkler für ihre bedeutsamen Beiträge; und den Mitgliedern des Center for Geometry and Computational Design, da sie für mich die Möglichkeit geschaffen haben, ein Doktorat zu beginnen. Aber vor allem möchte ich mich bei meiner Familie für ihre großartige Unterstützung während meines Studiums bedanken, denn ohne sie hätte ich es nicht so weit geschafft.





# Acknowledgements

I want to thank my advisor Przemyslaw Musialski for his guidance throughout the entire duration of my doctorate; my colleagues Stefan Pillwein, Michael Birsak, Christian Hafner and Bernhard Steiner with whom I spent many days in the office; my other collaborators Florian Rist, Paul Guerrero, Thomer Weiss, Lukas Gersthofer and Andreas Winkler for their meaningful contributions; and the members of the Center for Geometry and Computational Design for creating the opportunity for me to start a doctorate. But most of all I want to thank my family for their tremendous support during my studies because I would not have made it this far without them.



# Kurzfassung

Menschen verbringen einen großen Teil ihres Lebens in Innenräumen, sei es zu Hause, am Arbeitsplatz, oder in öffentlichen Räumlichkeiten wie Restaurants oder Museen, weswegen das Entwerfen von solchen Innenräumen eine besonders wichtige Aufgabe ist. Es gibt dabei viele verschiedene Anforderungen zu erfüllen, sowohl beim Designen von einzelnen Möbelstücken, als auch bei deren Anordnung innerhalb eines gegebenen Raums. Dies macht Expertenwissen zu einer absoluten Notwendigkeit um die erwünschten Ziele hinsichtlich funktionaler, ästhetischer und ergonomischer Qualität zu erfüllen. Computer-Aided Design Software kann zwar dem Benutzer bei dieser Aufgabe behilflich sein, aber existierende Anwendungen dienen oft nur einem spezifischen Aspekt des Design-Prozesses. Besonders die Frage, ob ein bestimmtes Design ergonomischen Richtlinien entspricht, wird oft dem Designer überlassen, was dies besonders für Benutzer mit wenig Erfahrung zu einer schwierigen Aufgabe macht.

In dieser Dissertation erforschen wir verschiedene Ansätze, wie ergonomische Aspekte der Raumgestaltung in interaktive, automatisierte oder datengestützte Methoden für das Designen von Sitzmöbeln und Innenräumen integriert werden können. Im ersten Teil der Dissertation wird eine interaktive Methode zum Designen von Sitzmöbeln präsentiert. Mit einem Dreiecksnetz eines menschlichen Körpers als Eingabe berechnen wir zunächst eine approximative Druckverteilung am menschlichen Körper um zu lernen, wo er am meisten gestützt werden muss. Ein gegebenes initiales Design für ein Sitzmöbelstück wird dann so deformiert, dass es den Körper optimal unterstützt und gleichzeitig das ästhetische Design so gut wie möglich beibehält. Das Design kann dann außerdem interaktiv bearbeitet werden. Wir zeigen, dass dieser Ansatz es sogar unerfahrenen Nutzern erlaubt, in kurzer Zeit Designs für komfortable Sitzmöbel zu kreieren.

Während ein solcher interaktiver Ansatz dem Benutzer mehr Kontrolle überlässt, erfordert er gleichzeitig erheblichen manuellen Aufwand, wodurch er für Benutzer mit wenig Erfahrung schwierig zu verwenden ist. Im zweiten Teil dieser Dissertation wird daher eine Methode zur automatisierten Erzeugung von Sitzmöbel-Designs vorgestellt, deren Resultat lediglich von der Ziel-Pose abhängt. Die Designs, die mittels dieser Methode generiert werden, können darüber hinaus als initiales Design für die Methode aus dem ersten Teil der Dissertation verwendet werden. Unser Ansatz für die Berechnung der Druckverteilung wird außerdem erweitert, indem wir das Gewicht individueller Körperteile in Betracht ziehen und die Reibungskräfte approximieren, die auf den Körper einwirken,

um eine höhere Genauigkeit zu erzielen. Unsere Ergebnisse zeigen, dass das Nutzen der präsentierten Methode in Zusammenspiel mit unserem interaktiven Design Tool die Designs der Sitzmöbel noch weiter verbessern kann.

Da die ergonomische Qualität von Sitzmöbeln nicht nur von dem Design der Möbel selbst abhängt, sondern auch deren Relation zu anderen Elementen der umgebenden Räumlichkeit, stellen wir im dritten Teil der Dissertation einen datengestützten Ansatz für das Designen von Innenräumen vor. Da die gebräuchlichen Richtlinien für Raumgestaltung so zahlreich sind, ist es nicht machbar, beim Entwickeln einer Computational Design Methode alle davon explizit zu definieren. Doch durch Verwendung eines datengestützten Ansatzes ist es möglich, diese Regeln implizit zu lernen, solange eine große Anzahl an geeigneten Beispielen verfügbar ist. Doch gute Beispieldaten sind oft nicht in ausreichender Menge vorhanden. Bestehende Datensätze mögen einigen der erwünschten Regeln entsprechen, jedoch in anderen Bereichen unzureichend sein. Deshalb präsentieren wir eine Deep-Learning Methode, die sich die Vorteile von datengestütztem Lernen zu Nutzen macht, und gleichzeitig Makel im Datensatz korrigiert, mithilfe einer kleinen Menge an explizit definierten Regeln basierend auf Literatur über Ergonomie. Wir evaluieren die Designs, die mit dieser Methode synthetisiert wurden, mithilfe einer perzeptuellen Studie und zeigen, dass die Ergebnisse als gleich oder mehr realistisch betrachtet werden, als die Beispiel-Designs die zum Trainieren des Deep Learning Modells verwendet wurden.

# Abstract

Humans spend a large proportion of their lives indoors, be it at home, at their workplace, or in public facilities like restaurants or museums, making the design of such indoor spaces an important task. There are many different requirements for both the design of individual pieces of furniture and their arrangement within a given space, making extensive expert knowledge in this field a necessity to achieve the desired design goals in terms of functional, aesthetic and ergonomic quality. While Computer-Aided Design software can aid the user in this task, existing tools often focus on just one specific aspect of the design process. In particular, making sure that a given design adheres to ergonomic guidelines is often left entirely to the designer, making it a difficult task especially for novice users.

In this thesis, we explore different approaches of how ergonomic aspects of interior design can be integrated into interactive, automated or data-driven methods for the design of seating furniture and indoor layouts. The first part of the thesis presents an interactive method for the design of seating furniture. Given a triangular mesh of a human body in a specific pose as input, we first compute an approximate pressure distribution on the human body to learn where it needs the most support. A given initial design for a piece of seating furniture is then deformed to provide optimal support to the body while still conforming to the aesthetic design as much as possible. The design can furthermore be modified interactively. We demonstrate that this approach allows even novice users to create comfortable seating furniture designs in a short time span.

While an interactive approach provides more control to the user, it also requires substantial manual design effort, making it difficult to use for novice designers. The second part of this thesis proposes a method for the automated generation of seating furniture which depends entirely on the target pose. The designs created with this approach can be used as-is or as an initial design for the approach introduced in the first part of the thesis. Our method for computing the pressure distribution is furthermore extended by taking into account the weight of individual limbs and estimating the frictional forces acting on the body for increased accuracy. Our results show that the proposed method can further improve the designs of seating furniture when used in conjunction with our interactive design tool.

Since the ergonomic qualities of seating furniture do not only depend on the design of the furniture itself, but also its relation to the other elements of the surrounding indoor

space, the third part of the thesis introduces a data-driven approach for the design of indoor layouts. Since common guidelines for interior design are so numerous, it is an infeasible task to define all of them explicitly when developing a computational design method. Using a data-driven approach makes it possible to implicitly learn about these rules given a large set of suitable examples. But good example data is not always readily available in large quantities; existing datasets may fulfill some of the desired rules, but lack in other areas. We therefore propose a deep-learning approach that makes use of the advantages of data-driven learning, while at the same time correcting flaws in the dataset using a small set of explicitly designed rules drawn from ergonomics literature. We evaluate the designs synthesized by this approach using a perceptual study and show that the results are seen as equally or even more realistic than the ground truth designs that were used to train the deep learning model.

# Contents

<b>Kurzfassung</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Contents</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Research Goals . . . . .	3
1.3 Thesis Overview . . . . .	4
1.4 Authorship Statement . . . . .	5
<b>2 Related Work</b>	<b>7</b>
2.1 Ergonomics . . . . .	7
2.2 Furniture Design . . . . .	9
2.3 Layout Synthesis . . . . .	11
<b>3 Interactive Design of Seating Furniture</b>	<b>15</b>
3.1 Introduction . . . . .	15
3.2 Overview . . . . .	17
3.3 Pressure Field Computation . . . . .	17
3.4 Surface Fitting . . . . .	20
3.5 Pose Relaxation . . . . .	21
3.6 Design Process and Results . . . . .	23
3.7 Implementation and Fabrication . . . . .	26
3.8 Discussion and Conclusion . . . . .	27
<b>4 Automated Design of Seating Furniture</b>	<b>33</b>
4.1 Introduction . . . . .	33
4.2 Computational Model of Sitting . . . . .	35
4.3 Furniture Synthesis . . . . .	44
4.4 Results . . . . .	51
4.5 Discussion and Conclusions . . . . .	52
	xv

<b>5</b>	<b>Data-Driven Design of Indoor Layouts</b>	<b>57</b>
5.1	Introduction . . . . .	57
5.2	Ergonomic Costs . . . . .	59
5.3	Layout Generation with Expert Knowledge . . . . .	65
5.4	Results and Evaluation . . . . .	71
5.5	Discussion . . . . .	77
<b>6</b>	<b>Conclusion</b>	<b>81</b>
6.1	Summary . . . . .	81
6.2	Outlook and Future Work . . . . .	82
<b>A</b>	<b>Appendix</b>	<b>87</b>
A.1	Surface Fitting Details . . . . .	87
A.2	Jacobians for Inverse Kinematics . . . . .	88
A.3	Reaction Force Computation . . . . .	89
A.4	Surface Optimization Gradient . . . . .	90
	<b>List of Figures</b>	<b>93</b>
	<b>List of Tables</b>	<b>99</b>
	<b>List of Algorithms</b>	<b>101</b>
	<b>Bibliography</b>	<b>103</b>



CHAPTER 1 

# Introduction

Humans spend a large proportion of their lives indoors, be it at home, at their workplace, or in public facilities like restaurants or museums. The design of such indoor spaces is an important task; since they each serve a variety of purposes, there are different requirements for both the design of individual pieces of furniture and the arrangement of room elements in relation to each other and the floor plan of the room itself.

As a result, the task of designing indoor spaces is the subject of research in many disciplines. Interior design and architecture focus on the functional and aesthetic aspects of furniture and room layouts, the discipline of ergonomics is concerned with the well-being of humans who spend their time in these spaces, and the development of tools such as manufacturing devices and Computer-Aided Design (CAD) software requires expert knowledge in different fields of engineering.

In this thesis, we explore different approaches of how ergonomic aspects of interior design can be integrated into interactive, automated or data-driven methods for the design of seating furniture and indoor layouts.

## 1.1 Motivation

As mentioned above, the design of furniture and indoor spaces requires expert knowledge from a variety of disciplines, including architecture, ergonomics and engineering. Typically, a design will go through many iterations, requiring the creation of several prototypes and necessitating extensive communication between the designer and other experts to make sure that the finished design adheres to all the requirements. Furthermore, the software that is commonly used in the field is often only focused on one specific aspect of the design process. The primary goal of this thesis is thus to investigate the viability of design methods that incorporate several aspects, such as ergonomic, functional or aesthetic requirements, into a single design framework.



Figure 1.1: The design of furniture and indoor spaces is a difficult task that requires expert knowledge from a variety of disciplines. In this thesis, we present different computational design approaches that incorporate several aspects of the design task, including ergonomic, functional or aesthetic requirements. Left: A fabricated multi-purpose seating surface designed for various types of sitting and leaning using the interactive approach described in Chapter 3. Middle: Design for a seating surface that was synthesized for the given input pose using the automated method presented in Chapter 4. Right: A living room furniture layout synthesized using the data-driven approach described in Chapter 5.

There are several challenges regarding this. First of all, rules of ergonomics and indoor space design are often either presented in terms of rough guidelines on the one hand, or very specific examples that are difficult to generalize on the other. In order to integrate these rules into computational design methods, it is thus necessary to derive equations or functions that make it possible to computationally evaluate how well a given design fulfills the desired rules. Furthermore, the sheer quantity of rules often makes it difficult, if not even impossible to create a design that perfectly adheres to all rules. Sometimes, modifying a design according to a specific rule will result in another rule being violated. Therefore it is often necessary to find tradeoffs between different rules or prioritize one over the other.

The second challenge concerns the development of the algorithms that support the user in the design process. It is important to keep in mind who will be using the tool. For example, it can be expected that an expert user possesses sufficient knowledge regarding the design goals that need to be achieved and will thus desire as much control as possible over the design. Therefore, the design tool should support the user in regards to the ergonomic qualities of the design without interfering too much with the design process. On the other hand, for a novice user it is often desirable to keep the manual design effort as small as possible and thus incorporate rules for the functional and aesthetic design goals into the algorithm itself.

Finally, the third challenge concerns the fact that different design approaches each come with their own specific requirements. For example, interactive approaches require the underlying computations to be reasonably fast, often at the cost of accuracy, while

data-driven methods require the derived evaluation functions to be compatible with the underlying data and learning algorithms. It is thus not sufficient to address the first and second challenges individually.

## 1.2 Research Goals

The aim of the research presented in this thesis is to address the above mentioned challenges. We therefore formulate the following goals:

1. Study literature related to ergonomics and interior design that are relevant for the chosen applications and derive mathematical functions that can be used to evaluate or improve a given design.
2. Develop algorithms for the design of furniture and indoor layouts that support the user in achieving the desired design goals.
3. Perform empirical experiments using the developed methods and utilize the obtained results to further improve and validate the ergonomic evaluation functions and computational design algorithms.

It is important to note that established practices regarding the topics of seating and interior design can vary greatly between different cultures and eras of history [Pil05, PH10]. Exploring these varying contexts is out of scope for this thesis, as acquisition of the necessary data and expert knowledge, e.g. for the purpose of design evaluation, would be infeasible without first laying the ground work in a more familiar setting. The scope of this thesis is therefore restricted to seating and interior design in a modern and western cultural context.

The rest of the thesis is structured as follows. In Chapter 2, we discuss the current state-of-the-art related to the research presented in this thesis.

In Chapter 3, we introduce a novel method for the interactive design of body-supporting surfaces like chairs, benches or beds. Given a triangular mesh of a human body in a specific pose as input, we first compute an approximate pressure distribution on the human body to learn where it needs the most support. A given initial design for a body-supporting surface is then deformed to provide optimal support to the body while still conforming to the aesthetic design as much as possible. The design can furthermore be modified interactively by making changes to the target pose or directly editing the control mesh of the surface. With this approach, we tackle the first and second challenges by using the pressure distribution to steer the ergonomic requirements of the design, while leaving the aesthetic requirements to the designer by allowing a high degree of interactivity. In order to address the third challenge, we introduce some simplifications for the computation of the pressure distribution, since computing it accurately would be infeasible for an interactive approach.

The work presented in Chapter 4 serves as an extension to the previous chapter. While the interactive approach offers more control over the aesthetic design of the surface to the designer, it also has some drawbacks. In particular, if the initial design is not well suited for supporting the target pose, it is likely that the quality of the resulting design will also decrease. We therefore propose a method for the automated generation of body-supporting surfaces which depend entirely on the target pose. This method can be used on its own to create suitable seating furniture for a given input pose, or in conjunction with the approach described in Chapter 3 by means of generating an initial design for a target pose that can then be interactively modified. We furthermore extend our method for computing the pressure distribution on the target body by taking into account the weight of individual limbs and estimating the frictional forces acting on the body, increasing the accuracy of the resulting pressure distribution. While this extension comes with a higher computational cost, we can afford this increase of computation time since our proposed approach is non-interactive.

Since the ergonomic qualities of seating furniture do not only depend on the design of the furniture itself, but also its relation to the other elements of the surrounding indoor space, we introduce a data-driven approach for the design of indoor layouts in Chapter 5. In contrast to the previous two chapters, which mostly feature one-to-one or one-to-many relations (one piece of seating furniture to one or more target poses), the design of indoor layouts requires handling of many-to-many relations since the relative position and orientation of any arbitrary number of furniture elements can be dictated by different rules. Since these rules are so numerous and sometimes even contradictory, it is an infeasible task to define all of them explicitly. Using a data-driven approach makes it possible to implicitly learn about these rules given a large set of suitable examples. But good example data is not always readily available in large quantities; existing datasets may fulfill some of the desired rules, but lack in other areas. We therefore propose a deep-learning approach that makes use of the advantages of data-driven learning, while at the same time correcting flaws in the dataset using a small set of explicitly designed rules drawn from ergonomics literature.

### 1.3 Thesis Overview

The research presented in this thesis was conducted during the author's doctoral programme from 2016 – 2022. The first 3 years of the doctoral programme were spent as part of the Doctoral College Computational Design (DCCD) created by the Center for Geometry and Computational Design (GCD) at TU Wien. From April 2018 to September 2021, the thesis author conducted research as part of scientific projects funded by the Austrian Science Fund (FWF P27972-N31, FWF P29981) and the Vienna Science and Technology Fund (WWTF ICT15-082), also at TU Wien. From November 2021 to May 2022, the author furthermore participated in a research stay at New Jersey Institute of Technology, USA.

Chapters 3, 4 and 5 of this thesis are based on the following publications:

1. Paper 1: Kurt Leimer, Michael Birsak, Florian Rist, Przemyslaw Musialski. Sit & Relax: Interactive Design of Body-Supporting Surfaces. In *Computer Graphics Forum*, volume 37, pages 349-359. 2018. [LBRM18]
2. Paper 2: Kurt Leimer, Andreas Winkler, Stefan Ohrhallinger, Przemyslaw Musialski. Pose to Seat: Automated Design of Body-Supporting Surfaces. In *Computer Aided Geometric Design*, volume 79. 2020. [LWOM20]
3. Paper 3: Kurt Leimer, Paul Guerrero, Tomer Weiss, Przemyslaw Musialski. LayoutEnhancer: Generating Good Indoor Layouts from Imperfect Data. *ACM SIGGRAPH Asia 2022 Conference Papers (SA '22 Conference Papers)*. 2022. [LGWM22]

Paper 1 was also presented at the Pacific Graphics 2018 conference and the GCD Symposium in 2018, and furthermore received a Best Paper Honorable Mention Award at Pacific Graphics 2018. Furthermore, Paper 3 was presented at SIGGRAPH Asia 2022 in Daegu, South Korea.

During the duration of the doctoral programme, the author has also contributed to the following publications which are not described as part of this thesis:

1. Kurt Leimer, Lukas Gersthofer, Michael Wimmer, Przemyslaw Musialski. Relation-Based Parametrization and Exploration of Shape Collections. In: *Computers & Graphics*, volume 67, pages 127-137. 2017. [LGWM17]
2. Stefan Pillwein, Kurt Leimer, Michael Birsak, Przemyslaw Musialski. On Elastic Geodesic Grids and Their Planar to Spatial Deployment. In: *ACM Transactions on Graphics*, 39(4). 2020. [PLBM20]
3. Kurt Leimer and Przemyslaw Musialski. Reduced-Order Simulation of Flexible Meta-Materials. *SCF '20: Symposium on Computational Fabrication*. 2020. [LM20]
4. Kurt Leimer and Przemyslaw Musialski. Analysis of a Reduced-Order Model for the Simulation of Elastic Geometric Zigzag-Spring Meta-Materials. In: *Computers & Graphics*, volume 102, pages 187-198. 2022. [LM22]

## 1.4 Authorship Statement

The research work presented in Chapters 3, 4 and 5 is the result of collaborative efforts between the thesis author and several other researchers. In this section we describe the contribution of each collaborator to the relevant research papers.

Paper 1: The concept and ideas behind this work were created during regular discussions between the thesis author, Przemyslaw Musialski and Florian Rist. The majority of the technical implementation was done by the thesis author. The thesis author, Przemyslaw

Musialski and Florian Rist all contributed to the evaluation of the method via creation of the experimental results and conducting the design studies. Florian Rist was also in charge of producing the physical prototype. Michael Birsak mainly contributed to the technical implementation and writing of the Inverse Kinematics section 3.5. The writing, editing and revision of the manuscript was a collaborative effort by all authors.

Paper 2: This work consists of two major technical contributions. The concept and ideas behind the first contribution, the pressure map computation (Section 4.2), were created in regular discussions between the thesis author and Przemyslaw Musialski, with the technical implementation done by the thesis author. The second contribution, the furniture synthesis algorithm (Section 4.3) was developed as part of a diploma thesis project by Andreas Winkler [Win19]. The ideas and concepts for this part were created during regular discussions between the thesis author, Przemyslaw Musialski and Andreas Winkler, with the technical implementation of the algorithm mostly done by Andreas Winkler. The writing, editing and revision of the manuscript was done collaboratively by the thesis author and Przemyslaw Musialski — Andreas Winkler did not directly contribute to the production of the paper, but his technical description of the developed algorithms were used as reference for writing of Section 4.3 and Appendix A.4. Stefan Ohrhallinger mainly provided additional funding for the project.

Paper 3: The early concept and ideas for this project, in particular the evaluation of indoor layouts in terms of ergonomic qualities, were created in discussions between the thesis author, Przemyslaw Musialski and Tomer Weiss, while ideas for the technical realization of the project were conceptualized in discussions between the thesis author, Przemyslaw Musialski and Paul Guererro. The technical implementation was mostly done by the thesis author. All authors contributed to the writing of the paper, with the thesis author, Przemyslaw Musialski and Paul Guererro contributing to editing and revision of the manuscript.

# Related Work

In this chapter we discuss the current state-of-the-art related to the research presented in this thesis. We begin with a discussion of ergonomics literature (Section 2.1) from which we derive the rules used to evaluate and improve the designs created with the methods presented in this thesis. In Section 2.2, we cover the state-of-the-art research related to the design of furniture, which is the main application of the methods described in Chapters 3 and 4. Finally, we discuss recent work on the topic of indoor layout synthesis in Section 2.3, which is the task of the approach we present in Chapter 5.

## 2.1 Ergonomics

The discipline of ergonomics concerns itself with the design of systems in which humans play a significant role, with a primary goal of making these systems not just efficient but also safe and comfortable [Kro17]. Related tasks include the design and arrangement of furniture and workstations, planning of work schedules and the study of human physiology and psychology. Since one of the main contributions of this thesis is the development of methods and algorithms for the design of furniture and indoor layouts, we will discuss aspects of ergonomics that are closely related to these tasks in this section.

### 2.1.1 Comfort

Comfort is an important measure to evaluate the functional requirements of furniture. In particular, the question of the comfort while sitting is not novel and has been addressed in the field of ergonomics already quite a while ago. For instance, Lueder [Lue83] provided a survey of the assessment of comfort that is relevant to the design of office furniture. Also, [ZHD96] analyzed the multidimensional factors of sitting comfort and discomfort. Other surveys studied the comfort of sitting in vehicle seats [DKEV03], and especially in wheelchairs [Sta95].

Historically, the most elementary way to determine comfort or discomfort of a seat is to keep note of the subjective feelings of its users ([Her58, Jon69]). Subjective measures are the most direct and reliable indicators of comfort, however, in most furniture design applications, objective measures would be advantageous compared to subjective ratings [DKEV03]. Therefore, researchers have aimed to find a relation from subjective feelings to objective measures for comfort and discomfort. De Looze et al. [DKEV03] identified a variety of objective measures for comfort or discomfort from literature in medicine and ergonomics and concluded that pressure distribution showed the most clear association with the subjective ratings. Similar findings have also been shown in many other studies [KTOK82, YDF92, LTPSR08, ZFBV12, NNL<sup>+</sup>12]. Since our goal for the design methods presented in Chapters 3 and 4 is to aid users in designing seating furniture that provide a high degree of comfort during sitting, we therefore choose the pressure distribution as an objective measure to evaluate and improve the designs.

### 2.1.2 Biomechanics

Related to the subject of comfort is the field of Biomechanics, which studies mechanical effects on human bodies, such as forces and torques (often also called moments) acting on the joints [Hal95, RCH<sup>+</sup>04]. For the approach presented in Chapter 4, we consider the weight of individual limbs and their effects on the joints of the body when computing the pressure distribution. We base these calculations on a link-segment skeleton model of a human body which is also often used for inverse dynamics [DAdG04, RCH<sup>+</sup>04, FAK<sup>+</sup>14]. Our method is based on a linear least-squares approach of inverse dynamics as often used for such purposes [Kuo98, DAdG04, Vli14]. However, we extend the model by combining the skeleton with a skin of the human using linear blend skinning [BP07], which is commonly used in computer graphics. In order to make our model physically more plausible, we assign each limb a weight and a center of mass that have been determined empirically by Plagenhoef et al. [PEA83]. In contrast to physical simulations, in our method we propose a linearized model of the transmission of forces from bones to the skin in order to determine an optimal distribution of reaction forces on the body in such a way that the moments acting in the given posture are minimized.

### 2.1.3 Interior Design

The field of interior design concerns itself with the planning, layouting and designing of interior spaces within buildings [CB12]. One of the tasks that is of particular importance is the selection and arrangement of furniture within a given indoor space such that functional, aesthetic and ergonomic requirements are fulfilled. Naturally, there exists a multitude of rules and guidelines that steer this design process. Here we will discuss a subset of these rules that we selected for the purpose of evaluating and improving indoor layouts generated by the deep learning approach presented in Chapter 5.

Visibility is an important aspect to keep in mind when arranging furniture, especially when involving activities performed in a seated position [GK97]. Ideally, a target object



should be placed within line of sight without the need for substantial rotation of the neck or eyes. Different ranges of angles have been reported as ideal in various studies, but for the sake of simplicity and ease of implementation, we regard the quality of an arrangement in terms of visibility to decrease as the angle between the orientation of seating furniture and the direction of the target object increases.

Illumination is another crucial design element of indoor layouts. While the human eye can adapt to differences in brightness, doing so takes time and thus results in an overall loss of visual performance [GK97]. Lighting conditions should therefore be designed to avoid such discrepancies in contrast. For this purpose we identify two rules: Light sources should be positioned in such a way that they provide sufficient illumination to a visible target object [Kro17] and light sources should not be directly within the field of view in order to avoid direct glare [GK97].

Finally, it is important that the elements of a layout that need to be interacted with are accessible. First, when in a seated position, target objects should be comfortably reachable without needing to leave the seat. For our approach, we base the evaluation of this criteria on reported ideal ranges [GK97]. And second, moving between different elements of a layout should be easy by avoiding obstructions on the path between objects. One way to achieve this is by making sure that sufficient space is available in front of each layout element [CB12].

## 2.2 Furniture Design

From a general perspective, the central goal of the methods presented in Chapters 3 and 4 is to provide a computational design system for usable seating furniture. Furniture creation is a very broad task with a rich history in a variety of fields including wood working, product design or medicine. An important question is whether a designed seating surface is aimed for a general application or to be used in a specific situation only. In any public place or transport that need to accomodate a large variety of people, the use of a one-size-fits-all solution is inevitable. With modern design methods, for instance using 3d scans of human body shapes [SBK<sup>+</sup>16], a large range of body sizes can be covered.

On the other hand, there is also a need for more personalized design solutions. In a human centered design process the attention is shifted to the needs and requirements of a human person. In the context of seating furniture design, the goal is thus to find a seating surface that optimally matches the requirements of a specific individual, ranging from physical properties such as shape or size [RP08] to semantic constraints. The approaches presented in Chapters 3 and 4 fall into this category. In this section, we first discuss research related to physically-informed furniture design in general before focusing on pose-driven design in particular.

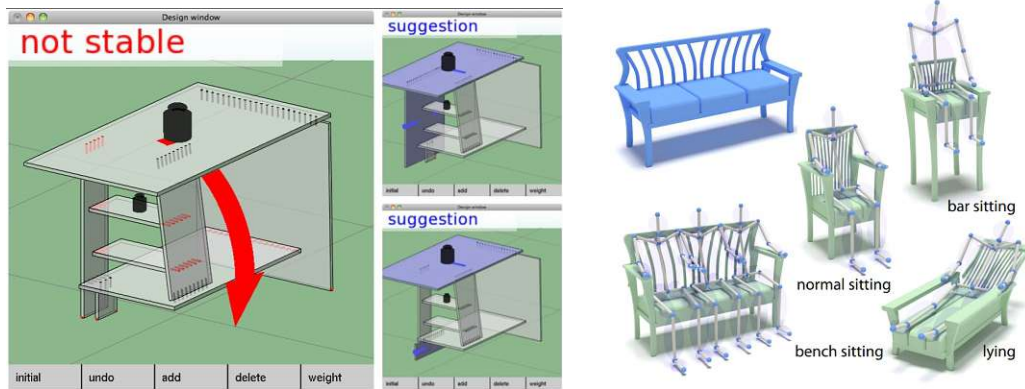


Figure 2.1: Examples of related work on the topics of physically-informed and pose-driven design. Left: An interactive tool by Umetani et al. [UIM12] provides suggestions for the design of stable furniture. Right: The approach proposed by Zheng et al. [ZLDM16] allows reshaping of existing 3D models to fit the dimensions of an input pose. Images sourced from the respective papers.

### 2.2.1 Physically-Informed Design

In Chapter 3 we present an interactive design method that is driven by physically derived information, which is quite common in the field of computer graphics. For example, Saul et al. [SLMI11] introduced SketchChair, an application for easy design of chairs for novice users using an easy-to-use 2d sketching interface. The stability of the sketched designs can be validated with a simple physical simulation, but the application does not feature algorithmical optimization of the design. The finished furniture sketch can be then fabricated from planar sheet materials by cutting them out and putting them together. Umetani et al. [UIM12] proposed a system for computational design of shelves and similar furniture from planar pieces, using a physical model which supports the users during the design such that only structurally stable models are created. In contrast to SketchChair, the framework also provides suggestions to the user that can improve the stability of the design. Lee et al. [LCMS16] propose a system designed around VR technology that allows users to design furniture via gestures and voice commands. In this approach, the body of the user is captured with a camera and can be used as reference to customize the dimensions of a furniture design.

Other research focuses on automated systems aimed to design suitable furniture in an automated process. User interaction is mostly limited to customizing input data and parameters. The method presented in Chapter 4 also falls into this category. For instance, Zheng et al. [ZLDM16] proposed a method for ergonomics inspired reshaping and exploration of collections of models in order to create novel shapes. While their method takes some ergonomic aspects into account, such as the dimensions of individual body parts, their designs are neither checked for physical validity nor are the physics of the human body considered. Researchers in furniture design have also used hybrid

approaches for their systems, where the furniture shapes are created by an automated system, but users can steer or manipulate the design process in various stages [FCSF17].

### 2.2.2 Pose-Driven Design

The field of computer vision and machine learning has also utilized poses for the analysis and classification of objects like sitting furniture. For instance, [GGV11] introduced a system that uses an affordance detector in order to determine the functionality of objects by the way humans could interact with them. A similar idea has been pursued by [KCGF14] who use supervised learning on a set of poses in order to further generate static poses of how a functional object could be potentially used. Furthermore, [KL14] also proposed a method for the generation of poses to a given geometry.

Since much information can be derived from human poses, this information can be used for the purpose of design. Early research related to pose-driven design aimed to find seating surfaces which match a general class of poses or guidelines [BRIT08]. In 2017, Fu et al. [FCSF17] introduced a shape synthesis approach with the goal of creating hybrid shapes usable by humans. While this work is not limited to furniture shapes, it serves as an example for pose-driven design. Lee et al. [LCMS16] proposed a novel user centric furniture design process, making digital design interfaces accessible for casual users by using poses and gestures, speech commands and augmented reality technology. Zheng et al. [ZLDM16] introduced an interactive system that selects and adapts seating furniture for user-specified human body and input poses. An entirely different approach at personalized furniture design is presented by Wu et al. [WWT<sup>+</sup>18]: ActiveErgo is a monitored workplace environment that dynamically adjusts its parameters like desk height or chair position in accordance with ergonomic guidelines, adjusted to the user.

From a technical point of view, the method presented in Chapter 3 incorporates fitting of Catmull-Clark subdivision surfaces [CC78] to a given human pose in order to support it optimally, which we have chosen due to their  $C^2$ -smoothness that result in very elegant shapes. Therefore we utilize methods for optimization of the control mesh for the fitting of subdivision surfaces [MK05, CWH<sup>+</sup>07]. However, in the literature there have also been a number of methods which propose interactive design of free-form surfaces, e.g., [IMT99, NISA07, JHR<sup>+</sup>15, LPL<sup>+</sup>17]. Finally, we also utilize inverse kinematics [Bus04] for additional relaxation of the given posture in order to improve the contacts surface [SZGP05, DSP06].

## 2.3 Layout Synthesis

Interior spaces and their layouts are part of everyday life. Hence it is not surprising that such layouts are also an important part of multiple virtual domains, ranging from entertainment, architecture, to retail. For example, organizations such as Ikea and Wayfair are actively working toward understanding their customers needs [ACW<sup>+</sup>19].

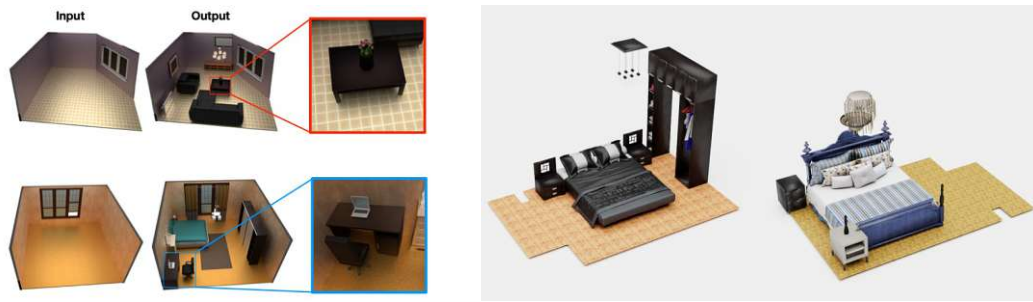


Figure 2.2: Indoor scenes synthesized by existing transformer-based deep learning models which are closely related to our approach described in Chapter 5. Left: Scenerformer [WYN20]. Right: ATISS [PKS<sup>+</sup>21]. Images sourced from the respective papers.

Typically, each domain has different requirements and needs, which require manual design [Way22].

In practice, designing layouts is a laborious task due to high dimensional design space, ranging from selecting relevant furniture pieces, to arranging the target space to fit the design goals. To alleviate such manual workflow, researchers have proposed multiple computational methods to assist in layout design. Below we classify previous work based on their approach.

### 2.3.1 Deep Learning Methods

With modern computer hardware enabling the use of increasingly complex deep neural networks, recent work in this field increasingly relies on example data to learn how to synthesize new layouts. Such 3d scene data and the data modality is an important factor in deep learning [FCG<sup>+</sup>21]. Examples of how this kind of data can be represented include images, graphs or sequences. Early deep learning work utilizes top-down images of layouts to understand object-object layout relationships [WSCR18]. However, images do not naturally contain sufficient detail for the network to synthesize complex human-centered layouts. Graphs have also been proposed as a means to encode spatial layout information [WLW<sup>+</sup>19, ZWK19, LZWT20]. Using this kind of representation, the scene synthesis problem is transformed to predicting appropriate graph nodes and edges.

In addition to images and graphs, researchers explored how to use other 3d scene data representations for synthesis. Zhang et al. [LPX<sup>+</sup>19] synthesize scenes by sampling from a vector that represents the spatial structure of a scene. Such structure encodes a hierarchy of geometrical and co-occurrence relations of layout objects. [ZYM<sup>+</sup>20] proposed a hybrid approach that combines such vector representation with an image-based approach. Also other utilize graph structures to describe scene layouts [DYZ<sup>+</sup>20]. Yang et al. [YZY<sup>+</sup>21] combine such vector representation with Bayesian optimization to improve furniture placement predictions of the generative network. Recently, variational autoencoders have been proposed for indoor layout synthesis [CZW<sup>+</sup>22].

Most recently, researchers have proposed to use neural networks based on transformers [WYN20, PKS<sup>+</sup>21]. The authors mention that one advantage of transformers is a faster synthesis compared to other other deep-learning approaches. For the approach presented in Chapter 5 we also make use of neural networks based on the transformer architecture. However, in contrast to existing work which does not account for ergonomic qualities and thus sometimes results in misplaced furniture items if such errors are also present in the training data, our approach aims to improve the learning process by introducing additional rules based on ergonomics.

### 2.3.2 Other Approaches

Before the era of deep learning, early work considered layout synthesis as a mathematical optimization problem, where a set of constraints describe the layout quality in terms of energy and energy functionals [YYT<sup>+</sup>11, MSL<sup>+</sup>11, WLD<sup>+</sup>18]. The layout is then optimized via stochastic or deterministic optimization process.

Other researchers proposed data-driven methods. In such methods, abstract layout structure and object-object relations are extracted from scene datasets. Earlier work by Fisher and colleagues [FRS<sup>+</sup>12] proposed a method for arranging small layout object arrangements, using pairwise spatial heat-map distributions for learning object relationships. Qi et al. [QZH<sup>+</sup>18] use interaction affordance maps for each layout object for stochastic layout synthesis. However, they only take into account static poses and spatial relationships between furniture pieces. Similarly, Fisher et al. [FSL<sup>+</sup>15] used annotated 3d scans of rooms to identify the kind of activities an environment supports. Based on such activity maps they can synthesize small objects arrangements. Other researchers also learn layout structure from 3d scans for scene synthesis [KLTZ16]. They extract manually defined geometric relationships between objects from such scans, which are then placed using a stochastic optimization.

Other research has made progress towards incorporating human-centered considerations for 3d scene synthesis. Fu et al. [FCW<sup>+</sup>17] use a graph of objects to guide a layout synthesis process. However, they only consider static human poses in relation to activities. Zhang et al. [ZHPY21] and Liang et al. [LLL<sup>+</sup>19] focus on optimal work-space design. While the authors demonstrate novel use of simulation and dynamic capture of agent in action metrics, they only focus on mobility and accessibility based factors. In [PRB<sup>+</sup>18], the authors demonstrate how to evaluate the functionality of layouts. However, this work does not include 3d scene synthesis.

Early work [YYT<sup>+</sup>11, MSL<sup>+</sup>11] has also included ergonomic and interior design knowledge into the layout design process. The approach we present in Chapter 5 differs from these existing methods in two major aspects. First of all, their methods require the manual definition of a number of additional layout design rules. Second, their methods are designed to optimize the arrangement of an existing furniture layout, while our approach can synthesize entirely new layouts with desired characteristics.



# Interactive Design of Seating Furniture

## 3.1 Introduction

Product design of human body-supporting objects—for instance sitting furniture—is a difficult design problem. Usually designers try to combine two major aspects: the actual function of the product as well as the aesthetics of its shape. One of the very well known maxims of this process is the *form follows function* principle, which dictates that the form of an object is at least in part determined by its function. However, a unique aesthetic is often desired to make the product stand out from similar products. Therefore the goal is to create an appealingly looking shape and at the same time to make this shape fulfill the functional requirements.

Traditionally, this is achieved by employing an iterative process where a number of prototypes needs to be produced in one-to-one scale in order to figure out what is actually comfortable. The main reason for this is that it is extremely difficult to judge in advance—especially if only a digital model is created—how comfortable and functional the final product will eventually be.

Our goal is to provide designers a novel way to create free-form surfaces that automatically adapt to human postures. Hence, we propose a new approach for the interactive design of body-supporting surfaces which automatically nestle to the shape of the body in order to make the human feel comfortable in the current pose. To achieve this goal, we propose a measure that gives an indication of how a human in a given posture should be supported in order to distribute the pressure uniformly on an as large an area as possible. Pressure distribution has been identified as one of the objective indicators of comfort [Lue83, Sta95, DKEV03].

### 3. INTERACTIVE DESIGN OF SEATING FURNITURE

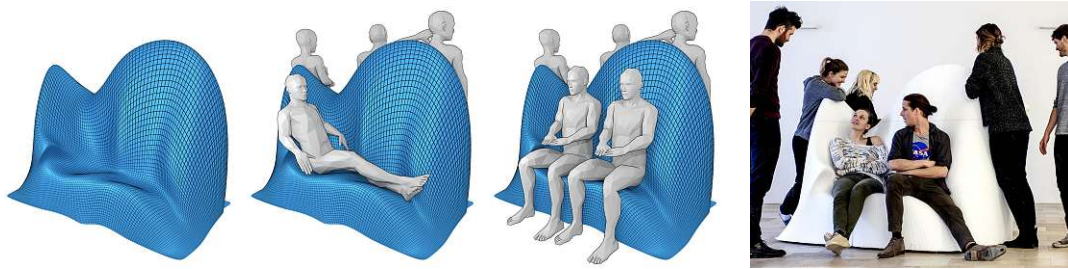


Figure 3.1: Multi-purpose sitting surface designed for various types of sitting and leaning using our method. Left: three renderings. Right: fabricated result inspected by design students.

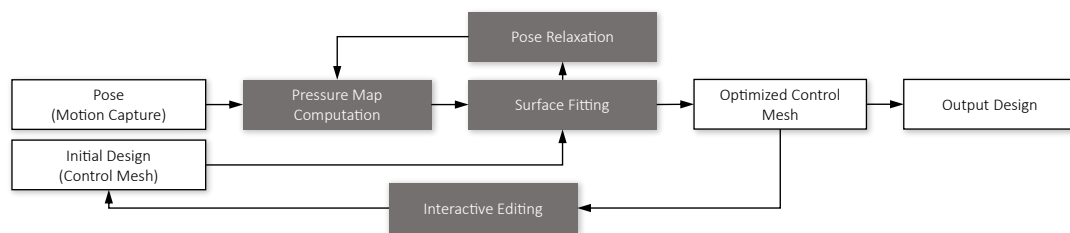


Figure 3.2: Overview of the interactive design system. Given initial design and a set of poses captured by a motion capture device, our system estimates a pressure distribution on the bodies in the given poses. The artist can then create a social scenario using the given poses and provide a initial control mesh for a surface. Our system then computes an optimized smooth subdivision surface and its control mesh using our surface fitting algorithm. In further design steps, the computed control mesh can be edited interactively and used as input again to generate a new design. Our pose relaxation algorithm also makes it possible to adapt the input poses to the computed subdivision surface if necessary to ensure that all poses can be supported well.

Technically, we accomplish it by computing an importance map on the human body which is proportional to the physical pressure which the body is exposed to if resting in a given pose. Further, this importance field on the surface of the body is used for fitting of optimally supporting Catmull-Clark subdivision surfaces in an interactive design application.

Our contributions are the following: (i) we cast the problem of interactive design of body-supporting surfaces driven by the pressure distribution acting on the body, where we propose an approximative, physically validated method for an efficient computation of body pressure in Section 3.3. (ii) Further, we provide an interactive design system for free-form surface fitting in Section 3.4 as well as for pose relaxation in Section 3.5. (iii) Finally, we utilize our solution in collaboration with designers in order to design a real-world example and we fabricate a functional product, which we document in Section 5.4.



## 3.2 Overview

Figure 3.2 shows the workflow of our interactive design system. The input is one or more human body poses given by triangular meshes and an initial design of the actual surface. In practice, since we are using Catmull-Clark subdivision, the initial design is given by its control mesh, and can be in the most trivial case just a flat patch.

For the bodies of humans we use the BLENDER plugin provided by Manuel Bastioni [Bas18], which allows the generation of body meshes with varying attributes, like gender, mass, size, stature, etc. Since the meshes are skinned and rigged to a skeleton, the user can adjust the poses manually, or alternatively, the poses can be created with a motion capturing device, for instance using the PERCEPTION NEURON system [Neu18]. We have utilized the latter for most of our designs.

In the next step the user can compose a scenario of sitting humans by selecting the captured poses and placing them on the modeling canvas with respect to the control mesh. Our system then computes the pressure distribution necessary to support each human in the given pose using the method further described in Section 3.3 and uses this information for fitting a subdivision surface as described in Section 3.4. Additionally, the user can decide to adjust selected poses, either manually, or using inverse kinematics in order to relax the pose with respect to the already computed surface, which is described in Section 3.5. Finally, the new control mesh can also be edited by either moving or fixing its vertices. This process can be repeated iteratively and the design can be explored until a satisfactory result is achieved.

## 3.3 Pressure Field Computation

Our goal is now to find a physically plausible distribution of pressure to which the body is exposed to if resting in a given pose. It is important to support the body where the relative pressure is high. At the same time, we want to make the contact area as large as possible to keep absolute pressure peaks low, so the pose can be considered as comfortable [DKEV03], as shown in Figure 3.3, right. Usually, the pressure distribution on the body would be found using a sophisticated physical simulation which is time consuming. Since our goal is to achieve interactive rates, we propose a simplified model where we assume the human is a rigid body.

### 3.3.1 Pressure Model

In order to compute an approximated distribution of the pressure acting on the human body, we build on top of the static Coulomb friction model (cf. Figure 3.3). In that model, the force  $\mathbf{f}$  acting at each contact point can be split into its normal component  $\mathbf{f}_n$  and its tangential components  $\mathbf{f}_{t_1}$  and  $\mathbf{f}_{t_2}$ . In order to keep the body in static equilibrium, we need to counteract these forces by reaction forces  $\mathbf{r} = \mathbf{r}_n + \mathbf{r}_{t_1} + \mathbf{r}_{t_2}$ , such that  $\int_{\Omega} \mathbf{r} \, dA = m_b \mathbf{g}$ , where  $m_b$  is the mass of the body and  $\Omega$  is the supported contact area.

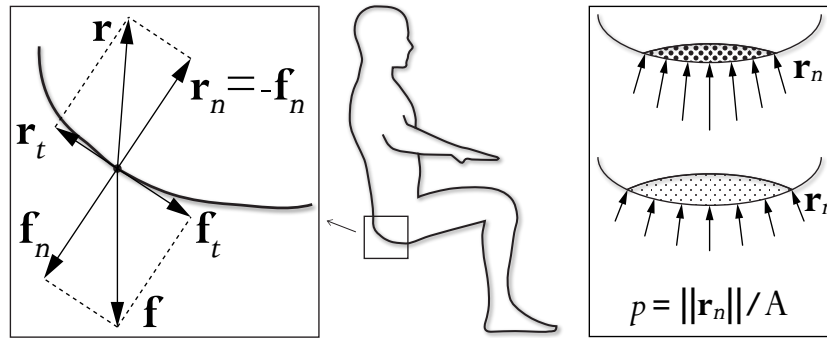


Figure 3.3: Illustration of the forces acting on a rigid body. Left: reaction forces split into the normal and tangential components. Right: definition of pressure as normal force acting per unit area. If the same forces are acting on a larger area, the pressure is lower.

The actual pressure  $p = \frac{1}{A} \|\mathbf{r}_n\|$  is the magnitude of the normal component of the reaction force  $\mathbf{r}_n$  divided by the area  $A$  the force is acting on (cf. Fig 3.3, right). The tangential components  $\mathbf{r}_t$  are the particular friction forces, whose magnitudes—according to the linear friction model—must be smaller than the magnitudes of the respective tangential components of the body force  $\mathbf{f}_t$ . Otherwise, the body would slip away from the support. This is usually ensured by setting the reaction force components to

$$\mathbf{r}_n = -\mathbf{f}_n, \quad \text{and} \quad \|\mathbf{r}_t\| \leq \mu \|\mathbf{r}_n\|$$

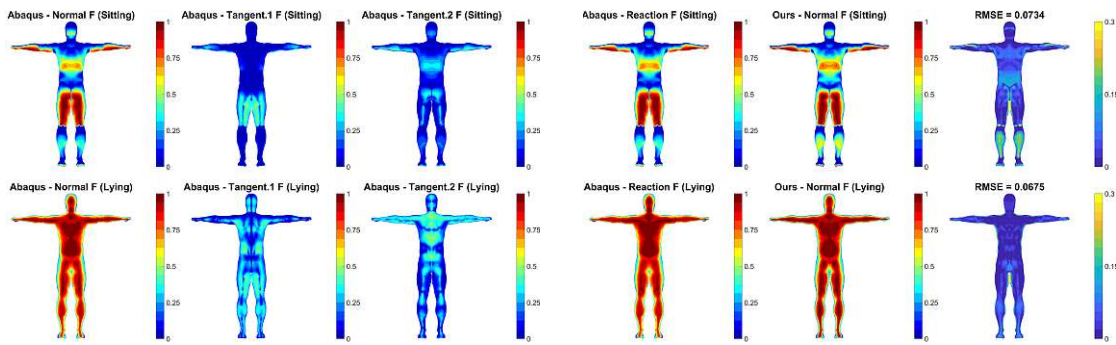
where  $\mu$  is the friction coefficient that depends on the roughness of the contacting surfaces. This ensures that the friction forces stay within the so-called friction cone.

### 3.3.2 Pressure Approximation

In practice, the normal components of the reaction forces are much larger and thus more important than the tangential components of the reaction forces. This is also evident from the physical simulation we have performed with a default value of  $\mu = 0.5$  (cf. Section 4.2.2 and Figure 3.4b). Moreover, considering the friction coefficients of common materials, humans usually do not run the danger to slide down from the seat. Since we aim for fast computation times over accuracy in our strongly simplified model, we ignore the tangential components and utilize only the normal directions for these reasons.

We assume there exists one reaction force  $\mathbf{r}$  for each vertex  $\mathbf{v}$  of the body mesh whose normal is pointing sideways or downwards (cf. Figure 3.3, right box). Vertices with normals pointing upwards are excluded from the computation, since they cannot be supported. In order to find out how the normal forces  $\mathbf{r}_n$  and the pressure  $p$  are distributed on the surface, we consider the projection of the gravity direction vector on the unit surface normal  $\mathbf{n}$  scaled by the Voronoi area of each vertex  $\mathbf{f} = (0, -A, 0)$ , such that

$$\mathbf{r}_n = -(\mathbf{n} \cdot \mathbf{f}) \mathbf{n} \quad \text{and} \quad p = \frac{1}{A} \|\mathbf{r}_n\|.$$



(a) Results of the forces computed by the physical simulation using ABAQUS (cf. Section 4.2.2). From left to right: magnitude of the normal forces followed by both tangential forces.

(b) Comparisons of our results to ABAQUS. Left: reaction forces (magnitude of the vector sum of normal and tangential forces). Middle: our result (magnitude of normal forces). Right: absolute difference of left and middle.

Figure 3.4: Results of physical simulation and comparison to ours. Top row: sitting pose—please note that for visualization purpose we render the results on a T-pose. Bottom row: lying T-pose. Please notice the different range for the error image.

In other words, we assume a good reaction force distribution can be approximated by considering the body as fully surrounded (e.g., enveloped) by a perfectly fitting support, where we ignore the friction forces and the body weight.

Note that the local vertex area  $A$  is factored out for the pressure computation and that in fact, the sum over the reaction forces on the body does not equal the body weight. Nonetheless, there is a linear relationship, and since we normalize the pressure in the range  $[0..1]$  and give up the physical units—which is still sufficient for our application (cf. Section 3.4)—, this issue can be ignored.

While this model might appear too oversimplified, in comparisons to professional FEM simulations we show (cf. Section 4.2.2), that this approximation has an error (RMSE) in the range of 0.07 on a  $[0..1]$  scale and it is still well-suitable for the purpose of an importance map for surface fitting.

### 3.3.3 Comparison to Physical Simulation

In order to justify our simplified model, we perform a FEM simulation using the professional physical simulation software ABAQUS [Smi09]. For this purpose we simulate two selected poses—a lying pose and a sitting pose—in a setup as proposed in our model in order to compute ground truth values. Note that here we also include the tangential forces.

For the simulation we create a rigid shell in the shape of a negative mold of the body by taking all body-mesh faces pointing downwards or sideways. This shell serves as the

contact surface which fully supports the body. For the human body, we create a single volume domain as a rigid body. The shell is completely locked in place by boundary constraints, while the body is moved downward by a force of  $735N$  (roughly equivalent to a body weight of  $75kg$ ) applied on the center of mass. This procedure is standard for the simulation of a single rigid body in ABAQUS.

To model the contact between body and shell we use a linear pressure-overclosure relationship with contact stiffness of  $8e+12$ . The tangential contact behavior is modeled with an isotropic friction coefficient of  $\mu = 0.5$  and an elastic slip of  $1e-10$ . Furthermore, we select the set of vertices at which reaction forces are computed in our method by choosing those vertices of the body mesh that are also included in the corresponding contact surface shell to make sure that the contact surface is the same in both methods. The computation time in ABAQUS took a total of 10 seconds for the simulation plus preprocessing, while our system takes 0.03 seconds on average.

In Figure 3.4a we show the magnitudes of the particular normal forces  $\hat{\mathbf{r}}_n$  as well as the tangential forces  $\hat{\mathbf{r}}_{t_1}$  and  $\hat{\mathbf{r}}_{t_2}$  computed in ABAQUS. Please note that all force magnitudes are taken absolute and normalized to the range of  $[0...1]$  to the maximum value of  $\hat{\mathbf{r}}_n$ .

For the comparison to our results we measure the root-mean-squared error (RMSE) between our estimated normal forces  $\mathbf{r}_n$  and the reaction forces  $\hat{\mathbf{r}}$  computed in ABAQUS. Note that the reaction force is the magnitude of the sum of normal and tangential forces. We have chosen this comparison (cf. Figure 3.4b) in order to emphasize why we can omit tangential components without a large error. In fact, the RMSE is about  $0.06 - 0.07$  for values scaled in a range of  $[0...1]$ .

## 3.4 Surface Fitting

In this section we present how the approximated pressure distribution can be utilized for fitting of optimal support surfaces in an interactive modeling application. Our goal is to generate a surface that allows one or multiple persons in a specified posture to sit or lean on it. To do this, we take the distribution of reaction forces of each given posture as input, and then use a Catmull-Clark subdivision algorithm to fit a smooth surface to the vertices on the body that need to be supported.

Using a quad-mesh  $M^0 = \{\mathbf{V}^0, \mathbf{F}^0\}$  as a control mesh, we want to compute the optimal control mesh  $M^* = \{\mathbf{V}^*, \mathbf{F}^*\}$  such that the subdivided mesh  $M^s = \{\mathbf{V}^s, \mathbf{F}^s\}$  (of some chosen subdivision level  $s$ ) has minimal distance to the target body mesh  $M^b = \{\mathbf{V}^b, \mathbf{F}^b\}$ . To do this, we first compute for each sample point  $\mathbf{v}_k^b \in \mathbf{V}^b$  the closest vertex point  $\mathbf{v}_k^s \in \mathbf{V}^s$  of the subdivided mesh  $M^s$ . Please note that two different points  $\mathbf{v}_j^b \neq \mathbf{v}_k^b$  may share the same closest vertex  $\mathbf{v}_j^s = \mathbf{v}_k^s$ . As a distance metric we use a linear blend of the point-to-point distance and the tangential distance, which is the distance from to the tangent plane of the closest point. The reasoning for this is that we would like regions of the surface to still be able to move tangentially once they are close enough to the target body.

Our aim is to find the control mesh vertices  $\mathbf{v}^* \in \mathbf{V}^*$  that minimize

$$E^{sds} = \sum_k (1 - \tau) \rho_k \|\mathbf{v}_k^b - \mathbf{v}_k^s\|^2 + \tau \rho_k \|(\mathbf{v}_k^b - \mathbf{v}_k^s) \mathbf{n}_k^s\|^2 + \sum_j \sigma \|L(\mathbf{v}_j^*)\|^2 + \gamma_j \|\mathbf{v}_j^* - \mathbf{v}_j^0\|^2,$$

with the first two terms being the point-to-point and tangential distances, the third being a smoothing term, and the last term being a penalty that prevents select control vertices from moving too far. The value  $\rho_k$  is the importance of the vertex  $\mathbf{v}_k^b$  based on the computed reaction forces and is given by

$$\rho_k = \frac{\|\mathbf{r}_k\|}{\max_i \|\mathbf{r}_i\|}.$$

$\mathbf{n}_k^s$  is the normal vector at  $\mathbf{v}_k^s$ . The parameter  $\tau$  blends between the two distance metrics and is initialized as 0 and is increased with each iteration step, up to a user defined maximum  $\tau_{max} \leq 1$ . The operator  $L(\cdot)$  is the discrete cotangents Laplace-Beltrami operator [BKP<sup>+</sup>10] and serves as a smoothing term, regulated using the weight parameter  $\sigma$ . The effect of this smoothing term is demonstrated in Figure 3.7. Furthermore, to give the user a control over the look of the resulting design, we can soft-constrain any control mesh vertices  $\mathbf{v}_j^* \in \mathbf{V}^*$  to their initial locations using the weight parameter  $\gamma_j$ .

To prevent the surface from intersecting with the target body, we use the following intersection constraint: For every previously found point  $\mathbf{v}_k^s \in \mathbf{V}^s$  on the surface we search for the closest point  $\hat{\mathbf{v}}_k^b \in \mathbf{V}^b$  on the target body. Then the constraint  $(\mathbf{v}_k^s - \hat{\mathbf{v}}_k^b) \hat{\mathbf{n}}_k^b \geq 0$  ensures that the point  $\mathbf{v}^s$  stays of the positive side of the tangent plane of  $\hat{\mathbf{v}}_k^b$  with normal vector  $\hat{\mathbf{n}}_k^b$ .

Minimizing the energy  $E^{sds}$  leads to a non-linear system of equations since the point  $\mathbf{v}_k^s$  on the surface closest to  $\mathbf{v}_k^b$  on the body changes as the surface deforms. We instead linearize this problem and solve the resulting linear system of equations iteratively:

$$\begin{aligned} \min_{\mathbf{V}^*} \quad & \|\mathbf{A}\mathbf{V}^* - \mathbf{b}\|^2 \\ \text{s. t.} \quad & \mathbf{C}\mathbf{V}^* \leq \mathbf{0} \end{aligned} \tag{3.1}$$

with  $\mathbf{A}$  and  $\mathbf{b}$  containing the terms of the energy  $E^{sds}$  and  $\mathbf{C}$  containing the non-penetration constraints. For a more detailed description of these terms, please refer to Appendix A.1.

To summarize, we perform algorithm 3.1 iteratively until the solution does not improve any more.

### 3.5 Pose Relaxation

A further step we introduce for the convenience of the designer is pose relaxation. It allows to register the given pose to a given surface and also to further relax it in order to better

---

**Algorithm 3.1:** Surface Fitting

---

```

1 for  $i \leq i_{max}$  do
2   | Subdivide  $M_i^0$  to compute  $M_i^s$  and its normals  $\mathbf{N}_i^s$ .
3   | Find the closest vertices  $\mathbf{v}_k^s \in \mathbf{V}_i^s$  for each sample point  $\mathbf{v}_k^b \in \mathbf{V}^b$ .
4   | Solve the linear system in Eq. 3.1 to compute  $\mathbf{V}_{i+1}^0$ .
5 end

```

---

adapt to the surface. Since our input meshes come from a BLENDER plugin [Bas18] and are rigged to a skeleton, we can also use this skeleton to perform inverse kinematics [Bus04] on the pose and propagate the deformation of the skeleton to the surface mesh using linear blend skinning [MTLT89, BP07].

Our goal is to minimize the distance between parts of the pose that need support and the corresponding regions on the computed surface while at the same time avoiding penetrations between the pose and the surface. Further, we do not want the pose to change significantly and therefore also want to penalize large changes of joint angles.

Thus, we parametrize the rigged pose by the joint angles  $\boldsymbol{\theta}$  of the skeleton, and we cast the problem as an optimization task

$$\min_{\boldsymbol{\theta}} \lambda_{d_p} \left\| E^{d_p} \right\|_2^2 + \lambda_{d_t} \left\| E^{d_t} \right\|_2^2 + \lambda_p \left\| E^p \right\|_2^2 + \lambda_s \left\| E^s \right\|_2^2,$$

where  $E^{d_p}$ ,  $E^{d_t}$ ,  $E^p$  and  $E^s$  are vectorial energy terms for the point-to-point and point-to-tangent-plane distance between pose and surface, the penetration between the pose and the surface, and the similarity between the input  $\boldsymbol{\theta}^0$  and the output  $\boldsymbol{\theta}$  joint angle vector respectively.

The terms  $E^{d_p}$ ,  $E^{d_t}$  and  $E^p$  are not evaluated on the whole body mesh but only on a subset of vertices that belong to regions that need support. Therefore, we start the inverse kinematics procedure with an evaluation of the reaction forces that are needed for an optimal support of the pose (cf. Section 3.3) and denote the subset of  $m$  supportable body vertices with  $\mathbf{V}^b$  and the magnitude of the reaction force for each  $\mathbf{v}_i^b \in \mathbf{V}^b$  with  $\|\mathbf{r}_i\| = r_i$ . The subset of corresponding surface vertices  $\mathbf{V}^s$  is identified by a nearest neighbor search. For the evaluation of the point-to-tangent-plane distance  $E^{d_t}$ , we additionally compute the tangent planes at all  $\mathbf{v}_i^s \in \mathbf{V}^s$  which are defined by the surface normals  $\mathbf{N}^s$  and the distance values  $\mathbf{D}^s$  such that  $\mathbf{n}_i^s \cdot \mathbf{v}_i^s - d_i^s = 0$ .

In detail, we define the energy terms as

$$\begin{aligned}
 E_i^{d_p}(\boldsymbol{\theta}) &= r_i \left\| \mathbf{v}_i^b(\boldsymbol{\theta}) - \mathbf{v}_i^s \right\|, \\
 E_i^{d_t}(\boldsymbol{\theta}) &= r_i \left( \mathbf{n}_i^s \cdot \mathbf{v}_i^b(\boldsymbol{\theta}) - d_i^s \right), \\
 E_i^p(\boldsymbol{\theta}) &= r_i \left( \frac{1}{2} - \frac{1}{\pi} \arctan \left( a \left( \mathbf{n}_i^s \cdot \mathbf{v}_i^b(\boldsymbol{\theta}) - d_i^s \right) + b \right) \right),
 \end{aligned}$$

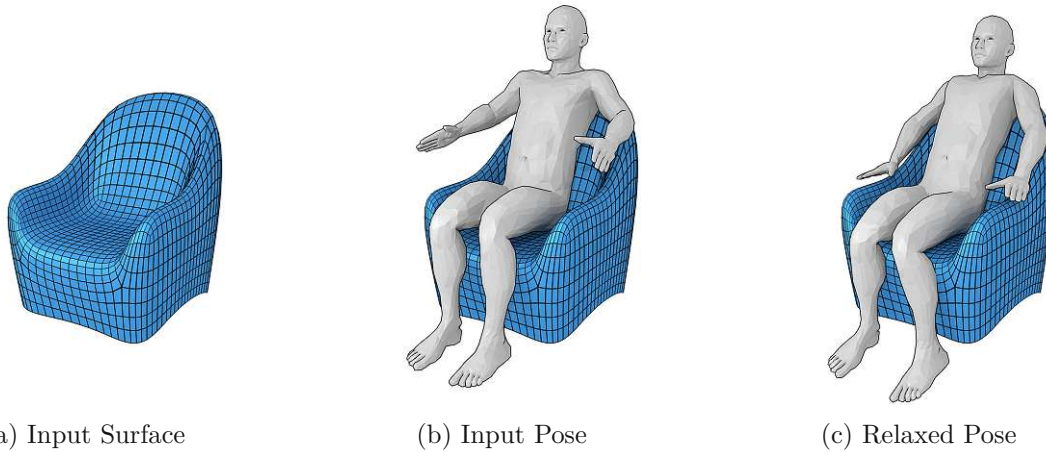


Figure 3.5: Pose relaxation using inverse kinematics. The designer can further relax the input poses in order to let them to adapt to the current surface (please refer to Section 3.5)

and

$$E^s(\boldsymbol{\theta}) = \boldsymbol{\theta} - \boldsymbol{\theta}^0.$$

The arctan  $(\cdot)$  function in  $E_i^p$  is used as a differentiable discrimination between penetration (values close to 1) and no penetration (values close to 0) of body and surface. We usually set the parameter  $a = 1000$  to limit the transition area between penetration and no penetration and  $b = 10$  to shift the inflection point of the function slightly into the penetrated area. We usually set the values  $\lambda_{d_p}$ ,  $\lambda_{d_t}$ ,  $\lambda_p$ , and  $\lambda_s$  to 0.2, 0.2, 0.5 and 0.1 respectively.

For an efficient optimization procedure we additionally compute the Jacobian matrices  $\mathbf{J}^{d_p}$ ,  $\mathbf{J}^{d_t}$ ,  $\mathbf{J}^p$ , and  $\mathbf{J}^s$  analytically, which we explain in Appendix A.2. Further we solve the non-linear optimization problem as proposed by [Bus04].

Figure 3.5 shows a result of pose relaxation performed on a given pose and a seat model. Please note that in this case the arms of the pose are relaxed to lean on the surface which allows to further explore the design with a new pose.

Note that it is possible for physically implausible poses to occur when the input pose is very different from the optimal pose. To prevent such poses, we use empirically chosen box constraints for the joint angle vector  $\boldsymbol{\theta}$ .

### 3.6 Design Process and Results

In this section we describe the design process of examples we have created in collaboration with design students.

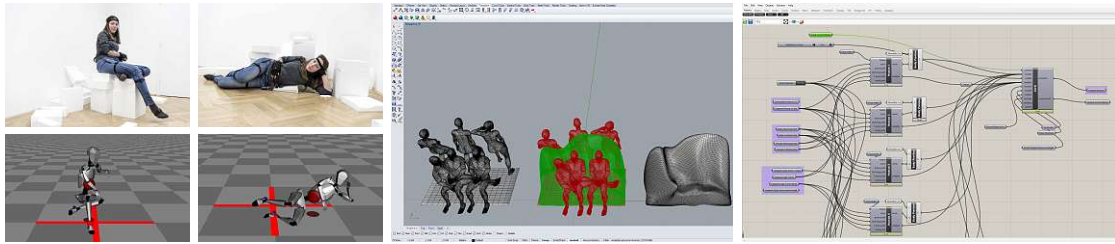


Figure 3.6: Design process. Left: recording of poses using a motion capture device. Center: interactive design using RHINOCEROS and GRASSHOPPER. Right: GRASSHOPPER canvas.

#### 3.6.1 Design Process

For the design process, we either allow the user to create own poses from a rigged mesh, or to import captured poses from a motion capture system. For our experiments, we created a database of 62 poses, which were captured by a PERCEPTION NEURON [Neu18] device. In this process, Styrofoam pieces of varying geometries were used as supports to make it easier to adopt different poses (cf. Fig. 3.6).

In the next step, we use the 3d-modeling software RHINOCEROS in combination with a custom GRASSHOPPER-plugin to design the surface (cf. 3.6, right). At this point, the designer can use the following operations to further control the process:

- Create a Catmull-Clark control mesh of an initial design or use a geometric primitive as starting point (e.g., flat surface).
- Fix selected vertices of the control polygon.
- Import poses from the database or create own poses.
- Place poses at desired location and orientations with respect to the initial design.
- Choose which body parts should be supported for each particular pose (torso, legs, arms, head, or their mutual combinations).

It is furthermore possible for the placed poses to overlap (like the design shown in Figure 3.1), although it is necessary to manually align them such that all of them can be supported well. Using these operations, the designer can create a scenario of the desired sitting landscape and let our solver create a new control mesh.

Having the new control mesh, further interactive editing steps are possible:

- Edit the control mesh by moving or fixing vertices or splitting the faces.
- Edit the scenario by moving, adding, or removing the poses.



- Relax the poses using inverse kinematics.

After each editing step (or a series of steps), our solver can generate a new control mesh and the final subdivision surface. The running times depend on the resolution of the control mesh, number of subdivision levels, and the number of poses, and are in practice in the range of one to several seconds.

### 3.6.2 Designer Response

We have asked 5 design students to test our modeling paradigm. Each student spent between 2-3 hours working with the tool and was asked to create a sitting scenario of her choice. The results of these sessions are depicted in Figures 3.9 and 3.10.

After the session, we asked the participants the following questions, which could be answered with four answers: poor (1), neutral (2), good (3), and very good (4).

1. How do you judge the general suitability of the system?
2. How do you judge the possibilities to control the outcome?
3. How do you judge the workflow simplification given by the tool?
4. How intuitive is the process?
5. How do you judge the quality of the achieved results?
6. Would you like to use the tool for your own project?

Q	P1	P2	P3	P4	P5	mean	median
1.	3	4	4	3	4	3.6	4
2.	2	3	3	2	3	2.6	3
3.	2	4	3	4	4	3.4	4
4.	3	4	2	3	2	2.8	3
5.	1	4	2	2	3	2.4	2
6.	2	3	4	2	4	3.0	3

Table 3.1: Results of the questionnaire given to the design students after using our system. Please refer to Section 3.6.2 for the particular questions.

The results shown in Table 3.1 allow to conclude that the method has been positively received in general. Please note the probands had only 2-3 hours for experimentation, which is truly a short period of time for creation of a design with a novel and unfamiliar tool.

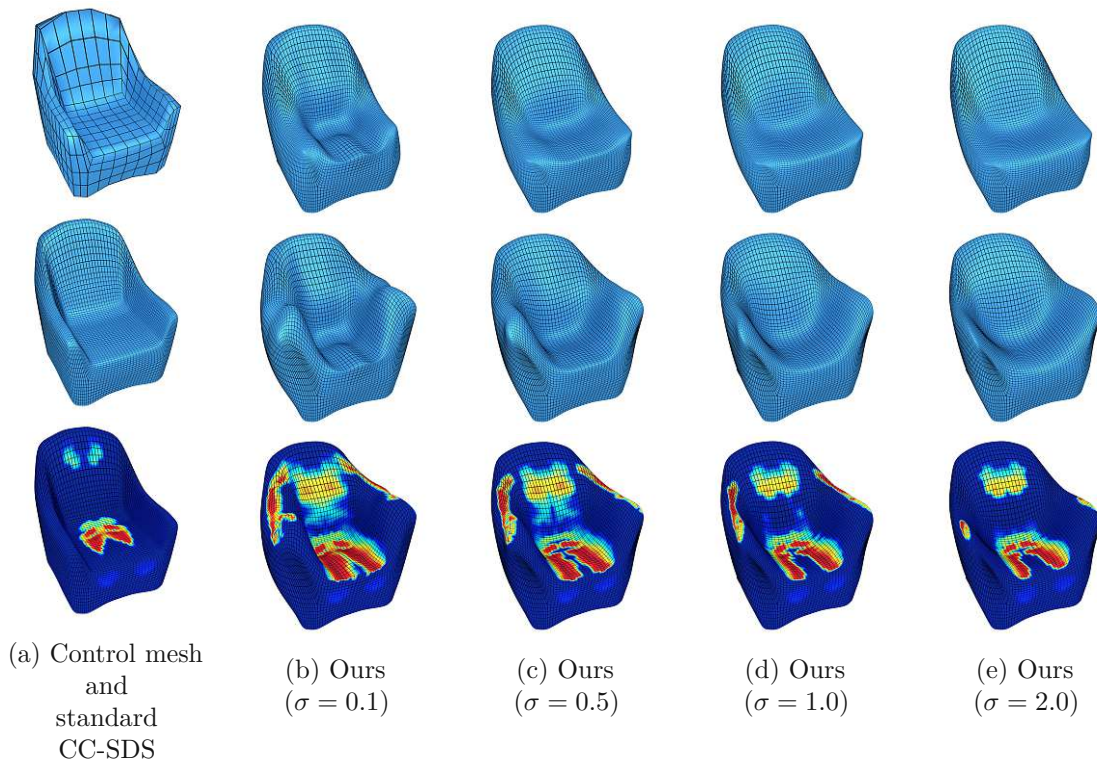


Figure 3.7: Variations using the control mesh shown in Fig. 3.7a created with varying values of the smoothing parameter  $\sigma$  which weighs the Laplacian operator. Using this parameter the designer can balance between the importance of the input body map and smoothness of the surface. Top row: arms of the input body have not been considered to be supported. Middle row: arms are supported. Bottom: contact area and pressure on the seat. Refer to Section 3.4 for more details.

## 3.7 Implementation and Fabrication

### 3.7.1 Implementation

Our algorithms are implemented in MATLAB using its optimization routines `lsqlin` for surface fitting (interior-point) and `lsqnonlin` (trust region reflective) for inverse kinematics. For the processing of the pose data obtained from the PERCEPTION NEURON motion capturing software, we use the MOCAP library [Law18]. We also make use of the geometry processing utilities provided by the GPTOOLBOX library [JO16]. As a front-end we use the 3d-modeling software RHINOCEROS with the node-based algorithmic modeling extension GRASSHOPPER. We developed our own GRASSHOPPER components that take input data from RHINOCEROS and feed it to a running MATLAB instance for use with our algorithms. The result then gets passed back to RHINOCEROS, enabling an interactive design process.

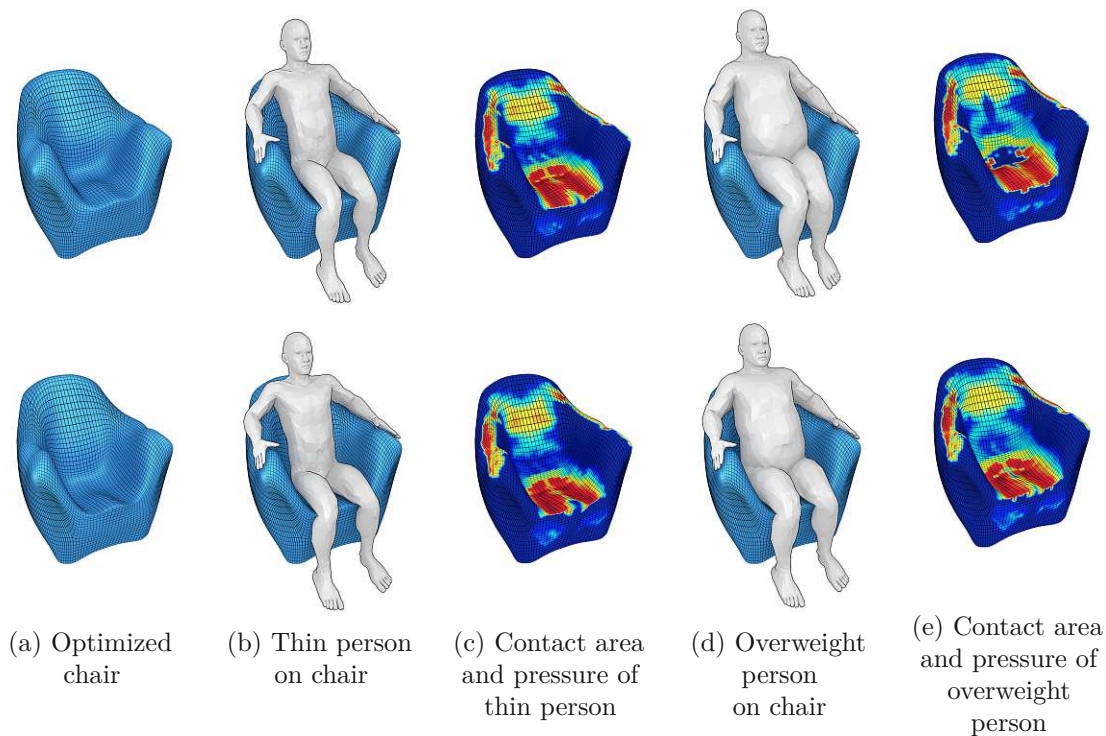


Figure 3.8: Chair optimized for 2 persons with different body types. Top: optimized for thin person. Bottom: optimized for overweight person.

### 3.7.2 Fabrication

As a result we have designed a multipurpose surface for three sitting poses (overlapping) and three standing poses (cf. Figure 3.1). The entire design process took about 6h, where the pose-capture session took about 3h and the following digital design and tuning session with Rhino took another 3h. The by far longest time was needed for the fabrication, where the preprocessing, preparation, and final milling time took about 4 days. For the milling from Styrofoam we have used the software SPRUTCAM 10 for the computation of the milling tool paths and a KUKA KR60 HA industrial robot arm for milling.

## 3.8 Discussion and Conclusion

### 3.8.1 Discussion

Our method does have a number of limitations. The major limitation is that we assume that the body surface is rigid, which is not the case in practice. However, without this assumption the computation of the pressure map would result in complex non-linear computation, which would be too complex for this kind of application. We think that

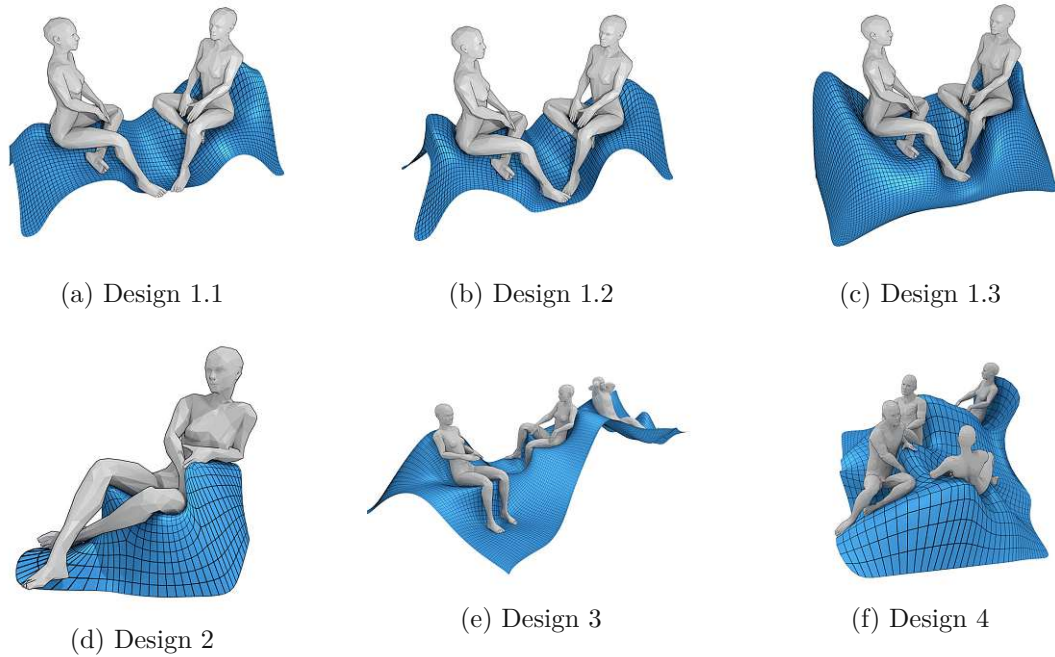
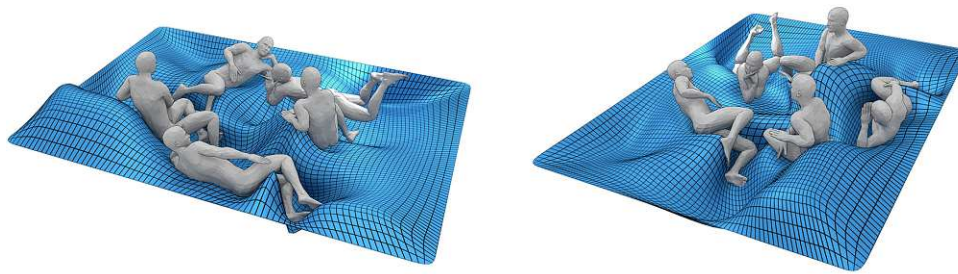


Figure 3.9: Several results created by design students using our method. The top row shows three design variations using the same input poses achieved by fixing different control vertices oder changing design parameters. Please refer to Section 5.4 for more details.

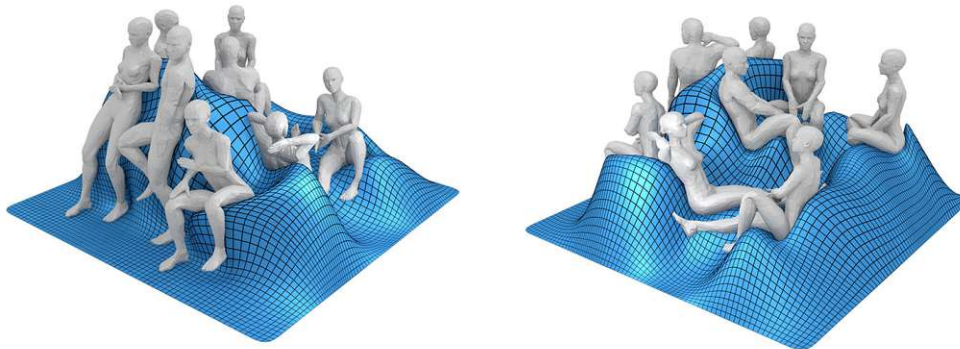
treating the input poses as articulated bodies consisting of rigid body segments connected by joints can lead to a more accurate pressure distribution, while still maintaining the interactivity of the design tool. This approach can be useful for applications that do have less strict requirements for low computation times, as we will discuss in Chapter 4, but for our application, the increased complexity outweighs the benefits as we are already able to generate interesting designs that are well optimized for the given input poses.

For the surface fitting, the assumption of rigidity of the body also poses a limitation, as the actual human body deforms when in contact with a surface. We account for this softness of the body by allowing a certain margin (about 3cm) for the connection between the body and the surface. While we use a type of collision detection (cf. Section 3.4), it still can happen that the body penetrates the surface if the resolution of the subdivision is lower than of the body (e.g., finger or toes). This, however, does not diminish the results. For inverse kinematics, we resolve it by using an  $\arctan(\cdot)$  function as a differentiable step-function in order to distinguish between penetration or not.

In the case of the multipurpose surface, the major technical limitation is that a small control mesh patch does not provide enough degrees of freedom in order to account for all poses. This can be approached with a higher resolution control patch, however,



(a) Design 5



(b) Design 6

Figure 3.10: Two additional designs of sitting landscapes for multiple persons.

this solution also has limitations and makes the design process more difficult. In the future, it would be interesting to provide a patch per pose and allow for stitching of several patches under maintenance of certain continuity (e.g.,  $C^1$ ), like proposed by Peters [Pet00]. Further, the incorporation of sharp edges and creases would be interesting as well. Finally, adaptive subdivision to improve fitting in areas where more freedom is needed could also be approached.

Finally, there is also a question of the body size and stature. While we use an average body for all of our designs, the designer is free to use different body types which can be easily created using the plugin of Bastioni [Bas18]. To optimize for various body types, it is possible to use multiple overlapping bodies as input to the surface fitting algorithm, but in order to make the optimized surface as comfortable as possible for all used body types, the body shapes need to be aligned manually. Another option would be to increase the smoothing operator  $\sigma$  in order to make the surface smoother, but at the same time less customized.

We examine how the results of the surface fitting can be influenced by the weight of the smoothing parameter  $\sigma$ . Figure 3.7 shows a number of designs generated from a simple

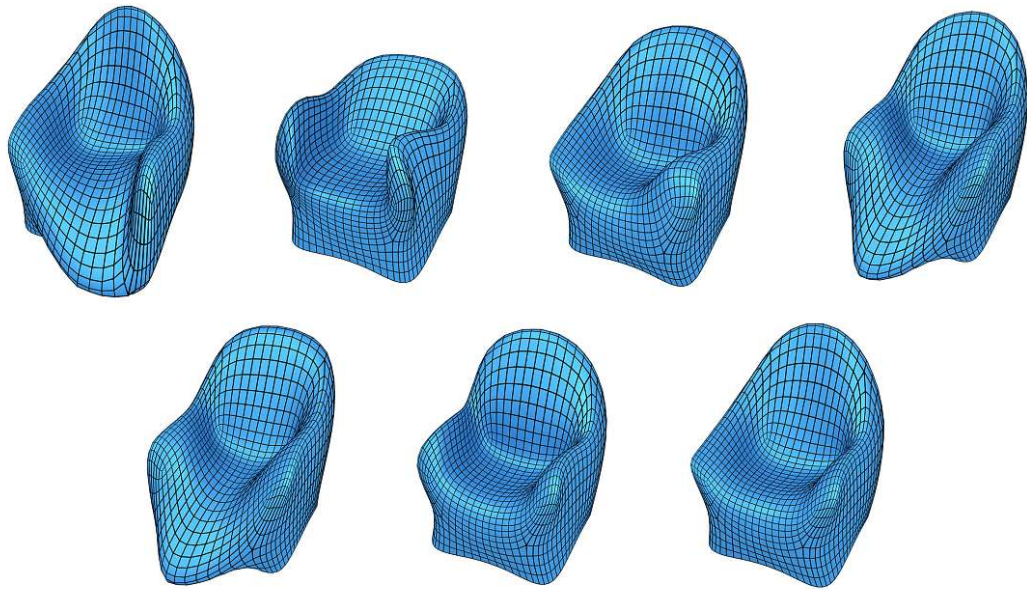


Figure 3.11: A series of designs using asymmetric input poses created with the control mesh shown in Fig 3.7a.

chair input mesh optimized for the sitting pose shown in Figure 3.5b. A small value for the smoothing parameter  $\sigma$  leads to a design that better supports the input pose, but could possibly lead to overfitting to the input body, making the design less comfortable for other body types.

Figure 3.8 shows how different body types influence the result of the optimized surface. As input we use the same chair mesh and pose as in Figure 3.5b, but with 2 different body types. The two optimized chairs shown on the left (Fig. 3.8a) may seem very similar at a glance, but greatly differ in how comfortable they are for persons of various body types. The top chair is optimized for a person with a thin build, while the bottom chair is optimized for an overweight person. We use our pose relaxation algorithm to see how a person of a certain body type fit into a chair optimized for the other body type. As can be seen in Fig. 3.8b and 3.8c, the thin person has no problem sitting on either chair. However, Fig. 3.8d shows that the overweight person needs to push the thighs together to fit into the chair optimized for the other person, and even then a large area of the buttocks does not have any contact with the seat, which can be seen in Fig. 3.8e.

#### 3.8.2 Conclusions and Future Work

In this chapter we have proposed a method for interactive design of body-supporting surfaces that is driven by the pose of the human body as well as the pressure distribution on the body's surface. Our method is intended to help designers create appropriate surfaces digitally without additional empirical design passes on the one hand, and to

ensure physical plausibility on the other hand. Further, it aims at interactive rates in the range of a few seconds.

Our main contribution is an interactive modeling system that utilizes captured body poses and computes an importance field that is proportional to the pressure distribution on the body for a given pose. This distribution indicates where the body should be supported in order to easily hold a particular pose, which is one of the measures of comfortable sitting.

We tested our system with design students and presented a number of results from these sessions. We also demonstrated a fabricated result. In the future, our method could serve as a basis for interactive design of various interesting furniture, for instance inflatable furniture, bean bags, as well as design furniture in general.

## Acknowledgments

We thank students who participated to the evaluation of the method: Ada Gulyamdzhis, Anna T. Pöll, Jasmin Plaikner, Jasmin Redl, Miriam Bachmann, and Tetyana Vovk. We thank Lukas Gersthofer for help with renderings. This research was funded by the Austrian Science Fund (FWF P27972-N31) and the Vienna Science and Technology Fund (WWTF ICT15-082).





# Automated Design of Seating Furniture

## 4.1 Introduction

In the modern life an increasing amount of time is spent in a seated position. The design of comfortable seating surfaces is an essential task in furniture design to ensure a person's well being. The optimal design of furniture in work environments has been well researched, especially in the area of ergonomics, where the general procedure is the application of design guidelines in the furniture design process. However, as most of these guidelines are restricted to specific environments, such as an office workplace, these insights cannot be applied to sitting in general. Therefore, specific situations require specialized solutions.

As form follows function, the design of furniture is greatly influenced by its intended use. For example, seating furniture designed for reading a book in a park requires a different form than furniture intended for watching a screen in a movie theater. The central task is therefore to find an optimal fit between a person in a specific situation and the seating surface.

However, there exists a trade off between general usability and individualization. On the two extremes of this spectrum one may find a simple flat board that anyone can sit on and the seat of a race car cockpit that is custom made to perfectly fit the driver. While the former is significantly lacking in comfort, it is easy to mass produce and a general solution that works for any individual. The latter may provide optimal comfort to its intended user, but the level of comfort is reduced significantly when used by someone else, making it unsuitable for mass production. Because of this, commercial furniture is usually designed with general usability and mass production in mind, offering less comfort for the individual. Our design approach based on poses and body types aims



Figure 4.1: Left: seating surface generated by our automated algorithm and suited to support the input pose . Right: the surface fitting algorithm described in Chapter 3 applied to the generated surface.

for increased individualization compared to commercial furniture while still maintaining general usability.

Providing seating surfaces for arbitrary seating poses is a difficult task and often requires substantial manual design effort to guide the computational design process. Our primary motivation for this chapter is to provide a furniture design solution capable of supporting a person in a specified sitting pose, while eliminating most manual design effort.

Furthermore, for optimal seating solutions, knowledge about a person's comfort is required. As human bodies exist in a variety of shapes and sizes, comfort is a subjective quantity. The most reliable way to determine the comfort of a seating surface is subjective evaluation, after producing a physical prototype. As this method is infeasible for most digital design frameworks, an objective definition of comfort is required.

The central goal of this chapter is to propose a solution for computational pose-inspired furniture design. Therefore we develop a software framework with the goal of automated generation of seating surface models for specific applications.

In order to create seating surfaces for specific situations, we aim to utilize human body shapes and sitting poses as input for the furniture design process. With the development of the design framework we aim to fulfill three major goals: (i) The created models must provide a high level of comfort for a person in a specific pose. The primary objective is to create a seating surface which optimally supports a given body shape for a specific pose. (ii) The proposed framework must have the ability to support a large variety of different sitting poses as well as body shapes. While a single created model is made for a specific body shape and pose, the algorithm must be capable of processing a large variety of sitting poses and human body shapes. (iii) We aim to achieve a high level of visual quality and create furniture models that are aesthetically pleasing.

In order to design a framework that fulfills these goals, we propose a fully automatic pose driven design approach. Our method is meant for computer-aided design of personalized furniture that can be used by inexperienced users to quickly create unique designs or by professionals for creating initial designs that can be modified and enhanced in further steps of the design process.

Our contributions are therefore the following:

- We propose a novel computational model for the approximation of the comfort of sitting in a given pose, based on both pressure distribution of the body on the seat and on the moments (torques) acting on the limbs of the body. Our model is driven by physical assumptions and extends previously proposed models. Nonetheless, it is simplified to a system of linear equations in order to account for interactive rates. Details of this model are described in Sec. 4.2.
- Moreover, we propose a generic furniture model which delivers a control mesh which can be used for further refinement, e.g., using subdivision surfaces. Our model is capable of supporting a large variety of poses and body shapes, while fulfilling the functional and visual requirements. Our template seating mesh can automatically support individual parts of a person's body.

In the following section we review related work, and in Section 5.4 we present and compare our results. Finally, we discuss and conclude the work in Sections 4.5.1 and 4.5.2.

## 4.2 Computational Model of Sitting

In this section we propose our simplified physical model of sitting for the computation of pressure distribution, the moments (torques) acting on the joints as well as friction forces acting on the body. This pressure distribution is used as an importance map in the furniture synthesis stage (see Section 4.3) to indicate where the human body needs to be supported. In Chapter 3 we proposed a similar simplified computation model, however, this extended method has three advantages:

1. Our extended model consists of individual body segments instead of a single rigid body, allowing us to consider a more realistic distribution of body mass, as well as the moments acting on the joints which are caused by the transfer of forces between body segments.
2. Using this model, we can also compute friction forces which are not available in the simple approach.
3. Finally, our extended algorithm yields physical pressure values instead of only a relative distribution, which we show to be in realistic range by comparing to FEM simulation.

### 4.2.1 Human Body Model

We propose a novel human body model that combines a skeleton with a surface that allows us to compute the moments acting on the joints and the pressure distribution on the surface of the body.

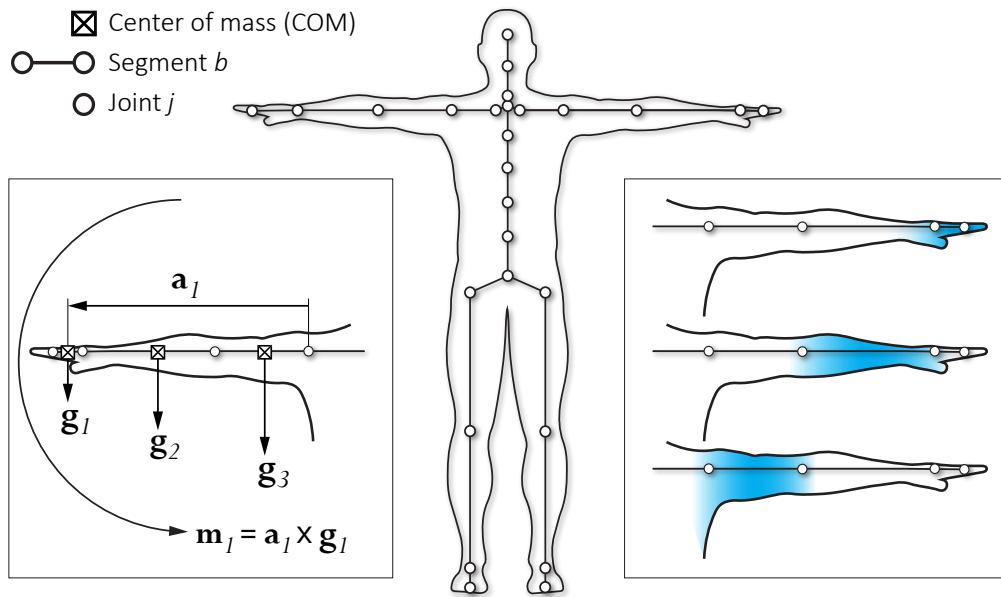


Figure 4.2: Left: a moment  $\mathbf{m}_l$  as a cross product of the moment arm  $\mathbf{a}_l$  and the force  $\mathbf{g}_l$ . Body segments have anatomical values (e.g.,  $\mathbf{g}_l$ ) assigned from [PEA83]. Middle: a link-segment-skeleton with 21 segments and a polygonal surface mesh. Right: The mesh is rigged using linear blend skinning [MTLT89].

### Skeleton Model

The skeleton is modeled using a link-segment-model with 21 segments as depicted in Figure 4.2. Such a model consists of segments that represent parts of the human body which are connected by joints that allow movement of the segments with varying rotational degrees of freedom. Each segment has its own mass concentrated at the center of mass (COM) and can be influenced by external forces such as gravity or contact with other surfaces. The mass and the locations of the center of mass of each segment are based on the data by Plagenhoef et al. [PEA83], which were determined empirically on experiments with human cadavers.

The joints themselves are assumed to have no mass and also to not be affected by external forces. They can, however, transfer forces and moments from one segment to another. We model this as two opposing forces (or moments) acting on the joint, one for each segment linked by the joint (cf. Fig. 4.4 for detailed depiction).

Segment-link-models are commonly used in biomechanics to examine the moments acting at joints during certain actions or movements [Hal95, RCH<sup>+</sup>04]. Since we use the same data set of poses utilized in Chapter 3, which was created using the Perception Neuron motion capturing system [Neu18], our segment-link model contains 21 segments. Please refer to Figures 4.2 and 4.4 for a depiction.

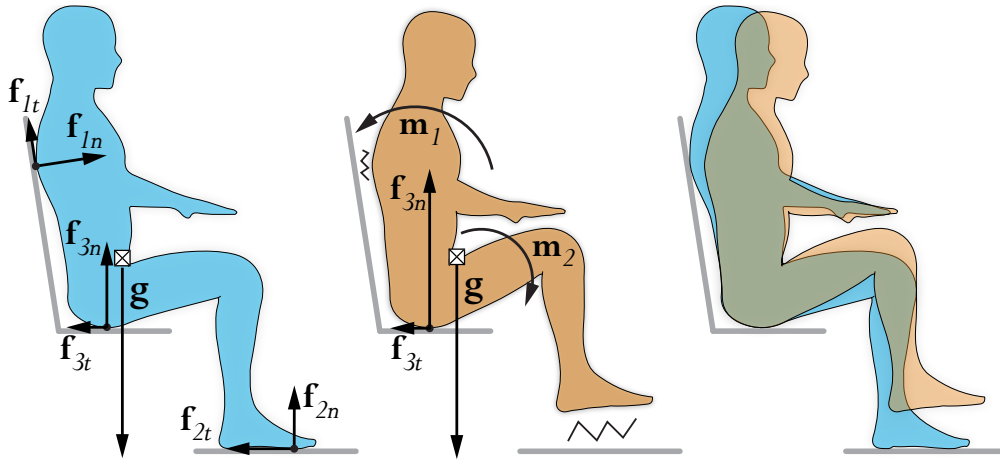


Figure 4.3: Physics of sitting: if contact with a support surface is given on the buttocks, back, and feet, the moments of the body are minimized and the forces are in equilibrium. If the contact on the feet is lost, the contact to the back is lost automatically due to the missing friction force on the feet.

### Skinning

We register the skeleton with a human body model given by a triangle mesh. For the generation of body meshes we have used the software provided by Bastioni [Bas18] which allows the generation of human bodies with varying parameters, like gender, mass, size, stature, etc. We rig the mesh with the skeleton using linear blend skinning [MTLT89], in particular all vertices are defined by their weighted linear combinations:

$$\mathbf{v} = \sum_b \alpha_{b,v} \mathbf{T}_b \mathbf{v}^0,$$

where  $\mathbf{v}^0$  are the initial and  $\mathbf{v}$  are the new vertex positions respective,  $\alpha_{b,v}$  are the weights which associate the vertex  $v$  to the segment  $b$ , and  $\mathbf{T}_b$  are the transformations of the assigned segments  $b$ . We use the algorithm of [BP07] implemented in BLENDER.

We further use the weights of the skinning to propagate forces from the segments to the surface vertices and vice versa (cf. Figure 4.2 and Figure 4.4).

### Friction Model

In the mechanics of sitting, friction plays an important role. Consider the example shown in Figure 4.3, left, where a body has three contact points: on the buttocks, the back, and the feet. The tangential reaction force  $\mathbf{f}_{1t}$  that is supporting the back is dependent on the normal force  $\mathbf{f}_{1n}$  at the same location. This normal force can only exist due to an opposing force  $\mathbf{f}_{2t}$  existing at the feet since all forces must sum to zero to maintain equilibrium. Therefore, if we lose the contact of the feet to the ground (e.g., by lifting the

legs), the force  $\mathbf{f}_{2t}$  disappears and we also lose contact on the back and the reaction forces acting there unless an additional force is introduced, for example by pressing the thighs against the seat which requires significantly more muscle activity. In consequence, the back is no longer supported and the overall contact area becomes much smaller, resulting in a higher force ( $\mathbf{f}_3$ ) on the remaining contact points. Additionally, higher moments ( $\mathbf{m}_1$ ,  $\mathbf{m}_2$ ) act on the joints, requiring more muscle forces to maintain the pose (cf. Figure 4.3, center).

We use the Coulomb model in which the frictional component of a reaction force depends only linearly on the normal component of the reaction force (refer to Figure 4.4 and to Eq. 4.1 later on). We choose this simplified model, since due to our assumptions, the by far biggest force is the gravity which implies that forces in any other direction tend to be much smaller.

### 4.2.2 Reaction Forces

Our goal is now to find a physically plausible distribution of reaction forces that supports the human body with as little need to use additional muscle forces to maintain its current pose as possible. Usually, such distribution would be found using a sophisticated finite elements simulation which is very time consuming.

Since our goal is to achieve interactive rates, we propose a model where we assume the human composed of rigid segments combined by joints, where the surface vertices are related to the segments of the bodies surface by linear combinations. This allows us to formulate it as a Pareto-optimization problem where we balance the minimization of the moments acting in the body with the uniformity of the distribution of the reaction forces.

In this section we first describe how we estimate the friction and normal forces on each vertex of the surface, and further we describe the details of the linear optimization problem. Finally, we compare our results to a rigid-body FEM simulation in order to validate our results.

#### Local Reaction Weights

In order to compute the optimal reaction forces for the entire system, we first introduce the local reaction and friction force model, which we utilize for the derivation of *local reaction weights*. In essence, we compute the maximum reaction forces that can occur if a local force  $\mathbf{f} = (0, -1, 0)^T$  acts on an isolated vertex. We first split  $\mathbf{f}$  into its normal component  $\mathbf{f}_n$  along the surface normal and its tangential components  $\mathbf{f}_{t_1}$  and  $\mathbf{f}_{t_2}$ . Making use of the well-known friction pyramid of the Coulomb model [Pop10], we have

$$\mathbf{r}_n = -\mathbf{f}_n, \quad \mathbf{r}_{t_1} = -\frac{\mathbf{f}_{t_1}}{\|\mathbf{f}_{t_1}\|} \min(\|\mathbf{f}_{t_1}\|, \mu\|\mathbf{f}_n\|), \quad (4.1)$$

and  $\mathbf{r}_{t_2}$  defined analogically, with the friction coefficient  $\mu \geq 0$ . The total reaction force is then  $\mathbf{r} = \mathbf{r}_n + \mathbf{r}_{t_1} + \mathbf{r}_{t_2}$  (cf. Figure 4.4, left box).

In other words, the magnitude of the friction force must be smaller or equal to the magnitude of the normal force multiplied with the friction coefficient  $\mu$ , which depends mainly on the roughness of the surface material, hence, in our experiment we use  $\mu = 0.5$  which is a common default value if the material is not known.

Since the distribution of the reaction forces on the body depends on the overall forces acting on the system, which are not known in advance, looking only at each vertex individually is not sufficient. But we can use this information to introduce a weight vector  $\mathbf{w} = (w_n, w_{t_1}, w_{t_2})^T$  per vertex with

$$w_n = \frac{1}{\|\mathbf{r}\|}, \quad w_{t_1} = \frac{1}{\|\mathbf{r}_n + \mathbf{r}_{t_1}\|}, \quad w_{t_2} = \frac{1}{\|\mathbf{r}_n + \mathbf{r}_{t_2}\|}, \quad (4.2)$$

which serves us later to indicate the actual contribution of each individual reaction force to their global distribution during the optimization (cf. Eq. 4.8).

### Computation of Reaction Forces

Please refer to Figure 4.4 for a depiction of the components.

First, we compose the vector  $\mathbf{x}$  of unknowns of the following physical entities:

- $\mathbf{f}_{b,j}$  ... the forces acting on the joints  $j$  in each body segment  $b$ . There are 2 such forces per body segment, except for the hands, feet, and head since they are connected to only 1 joint,
- $\mathbf{m}_{b,j}$  ... the moments acting on the joints  $j$  in each body segment  $b$ . Again there are 2 such moments per body segment, except for the hands, feet, and head since they are connected to only 1 joint,
- $\mathbf{r}_v$  ... the reaction forces acting at each body vertex  $\mathbf{v}$ , caused by contact with an external surface.

The vector of unknowns  $\mathbf{x}$  is the  $3(n_{\mathcal{F}} + n_{\mathcal{M}} + n_{\mathcal{R}})$  column vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{f}_{b,j} \\ \mathbf{m}_{b,j} \\ \mathbf{r}_v \end{bmatrix},$$

where  $n_{\mathcal{F}}, n_{\mathcal{M}}, n_{\mathcal{R}}$  denote the cardinality of the sets for joint forces  $\mathcal{F}$ , moments  $\mathcal{M}$ , and reaction forces  $\mathcal{R}$  respective.

Our main constraint is that the human body must be in static equilibrium, meaning that it must be physically able to maintain its current pose through contact forces, friction, and acting moments (i.e., muscle strength), so that there is no translational or rotational movement of any segment. According to the equations of motion [GPS02], a body is in equilibrium if the sum of all acting forces and moments sum to 0. Applied to our link-segment-model, this includes the following forces:

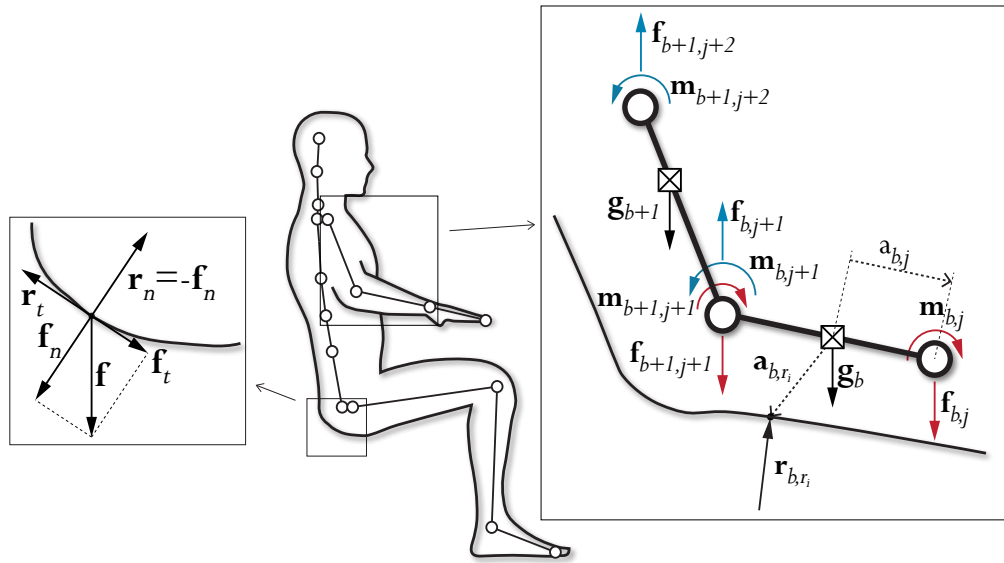


Figure 4.4: Computational human body model. Left: simplified friction model, right: free-body diagram of the skeleton model. Please refer to Section 4.2.2 for the details.

- $\mathbf{g}_b \dots$  the gravity acting on the center of mass (COM) of the body segment  $b$ ,
- $\mathbf{r}_v \dots$  the reaction forces at each vertex of the body segment caused by contact with an external surface,
- $\mathbf{f}_{b,j} \dots$  the forces caused by other body segments transmitted through the joints  $j$ ,

and the following moments:

- $\mathbf{a}_{b,j} \times \mathbf{f}_{b,j} \dots$  the moments acting on the COM of body segment  $b$  caused by the forces from other body segments transmitted through joint  $j$ , with the moment arm  $\mathbf{a}_{b,j}$  being the vector pointing from the COM to joint  $j$ .
- $\mathbf{a}_{b,r} \times \mathbf{r}_v \dots$  the moment acting on the COM of body segment  $b$  caused by the reaction force through contact with an external surface, with the moment arm  $\mathbf{a}_{b,r}$  being the vector pointing from the COM to the contact point.
- $\mathbf{m}_{b,j} \dots$  the moments caused by other body segments transmitted through the joints  $j$ .

Please note that reaction forces  $\mathbf{r}_v$  at the vertices  $v$  are connected to the body segments  $b$  by the linear blend skinning weights  $\alpha_{b,v}$ . This leads to the following constraints for



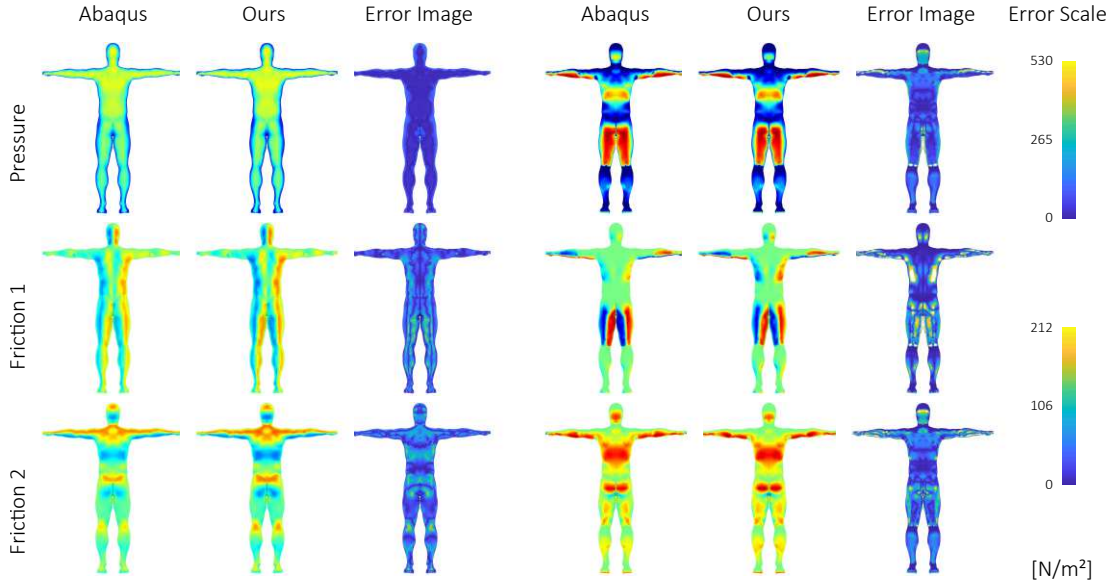


Figure 4.5: Comparison to the FEM (cf. Section 4.2.2). Left: results of a lying T-pose. Right: a sitting pose. Please note that we plot the pressure and shear distributions of the sitting pose on a T-pose mesh for better visualization purpose.

each body part  $b$ :

$$\sum_{j \in \mathcal{J}_b} \mathbf{f}_{b,j} + \sum_{v \in \mathcal{V}_b} (\alpha_{b,v} \mathbf{r}_v) - \mathbf{g}_b = \mathbf{0}, \quad (4.3)$$

$$\sum_{j \in \mathcal{J}_b} \mathbf{a}_{b,j} \times \mathbf{f}_{b,j} + \sum_{v \in \mathcal{V}_b} \mathbf{a}_{b,r} \times (\alpha_{b,v} \mathbf{r}_v) + \mathbf{m}_{b,j} = \mathbf{0}, \quad (4.4)$$

with  $\mathcal{J}_b$  being the set of joints connected to body segment  $b$  and  $\mathcal{V}_b$  being the set of vertices of body segment  $b$ . Naturally, the sum of forces, as well as the sum of moments, acting on a joint must also equal 0, i.e.:

$$\sum_{b \in \mathcal{B}_j} \mathbf{f}_{b,j} = \mathbf{0} \quad \text{and} \quad \sum_{b \in \mathcal{B}_j} \mathbf{m}_{b,j} = \mathbf{0}, \quad (4.5)$$

with  $\mathcal{B}_j$  being the set of body segments connected to joint  $j$ .

These constraints can be formulated as a system of linear equations  $\mathbf{C}\mathbf{x} = \mathbf{z}$ . The matrix  $\mathbf{C}$  is a  $(6n_{\mathcal{B}} + 6n_{\mathcal{J}}) \times 3(n_{\mathcal{F}} + n_{\mathcal{M}} + n_{\mathcal{R}})$  matrix—6 rows for each body segment and each joint (3 for the forces and 3 for the moments), as well as 3 columns for each unknown force, moment, and reaction force in a body segment.

To ensure that the resulting reaction forces do not point out of the body (which would be physically equivalent to gluing the body to a surface), we also require inequality

constraints

$$-r_y \leq 0, \quad (4.6)$$

with  $y$  being the up-direction of the global coordinate system.

Finally, since the weights computed in Eq. 4.2 are used in the objective function and are therefore soft-constraints, we additionally restrict the magnitudes of the friction forces based on the normal force with hard constraints.

To do so, we consider the reaction force vector  $\bar{\mathbf{r}}$  in the tangent space of vertex  $\mathbf{v}$ , with the first coordinate being the normal force and the second and third components being the friction forces, and limit the magnitude of the latter in relation to the normal force using

$$\bar{r}_y \leq \mu \bar{r}_x \quad \text{and} \quad \bar{r}_z \leq \mu \bar{r}_x. \quad (4.7)$$

We denote the matrix containing these inequality constraints as  $\mathbf{D}$ .

We can now formulate the objective function as

$$E_{\text{pres}} = \sum_{i=1}^{n_{\mathcal{M}}} \|\mathbf{m}_i\|^2 + \lambda \sum_{i=1}^{n_{\mathcal{V}}} \frac{1}{A_i} \|\mathbf{w}_i \circ \bar{\mathbf{r}}_i\|^2, \quad (4.8)$$

where  $\circ$  denotes the Schur-product,  $\mathbf{w}_i$  are the reaction weights (cf. Eq. 4.2) and  $\bar{\mathbf{r}}$  is the reaction force at the vertex  $\mathbf{v}_i$  in its tangent space. Note, that we need to divide the reaction force by the (Voronoi) area  $A_i$  of each vertex since we want the forces to be distributed equally over the surface regardless of mesh resolution.

Minimization of the function in Eq. 4.8 with constraints in Eq. 4.3, 4.4, 4.5, 4.6, and 4.7 leads to a system of linear equations with equality and inequality constraints, which we formulate in matrix form as

$$\begin{aligned} \min_x \quad & \|\mathbf{A}\mathbf{x}\|^2 \\ \text{s. t.} \quad & \mathbf{C}\mathbf{x} = \mathbf{z}, \\ & \mathbf{D}\mathbf{x} \leq \mathbf{0} \end{aligned}$$

and solve it using MATLAB's `lsqlin` function. The details of how the matrices  $\mathbf{A}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  are constructed can be found in Appendix A.3.

The free parameter we introduce in Eq. 4.8 is the value of  $\lambda$ . Intuitively, it is a weight which allows to balance between the terms which minimize the moments in the body and which distribute the reaction forces on the surface.

Physically, we can interpret this parameter as the 'stiffness' of the joints. If it is 0, no muscle force can be expended to maintain the pose. If it is infinite, the entire human body can be treated as completely rigid. Realistically, we cannot set the parameter to 0 because we only have a finite number of reaction forces acting at predetermined locations, making it either impossible to fulfill the equilibrium constraints or resulting in a physically implausible solution for most poses. In empirical experiments, we determined a default value of  $\lambda = 0.013$ , which we have further used in our applications.

In order to actually compute the pressure distribution of a pose which is used as an importance map in the next step of our approach, we need to know which vertices of the body surface are in contact with the support surface. For this we proceed like in the approach described in Chapter 3 and assume that the given body is supported everywhere, meaning that we consider every vertex of the body surface to be in contact.

### Comparison to Finite Elements Simulation

In order to evaluate our computational model, we compare it to a FEM simulation using the professional physical simulation software ABAQUS [Smi09]. We select 2 poses for this purpose—a lying pose and a sitting pose. For each pose, we create 2 parts in ABAQUS, one being the body with the geometry of the original mesh, the other being a shell generated from the original geometry which serves as the contact surface. To create this shell, we first include all mesh faces whose normal is not perpendicular or opposite of the gravity direction, and then manually reduce this set by deleting isolated faces or faces where we do not want to support the body (e.g. under the armpits).

We create a volume mesh of the body using the ABAQUS meshing algorithm such that the body consists of roughly equally sized tetrahedrons. We use the same element size to subdivide the shell such that the surfaces of body and shell are still perfectly aligned. We assign both body and shell a Young’s Modulus of  $2.1e + 18$  and Poisson’s Ratio of 0.3, thus making both parts close to rigid. The shell is completely locked in place by boundary constraints, while the body is moved downward by forces totaling  $735N$  (roughly equivalent to a body weight of  $75kg$ ). The forces are applied per vertex (both on the surface and the inside of the body), their distribution computed from the weight of each body segment and the skinning weights that determine which vertex belongs to which body part (for each inner vertex we simply copy the weights of the closest surface vertex).

To model the contact between body and shell we use a linear pressure-overclosure relationship with contact stiffness of  $8e + 12$ . The tangential contact behavior is modeled with an isotropic friction coefficient of  $\mu = 0.5$  and an elastic slip of  $1e - 10$ . We determined these parameter values empirically, as other values would often result in unstable contact conditions and a physically implausible pressure distribution with immense pressure peaks at some isolated vertices and no contact at all at other vertices.

In our model, we select the set of vertices at which reaction forces are computed by choosing those vertices of the body mesh that are also included in the corresponding contact surface shell to make sure that the contact surface is the same in both methods. We furthermore do not optimize for the joint moments ( $\lambda = \infty$ ), which is equivalent to making the body completely rigid, as is also the case in the ABAQUS simulation.

For the lying pose, the ABAQUS simulation took a total of 32 minutes and 5 seconds (18 minutes and 12 seconds for preprocessing and 13 minutes and 53 seconds for actual simulation), while our system takes 1.6 seconds on average. For the sitting pose, the ABAQUS simulation took a total of 141 minutes and 50 seconds (78 minutes and 8 seconds

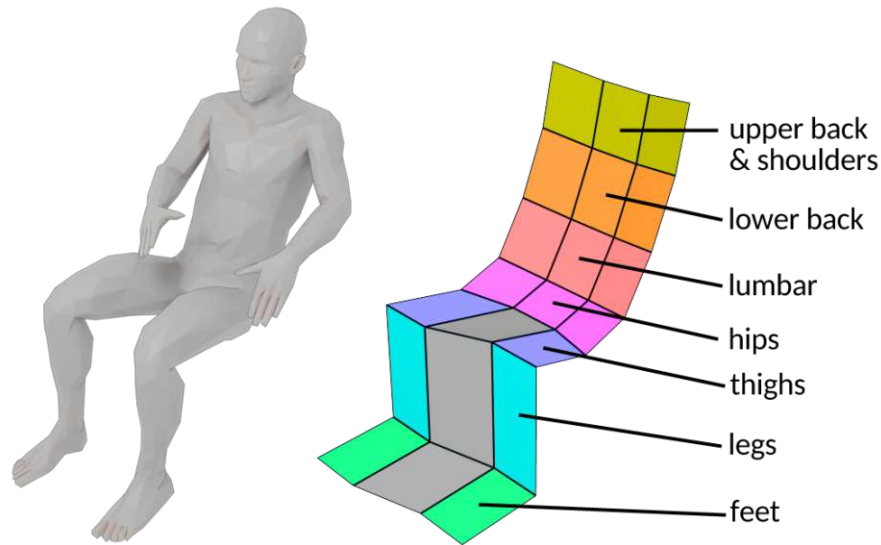


Figure 4.6: Body part mapping in the advanced template model. The rows of the model are mapped to individual body parts. Within a row, the segment in each column is mapped to a subset of the corresponding body vertices. The leg segments are mapped independently to the corresponding body parts.

for preprocessing and 63 minutes and 42 seconds for actual simulation), while our system takes 1.5 seconds on average. The results of both methods can be compared in Figure 4.5. Note that we use the same color scale for the visualization of the results of both methods, but different scales for the pressure and shear values.

### 4.3 Furniture Synthesis

The general goal of this stage is to automatically create a control mesh for the seating surface that closely fits a human body in a specific pose. The template model utilizes a hierarchy of non-planar quads for this purpose, chosen for simplicity as well as suitability for the task. As a secondary optimization goal, we aim to produce a visually pleasing piece of furniture. Therefore, we impose rough guidelines on the geometric shape of the seating surface regarding planarity and regularity ([LXW<sup>+</sup>11, ZSW10]).

#### 4.3.1 Template Model

The general design concept for the template model is to find a suitable structure of quadrilateral faces which can be fit to the human body in a specific pose according to the comfort measures (represented by a supplied importance map) under the defined constraints. For the proposed framework we decided on using a 3x7 grid of faces for the main body shape, excluding the person's arms and head, which are treated separately.

Going forward, we refer to the faces along the height direction of the body as *rows* and the faces along the width as *columns*.

Figure 4.6 shows the assignment of the template faces to each body part. The legs are each mapped to an individual column of 3 faces (foot, shank and thigh), while each part of the upper body (hips, lumbar, lower back, upper back) is mapped to a row of 3 faces. A person's arms are supported by additional faces which are added in a later stage in the algorithm.

Since each pose is determined by a 66 parameter vector, the space of possible poses is vast. This makes it impossible to support all possible poses using a template model with a predefined topology. Problems arise when the projections of the supported body segments onto the ground plane intersect. For a pose to be supported without special treatment, we therefore require that the shortest line between any supported vertex and the ground plane does not intersect the body geometry. Since this requirement significantly limits the space of valid poses, we detect and handle a number of special cases: crossed legs, upper body leaning forward and arms positioned above the body (see Section 4.3.1).

### Surface Fitting and Mesh Generation

The process of fitting the model is as follows: For each face in the grid, a plane is fitted to the shape of the respective body parts. The fitting algorithm utilizes the geometry of a human body mesh transformed into specific sitting pose as well as its computed importance map, indicating which vertices are most important to support to reach optimal comfort. As our model consists of 21 free floating planes, hierarchical constraints are introduced to the fitting process to prevent error cases and maintain the general structure of the model.

The mesh generation stage consists of four steps. In the first step, we create the two 1x3 strips of faces that support each individual leg. These individual leg supports are then connected by another 1x3 strip in the second step. In the third step, the middle 1x4 strip of the upper body support is generated. Finally, we complete the mesh generation stage by creating the two outer 1x4 strips supporting the upper body.

In the first and third step, we use the RANSAC [FB81] algorithm to find the plane that best supports a given body segment while also satisfying the structural constraints of the model hierarchy. A candidate plane is defined by randomly choosing 3 vertices of the body segment with probabilities based on their importance. If the candidate plane intersects with the vertices of an adjacent body segment, it is discarded outright. Otherwise we define a local coordinate system on the candidate plane using the unit vectors  $\mathbf{n}^P$ ,  $\mathbf{d}^{up}$  and  $\mathbf{d}^{side}$  (Fig. 4.7, left).  $\mathbf{n}^P$  is simply the normal vector of the candidate plane.  $\mathbf{d}^{up}$  is constructed by taking the direction  $\mathbf{d}^{body}$  of the skeleton bone corresponding to the given body segment and projecting it onto the plane. Finally, we have  $\mathbf{d}^{side} = \mathbf{d}^{up} \times \mathbf{n}^P$ . Additionally, we consider the direction  $\mathbf{d}^{ir}$  of the line of intersection  $\mathbf{l}^{ir}$  between the candidate plane and the fitted plane of the previous row of the mesh template.

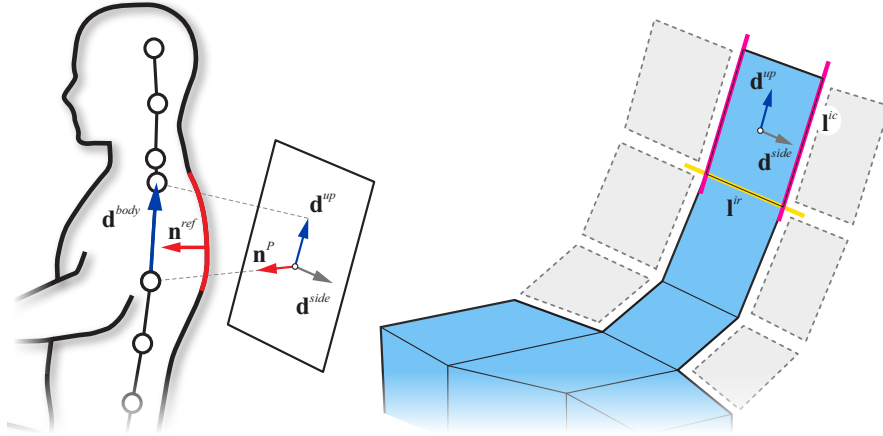


Figure 4.7: Plane fitting and mesh generation process. Left: A plane is fitted to the body segment marked in red. The vectors show the local coordinate system on the plane consisting of plane normal vector  $\mathbf{n}^P$ , the projection of skeleton bone direction  $\mathbf{d}^{body}$  onto the plane  $\mathbf{d}^{up}$  and the vector  $\mathbf{d}^{side}$  which is orthogonal to both. Right: Template mesh during the third step of the mesh generation process. Row and column intersection lines  $\mathbf{l}^{ir}$  and  $\mathbf{l}^{ic}$  are shown in yellow and magenta respectively.

To evaluate the quality of the candidate plane, we introduce two penalties. The penalty

$$p^r = 1 - \min \left( \frac{4}{\pi} \cdot \left| \text{atan2} \left( \left\| \mathbf{n}^P \times \mathbf{n}^{ref} \right\|, \langle \mathbf{n}^P, \mathbf{n}^{ref} \rangle \right) \right|, 1 \right) \quad (4.9)$$

penalizes planes with a normal vector  $\mathbf{n}^P$  deviating from the reference vector  $\mathbf{n}^{ref}$  which is computed by applying PCA on the vertices of the given body segment. Furthermore, for the upper body, the penalty

$$p^d = 1 - \frac{1}{m^\alpha} \min \left( \left| \arccos \left( \frac{\langle \mathbf{d}^{side}, \mathbf{d}^{ir} \rangle}{\|\mathbf{d}^{side}\| \|\mathbf{d}^{ir}\|} \right) \right|, m^\alpha \right) \quad (4.10)$$

ensures that the direction of the line of intersection between adjacent rows of the template  $\mathbf{d}^{ir}$  conforms to the body geometry. Planes with an intersection direction that deviates from  $\mathbf{d}^{side}$  by more than a chosen value  $m^\alpha$  are penalized (cf. Fig. 4.7, right).

The total quality of a plane is then given by

$$w_P = \sum_{\mathbf{v} \in \mathbf{V}_P} w(\mathbf{v}) (1 - \lambda^r + p^r \lambda^r) (1 - \lambda^d + p^d \lambda^d) \quad (4.11)$$

with  $\mathbf{V}_P$  being the set of body segment vertices within a set distance of the plane,  $w(\mathbf{v})$  being the importance of vertex  $\mathbf{v}$  based on the pressure value, and the weights  $\lambda^r$  and  $\lambda^d$  to set the influence of each penalty term.

To create the actual mesh geometry, we first estimate the width of the faces for the already fitted planes by taking a line with direction  $\mathbf{d}^{up}$  and offsetting it in the positive and negative direction of  $\mathbf{d}^{side}$  by half (or in the case of the upper body less than half) the width of the body segment to obtain the column intersection lines  $\mathbf{l}^{ic}$  (cf. Fig. 4.7, right). By intersecting the row and column intersection lines  $\mathbf{l}^{ir}$  and  $\mathbf{l}^{ic}$ , we obtain estimates for the corner vertices of the current face. The final coordinates of the corner vertices are obtained by averaging the positions of the estimated vertices of adjacent faces.

In the final step of the mesh generation process, the outer column faces of the template are determined again via surface fitting while utilizing the inner column segments as hard constraints, i.e., the two vertices incident to both the inner face and outer face of the row are fixed, so only one additional vertex is necessary to construct a plane. We iterate over all relevant vertices of the body segment to find the plane with the best support, using Eq. 4.10 as a quality measure. However, if the angle between the new row intersection line and the previous row intersection line is too large, the face in the previous row could degenerate into a triangle. In such cases we reject the plane. Once a suitable plane has been found, the two inner vertices are then shifted along the corresponding intersection lines by a set distance to create the remaining two vertices of the outer face. The resulting geometry is a connected 3x7 grid of non-planar quadrilateral faces fitted to the given body shape.

### Special Case Handling

The proposed algorithm is capable of providing suitable solutions for basic sitting poses, formally defined as poses where the shortest line between any supported vertex on the body and the ground plane does not intersect the rest of the body geometry. However, there exist some common sitting poses that do not fulfill this criteria, where the orientations of body parts can cause errors and require additional measures. In this stage, we empirically identify two primary cases that require special attention: Poses where the person is leaning forward as well as poses where the person's legs are in a crossed position.

In the first case, the back cannot be actively supported by a chair's backrest. This is easily detected by evaluating the vertex weights on the corresponding body parts. If the back does not need any support, no backrest is created.

To detect crossed legs, we evaluate the distance between the computed planes for the outer columns of the respective rows. When the distance is under a defined minimal value, we assume that it is not possible to support both legs individually and instead fit a single plane for the combined vertices of both legs. Figure 4.8 shows an example of a pose where one foot rests on top of the other, so the initial surface mesh needs to be corrected.

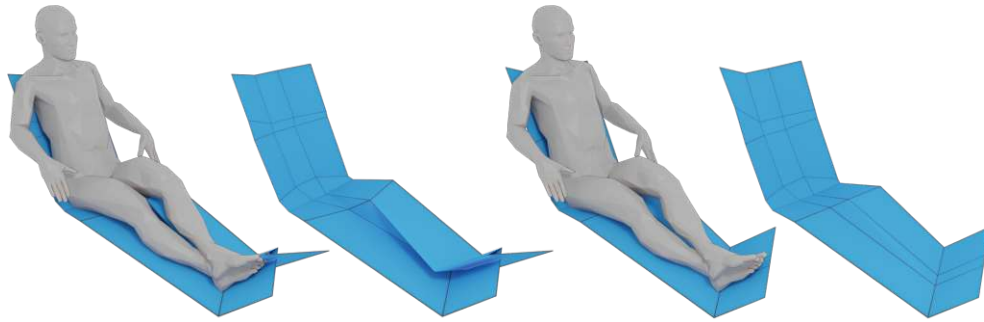


Figure 4.8: Error handling on a pose where one foot rests on top of the other. Left: surface model generated without error handling showing intersections. Right: corrected surface mesh.

### Refinement Stage

In the final stage, we add armrests to the model if they are required and connect the borders of the model to the ground. We start by constructing the armrests:

First, the algorithm starts by finding optimal planes supporting the person's upper arms and forearms. For this task, regular surface fitting is performed on the respective body parts, using PCA and an unconstrained RANSAC variant. We then find the minimal spanning rectangle on the computed plane that contains all relevant vertices that lie within supporting distance of the plane.

The next step is the integration of the armrest into the mesh grid structure. This is only possible if the armrest does not intersect the body and if it is sufficiently far away from the mesh grid. If the requirements are met, two additional columns are added to the mesh, one containing the armrest itself and another to connect the first column to the geometry.

Finally, the mesh grid is expanded in each direction by two additional rows or columns of quadrilateral faces. The outermost vertices of the resulting geometry are moved to ground height and arranged to form a rectangle. In case the surface geometry contains overhanging faces, invalid quadrilateral faces in the outermost columns of the model are possible. To correct these issues, linear optimization is performed on the outer vertices on each side of the model. This process rearranges the corresponding vertices so that each outer column face is convex.

The left side of Figure 4.9 shows visual examples for intermediate results generated from the advanced model after the refinement stage. The added border sections are lacking in visual quality in regards to planarity and regularity. Therefore, we apply an additional optimization step in which we aim to smooth the geometry and improve the planarity and regularity of the faces, while keeping the functional aspects of the surface intact.



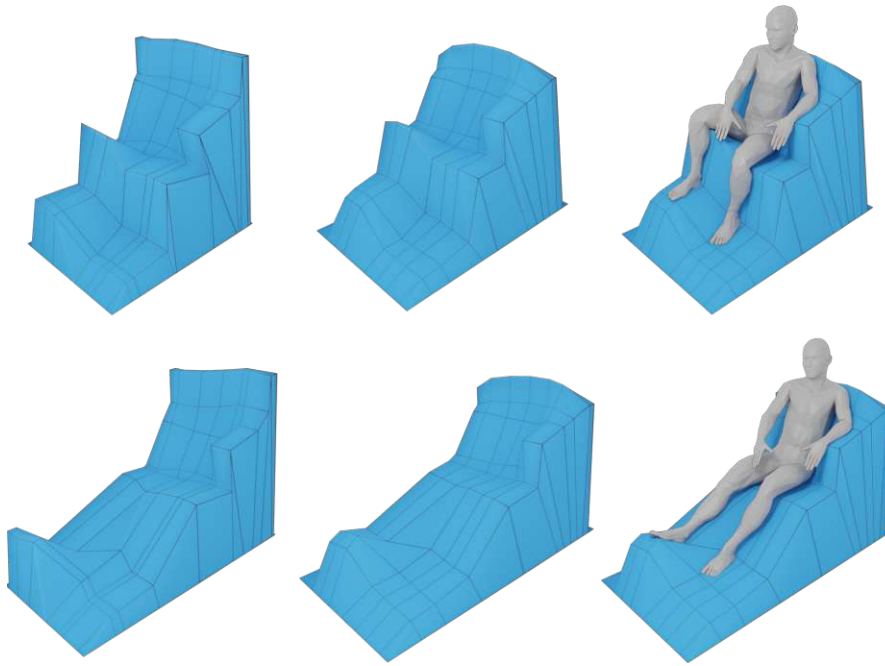


Figure 4.9: Finalized seating surface results after the optimization process. Left: seating surface before optimization. Center/Right: results after optimization .

### 4.3.2 Optimization

While the functional requirements of our furniture model are now satisfied to an adequate degree, the visual quality can still be improved. For this task we apply a non-linear local optimization process.

We formulate this as an energy minimization problem containing two terms. The first is the data term, which is used to preserve the initial configuration as much as possible, since it is the one that best satisfies the functional requirements. The second term is the visual term, which describes the visual quality regarding the smoothness of the surface as well as the regularity and planarity of its faces. The energy function is defined as

$$E = \lambda_S (S_L + S_A) + \lambda_D (D_V + D_P), \quad (4.12)$$

where  $(S_L + S_A)$  is the visual term,  $(D_V + D_P)$  is the data term and  $\lambda_S$  and  $\lambda_D$  are global weights balancing the two terms.

The *Laplacian error metric*  $S_L$  is computed as the sum of squared distances between the vertex positions and the average position of their neighboring vertices:

$$S_L = \sum_{i=1}^{n^V} \left\| \mathbf{v}_i - \frac{\sum_{j \in N_1(i)} \mathbf{v}_j w_j}{\sum_{j \in N_1(i)} w_j} \right\|^2 \lambda_S^l \quad (4.13)$$

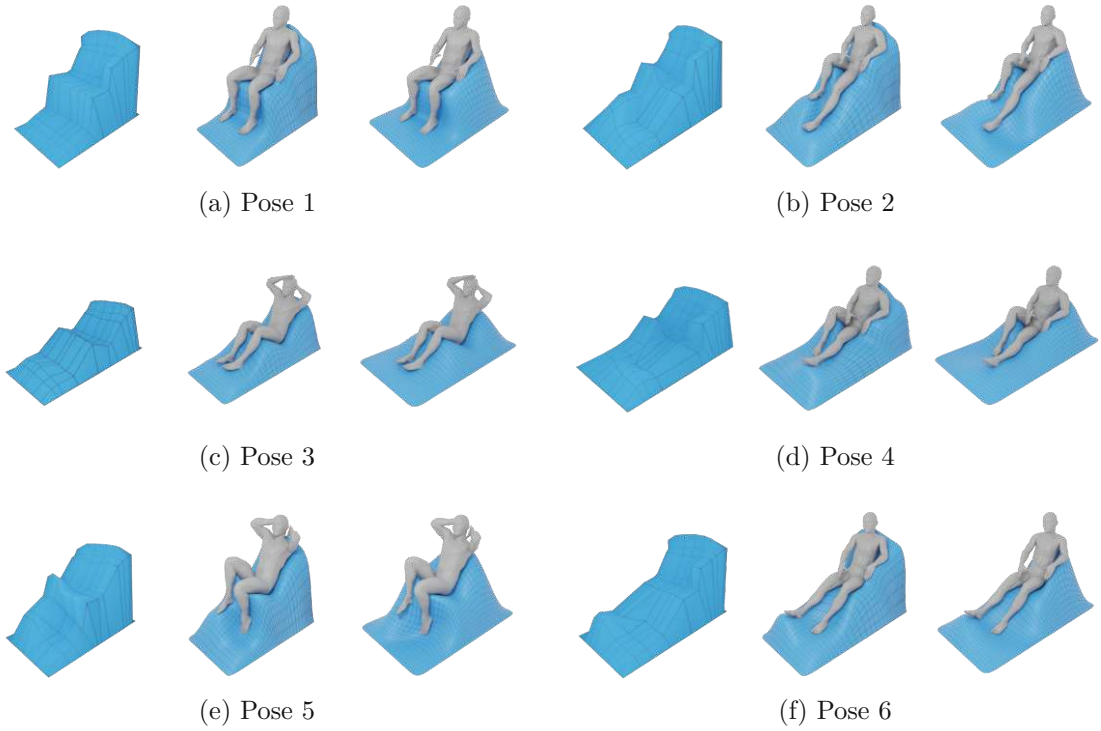


Figure 4.10: Results of our method. Left: control mesh generated with our method. Center: the fitting algorithm described in Chapter 3 applied to our control mesh. Right: the same fitting algorithm applied to a flat patch serving as the control mesh.

with  $N_1(i)$  being the 1-ring neighborhood of  $\mathbf{v}_i$ ,  $w_j$  being the importance weight for  $\mathbf{v}_j$  and  $\lambda_S^l$  being a global weight for the Laplacian error metric term.

The *angle based smoothing* term  $S_A$  is defined as

$$S_A = \sum_{j=1}^{n^F} \left( \left( \sum_{i \in F_j} \alpha_i - 2\pi \right)^2 w_j^1 \lambda_S^{A1} + \left( \sum_{i \in F_j} (\alpha_i - \pi)^2 \right) w_j^2 \lambda_S^{A2} \right) \quad (4.14)$$

with  $n^F$  being the total number of faces,  $\alpha_i$  being the  $i$ th interior angle of the face  $F_j$ ,  $w_j^1$  and  $w_j^2$  being term-specific importance weights for each face, and  $\lambda_S^{A1}$  and  $\lambda_S^{A2}$  being global weights. The first part of the term penalizes non-regular faces, while the second part aims to maximize each interior angle.

The *vertex distance term*  $D_V$  is computed from the sum of squared distances between the vertex positions of the current configuration and their corresponding original positions:

$$D_V = \lambda_D^V \sum_{i=1}^{n^v} \|\mathbf{v}_i - \tilde{\mathbf{v}}_i\|^2 w_i \quad (4.15)$$

with  $\tilde{\mathbf{v}}_i$  being the original position of vertex  $\mathbf{v}_i$ ,  $w_i$  being the importance weight of  $\mathbf{v}_i$ , and  $\lambda_D^V$  being a global weight for the term.

Finally, the *plane distance term*  $D_P$  utilizes the supporting planes that were computed in the surface fitting stage of the algorithm. Each face in the current configuration is compared to its supporting plane by computing the distance to the plane for each corner vertex:

$$D_P = \lambda_D^P \sum_{j=1}^{n^F} \left( \sum_{i \in F_j} w_i \langle \mathbf{v}_i - \mathbf{c}_j^P, \mathbf{n}_j^P \rangle^2 \right), \quad (4.16)$$

where  $\mathbf{c}_j^P$  and  $\mathbf{n}_j^P$  are the center position and surface normal of the supporting plane for face  $F_j$ ,  $w_i$  is the importance weight of vertex  $\mathbf{v}_i$ , and  $\lambda_D^P$  is a global weight for the term.

The vertex weights are chosen such that the data term is given more importance for vertices belonging to faces that support a large area of the body, while other vertices can be moved more freely to improve the visual term. We furthermore add 2 kinds of hard constraints: first, we need to constrain the position of the border vertices to stay on the edges of the rectangular base, and second, we define a minimal edge length between vertices to prevent degeneration of the geometry.

To improve the performance of solving the optimization problem, we furthermore compute the analytical gradient of the objective function. A detailed description of the gradient can be found in Appendix A.4. To solve the problem, we use MATLAB's `fmincon` function. A comparison of results from before and after the optimization can be seen in Figure 4.9.

## 4.4 Results

We apply our surface generation algorithm to a number of different poses to create a variety of body-supporting surfaces. We furthermore apply the surface fitting algorithm described in Chapter 3 using our generated surfaces as the input for the control mesh and compare the results to surfaces created using a flat patch as the control mesh, which was sometimes used for the results shown in the previous chapter. The poses are also selected from the same pose data set, which was recorded by having a design student wearing a motion capturing suit find poses that were considered comfortable. Figure 4.13 shows the control meshes created with our method on the left, with the fitting algorithm applied to it in the center, and finally the fitting algorithm applied to a flat patch on the right.

One advantage of our method is that we can infer from the construction of our control mesh which body parts should be supported or not. For example, if the creation of an armrest is impossible, it is also unlikely that we can properly support the arm by applying the fitting algorithm. We therefore do not try to fit the surface to the arm of the person. On the other hand, when simply using a flat patch as the input, the algorithm will always attempt to do so, unless additional user input specifically designates some body segments

to not be supported. This often leads to very thin regions or even self-intersections of the surface. But for our comparison, we choose to support the same body parts in both methods and also use the same algorithm parameters.

As can be seen, using a control mesh that already serves as a suitable support for the given pose improves the fit to the body when using the fitting algorithm, especially in areas of the back and arms. The reason for this is because the fitting algorithm uses a closest-point search as the basis for the assignment between surface and body. Therefore, a flat patch will have less available area for regions on the body that are further away or perpendicular to the ground plane like the back. This results in greater distortions of the surface and a worse fit. Our surface control mesh generation alleviates this problem by ensuring that each region on the body has a larger area of the surface in close proximity to enable a better fit.

To quantify the advantages of our method, we apply our pressure computation method on the poses shown in Figure 4.13 and 5.1, using only the subset of body vertices that lie within a certain distance to the corresponding generated surface. The results can be seen in Table 4.1. All of our generated surfaces result in lower values for the average distance from the body vertices to the closest surface vertex, maximum joint moments, average pressure and maximum pressure. The average joint moments are also lower in all but 2 examples, which are the result of our algorithm optimizing for both moments and pressure.

Although not all possible poses are supported by our surface generation algorithm (see Section 4.5.1), we do support a wide variety of sitting poses, including special cases like crossed legs and forward-leaning poses that do not require a backrest (left side of Figure 4.11). Also, while not treated in a special way, we can also support poses leaning back while standing (right side of Figure 4.11).

The total computation time generally ranges from 20 to 30 seconds, with control mesh generation taking between 7 and 12 seconds, visual optimization taking 12 to 14 seconds, and application of the fitting algorithm taking 1 to 5 seconds.

## 4.5 Discussion and Conclusions

### 4.5.1 Limitations and Future Work

While the developed framework fulfills our goals to a satisfying degree, we acknowledge a number of limitations and weaknesses. While the algorithm covers various difficult special cases, there is a number of common sitting poses that are currently not supported. Poses like sitting in a sideways orientation or having the feet tucked underneath the body (see Figure 4.12), would require changing the fundamental structure of the surface template to avoid intersections of the surface with the body or with itself and are thus not supported by our algorithm. Improvements to the special case detection and processing steps in the framework could increase the overall robustness of the algorithm and expand the potential input set of poses.

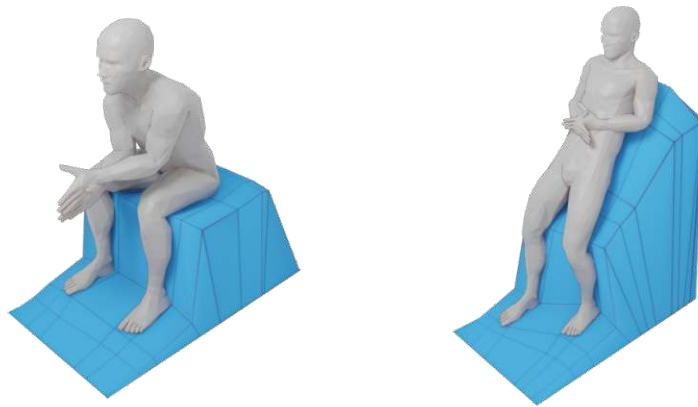


Figure 4.11: Forward-leaning poses and leaning back while standing are also supported by our method.

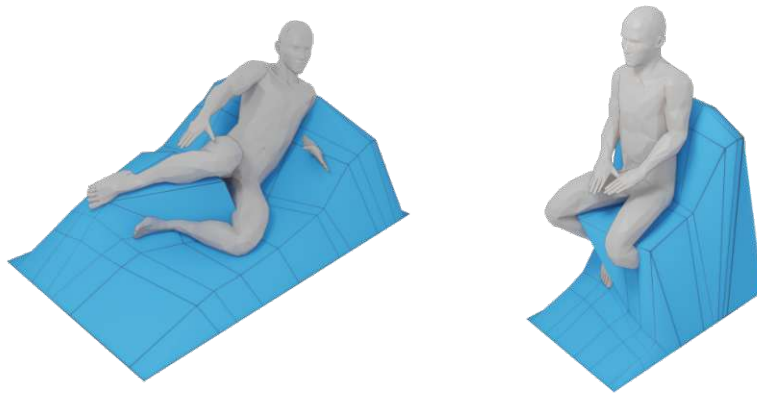


Figure 4.12: Our method fails to generate a valid surface from unsupported poses like lying on the side or placing the feet underneath the body.

Our algorithm can only be used to create a seating surface for a single person, based on a static pose. If a surface that allows seating for multiple persons is desired, the surfaces generated by our algorithm have to be manually edited. In the future, it would be interesting to extend our approach to support the generation of surfaces for multiple input poses simultaneously. This could also be used to define a range of possible input poses to allow for optimization of the seating surface for non-static poses that include possible movement while in a seated position.

Finally, the generic template only accounts for geometric consistency and functional quality. However, since our computational model of sitting delivers physical quantities which are close to reality, a fabrication-aware structural optimization of the furniture could be considered. We leave this extension for future work.

pose	avg. dist.	avg. moment	max. moment	avg. press.	max. press.
P1 - S&R	6.42 cm	4.02 Nm	11.57 Nm	449.81 N/m <sup>2</sup>	3995.31 N/m <sup>2</sup>
P1 - ours	4.00 cm	4.23 Nm	11.47 Nm	433.85 N/m <sup>2</sup>	3598.34 N/m <sup>2</sup>
P2 - S&R	7.92 cm	4.08 Nm	10.79 Nm	454.52 N/m <sup>2</sup>	4356.73 N/m <sup>2</sup>
P2 - ours	4.09 cm	4.08 Nm	10.63 Nm	438.51 N/m <sup>2</sup>	3889.42 N/m <sup>2</sup>
P3 - S&R	9.96 cm	4.62 Nm	12.05 Nm	478.29 N/m <sup>2</sup>	4186.42 N/m <sup>2</sup>
P3 - ours	6.08 cm	3.06 Nm	9.17 Nm	458.53 N/m <sup>2</sup>	3390.74 N/m <sup>2</sup>
P4 - S&R	7.36 cm	3.87 Nm	10.84 Nm	454.52 N/m <sup>2</sup>	4208.46 N/m <sup>2</sup>
P4 - ours	4.15 cm	3.84 Nm	10.74 Nm	441.37 N/m <sup>2</sup>	3953.73 N/m <sup>2</sup>
P5 - S&R	9.19 cm	4.44 Nm	11.74 Nm	478.88 N/m <sup>2</sup>	4275.08 N/m <sup>2</sup>
P5 - ours	5.15 cm	2.73 Nm	8.76 Nm	451.05 N/m <sup>2</sup>	3739.16 N/m <sup>2</sup>
P6 - S&R	7.58 cm	4.23 Nm	12.88 Nm	455.13 N/m <sup>2</sup>	3833.27 N/m <sup>2</sup>
P6 - ours	4.10 cm	4.27 Nm	12.72 Nm	432.71 N/m <sup>2</sup>	3507.86 N/m <sup>2</sup>
P7 - S&R	6.60 cm	2.22 Nm	5.66 Nm	453.61 N/m <sup>2</sup>	3351.65 N/m <sup>2</sup>
P7 - ours	3.17 cm	2.07 Nm	4.89 Nm	422.26 N/m <sup>2</sup>	2654.30 N/m <sup>2</sup>

Table 4.1: Quantitative comparison between the results when using a flat patch as input and our method for the poses shown in Figure 4.13 (P1-P6) and the pose shown in Figure 5.1 (P7). We measure the average distance from the body vertices to the closest surface vertex, average and maximum joints moments, as well as average and maximum contact pressure.

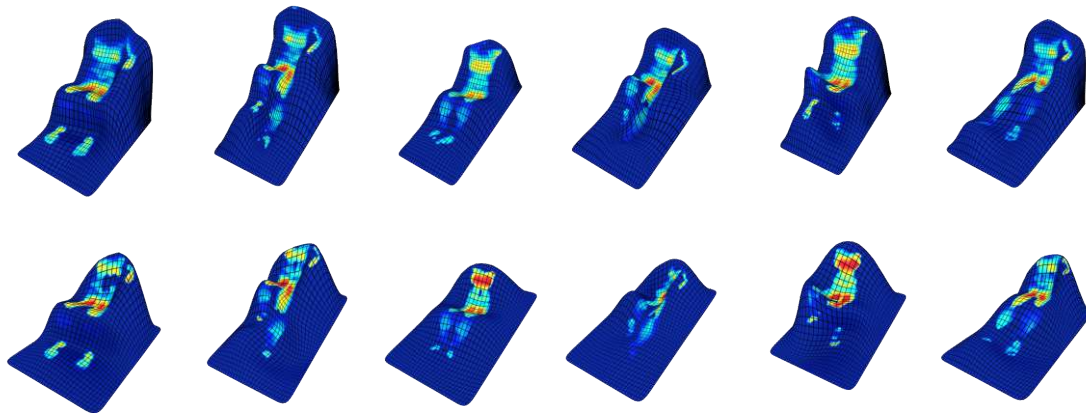
#### 4.5.2 Conclusions

We presented an automated computational framework for the generation of functional body supporting furniture that optimizes for comfortable seating in a given pose. As a measure of comfort we combine two categories considered as objective: pressure distribution and moments acting on the body. Additionally, we incorporated a friction component and proposed a computational method that handles it in interactive time in adequate accuracy as compared to sophisticated FEM methods.

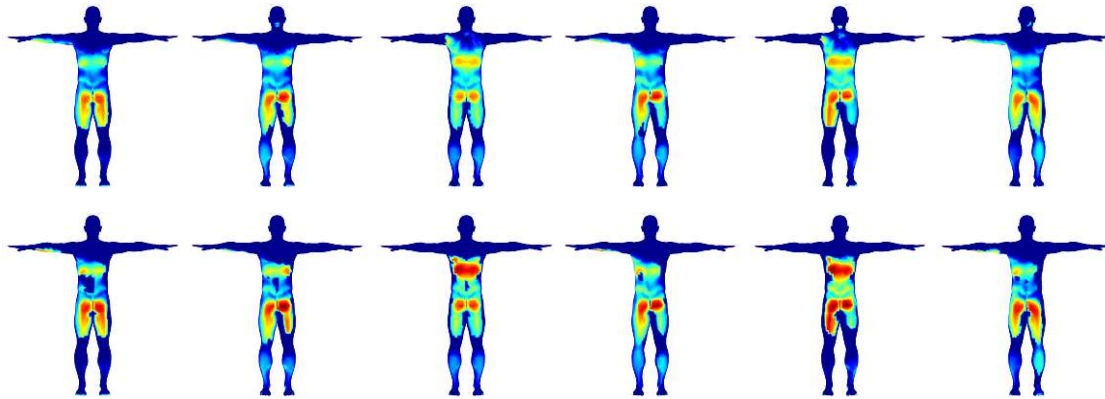
Additionally, we proposed an algorithm for an automatic generation of generic template meshes that utilizes the given comfort measure to optimally support the body. Our method is meant for computer-aided design of personalized furniture, where it can be used by professionals in order to create an initial design as well as by inexperienced users. Such designs can be then used for fabrication with modern digital manufacturing methods.

#### Acknowledgments

This research was funded by the Austrian Science Fund (FWF P27972-N31) and the Vienna Science and Technology Fund (WWTF ICT15-082).



(a) Contact pressure mapped onto surfaces created by applying the fitting algorithm of Chapter 3 to our control mesh (top) or to a flat patch serving as the control mesh (bottom).



(b) Comparison of contact pressures using our control mesh (top) or using a flat patch as the control mesh (bottom).



(c) Input contact pressure computed by assuming that the input body is perfectly supported.

Figure 4.13: Mapping of the contact pressure of Poses 1-6 onto their corresponding generated surfaces. Each column corresponds to one pose. Red color indicates high pressure values.

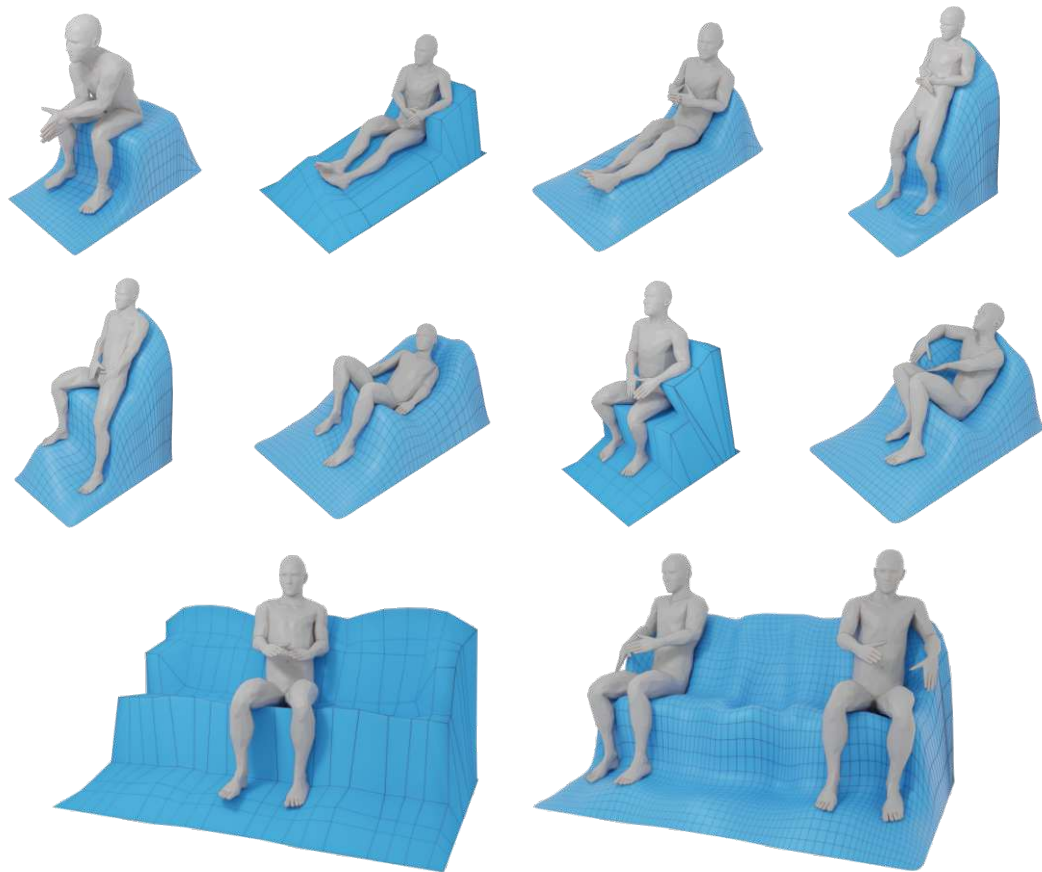


Figure 4.14: Additional results generated using our algorithm. The surface in the bottom row is created by manually editing and combining multiple control meshes.



# Data-Driven Design of Indoor Layouts

## 5.1 Introduction

Indoor spaces play a central role in our everyday lives. The synthesis and design of indoor layouts (apartment layout, workplace layout) is a long-standing problem in several disciplines, including graphics [MSL<sup>+</sup>11, FRS<sup>+</sup>12].

In this chapter, we address the problem of data-driven layout synthesis, which has recently gained renewed interest in computer graphics due to the advent of a new generation in generative machine learning [PKS<sup>+</sup>21, PGK<sup>+</sup>21]. However, despite recent progress, interior layout synthesis is still challenging for machine learning methods. The problem is twofold:

First, reliable training data is difficult to obtain. Designs need to be crafted manually by professionals, making the process labor- and time-intensive and hence expensive.

Second, readily available datasets may have been created by non-experts and may contain several issues like incorrect intersections, unrealistic placement, misplaced objects, etc. (cf. Figure 5.2). However, high-quality indoor design requires expert knowledge because good furniture arrangements are connected to several considerations like functionality, usability, aesthetics, cost-effectiveness, and ergonomics. These may not all be reflected in a dataset, which contains layouts that were most likely not created by interior design experts.

We address these problems by using a Transformer-based generative model with additional expert knowledge “injected” into the data-driven training process. Transformers are generative models originally proposed for natural language processing that have proven very successful in a wide range of domains [VSP<sup>+</sup>17]. Recently, several methods have successfully used transformers for layout generation [PGK<sup>+</sup>21, WYN20, PKS<sup>+</sup>21].

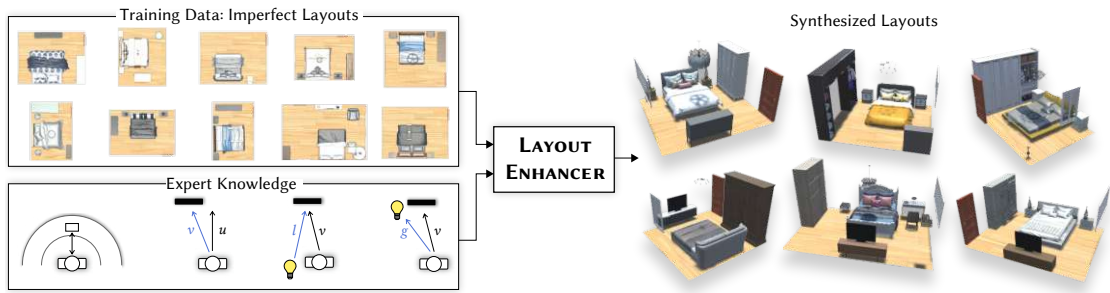


Figure 5.1: Our proposed LayoutEnhancer combines data-driven learning from potentially imperfect data with expert knowledge. Generated layouts are biased to follow rules laid out in the expert knowledge, effectively reducing the impact of data imperfections. See Figure 5.2 for examples of imperfections that are avoided due to the inclusion of expert knowledge.

In our approach, a layout  $S$  is defined as a sequence of discrete elements  $S := \{F_0, \dots, F_N\}$ , each represented with a fixed-length parameter vector. A traditional generative model learns to generate new layouts according to a probability distribution  $p(S)$  that approximates the probability distribution of the dataset  $p(S) \approx p_{\text{data}}(S)$ .

We propose to inject additional information based on expert knowledge into the learning process to obtain a learned distribution  $p'(S)$  that reflects both the dataset distribution and the additional information. The expert knowledge biases the learned probability distributions to emphasize or de-emphasize specific properties of the layouts. In Section 5.2 we derive a set ergonomic rules from expert literature [Kro17].

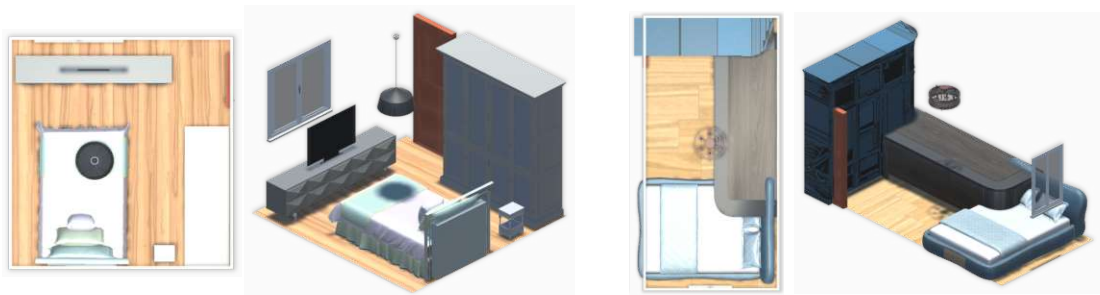


Figure 5.2: LayoutEnhancer can learn to improve issues found in imperfect data like *ergonomic issues* (left room): (i) a window directly behind the TV causes glare on sunny days, making it difficult to watch due to a big contrast in brightness. (ii) Insufficient illumination for reading a book without a light source behind or beside the bed; and *geometric issues* (right room): (i) desk is intersecting with the bed and the closet; (ii) closet is covering the door.

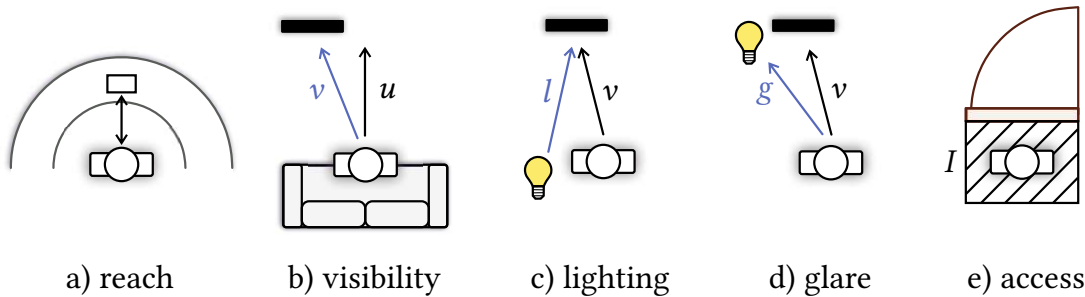


Figure 5.3: Ergonomic rules implemented in our system. We chose these guidelines as they are essential in most indoor scenarios, like reading a book, watching TV, or working at the desk or the computer. We convert the rules to scalar cost functions and evaluate them using activities (cf. Section 5.2).

We integrate this information into the loss function of our transformer-based generative model in two ways: (i) as weights of training samples and (ii) as additional loss that assesses the quality of samples proposed during the training process. In the second case, expert knowledge needs to be differentiable w.r.t. the predicted probabilities. We discuss the details in Section 5.3.

In Section 5.4, we evaluate the proposed method and compare it to a recent data-driven method that does not utilize expert knowledge [PKS<sup>+</sup>21]. We demonstrate that with our approach we can improve the ergonomic quality of generated layouts, effectively increasing the perceived realism compared to others.

In summary, the contributions presented in this chapter are three-fold:

- We introduce a differentiable ergonomic loss that can be used to assess the ergonomic quality of interior layouts. We derive this loss from the expert knowledge in ergonomics (Section 5.2).
- We integrate this differentiable loss into the training of a Transformer network (Section 5.3).
- We empirically show that we can train a generative model with this loss that creates samples with increased ergonomic quality and realism compared to the state of the art (Section 5.4).

## 5.2 Ergonomic Costs

To derive a set of rules used to quantify an ergonomic quality of a design, we studied the literature of ergonomic guidelines [Kro17]. As a result, we order the information in a hierarchical manner, using the building blocks of activities, actions and ergonomic costs.

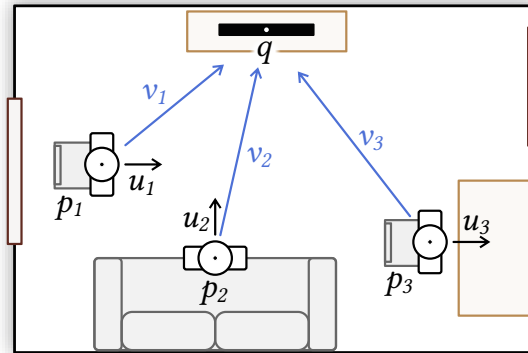


Figure 5.4: Human activity in the room based on the example of *Watch TV*. For all possible sitting locations  $p_j$  an avatar is sampled and the ergonomic rules for visibility and glare are evaluated. The final contribution is the weighted sum of costs over every combination of a sitting possibility  $p_j$  and all TVs  $q_k$ . Please refer to Section 5.2 for more details.

An activity is a set or sequence of actions that need to be performed to accomplish a specific goal [PRB<sup>+</sup>18]. An activity could be, for instance, reading a book or watching TV. A single action puts specific elements of a layout into a common context, for example looking at the TV while sitting on the sofa. Ergonomic costs are evaluated for each action to quantify how suitable the arrangement of the layout elements is in an ergonomic sense.

The ergonomic losses obtained for each evaluated ergonomic rule are then aggregated up the hierarchy to obtain the losses for each action, activity and finally for the whole layout. This formulation makes it easy to define new evaluation functions for different activities by combining the various building blocks. In our approach, we consider the following ergonomic costs (cf. Figure 5.3):

- Reach measures how easy it is to interact with a target object from a given position.
- Visibility measures how visible a target object is for a given position and viewing direction.
- Lighting measures how well an object is illuminated by light sources in the room.
- Glare measures the decrease in visual performance from strong brightness contrast caused by having bright light sources in the field of view.
- Accessibility measures how much free space is in front of a target object to allow easy interaction and walking by.

Naturally, this list of rules is far from exhaustive and one may choose to include additional rules depending on the type of space that needs to be designed. However, we choose the

above five rules as examples for two reasons. First, they are all relevant for the kinds of activities that are often performed in the prevalent room types that are included in publicly available indoor layout datasets. The second reason is a practical one, since these rules can be defined as (piecewise) differentiable scalar functions in a range of  $[0, 1]$ , which perfectly suits our needs. In this section, we first describe the individual ergonomic cost functions for each rule, followed by the activities we use to evaluate the layouts.

### 5.2.1 Ergonomic rules

#### Reach

While being seated, a person only has limited mobility and thus objects that need to be interacted with should be within a distance that is easy to reach without the need to stand up. We can broadly categorize the area around a seated person into 3 zones. In the inner zone, objects can be reached without much effort, while objects in the outer zone are beyond reach. Objects in the middle zone can still be reached, but require more effort the further away they are. We model this reach cost  $E_R$  as a sigmoid function that measures how difficult it is to reach an object at position  $q$  from position  $p$ :

$$E_R = \frac{1.0}{1.0 + \exp(-\beta_R (\|q - p\| - d_R))}. \quad (5.1)$$

The function is centered at  $d_R$  with scaling parameter  $\beta_R$ . We use  $d_R = 0.8$  and  $\beta_R = 15$  to model the zones of easy and extended reach. These parameters roughly correspond to an easy reach up to  $0.5m$  up to which the cost is close to 0 and an extended reach up to  $1.0m$ , towards which the cost increases to 1.0.

#### Visibility

Visibility cost measures how visible a target object is from the viewpoint of the avatar given by position  $p$  and viewing direction  $u$ . This measure is important for activities like watching TV or using the computer (cf. Table 5.1), since seating furniture with sub-optimal positions or orientations may require the user to take on unhealthy postures. To introduce this cost as smooth scalar function  $E_v$  which can be minimized, we define the cost to increase with the angle between the two vectors  $u$  and  $v = \frac{q-p}{\|q-p\|}$ :

$$E_V = 1 - \left( \frac{1 + \langle u, v \rangle}{2} \right). \quad (5.2)$$

#### Lighting

Lighting cost measures how well an object is illuminated by light sources in the room. Ideally, when looking at an object, the viewer and the light source should be positioned in the same half-space of the viewed object, as otherwise the object itself would partially obstruct the direct illumination and cause self-shadowing. A light source  $b_i$  is thus well suited for illuminating the object at position  $q$  when viewed from position  $p$  as long as

the position-to-object vector  $v = \frac{q-p}{\|q-p\|}$  and the vector  $l_i = \frac{q-b_i}{\|q-b_i\|}$  pointing from a light source at position  $b_i$  to  $q$  do not point in opposite directions:

$$e_i^L = \left(1 - \frac{1 + \langle v, l_i \rangle}{2}\right).$$

Since multiple light sources can contribute to this cost, we compute their contribution by applying the softmax function to the vector  $e^L = [e_i^L]_{i \in B}$  and using them as weights for computing the weighted sum:

$$E_L = \langle e^L, \text{softmax}(\beta \cdot e^L) \rangle, \quad (5.3)$$

with  $\beta$  being a temperature parameter that determines the hardness of the softmax function. We use  $\beta = 10$ . Since the computation of indirect illumination is prohibitively expensive, we only consider direct lighting.

### Glare

Glare cost  $E_g$  measures the decrease in visual performance from strong brightness contrast caused by having bright light sources in the field of view. Given position-to-object vector  $v = \frac{q-p}{\|q-p\|}$  and glare vector  $g_i = \frac{b_i-p}{\|b_i-p\|}$  pointing from  $p$  to the light source at  $b_i$ , the cost increases as the angle between the vectors decreases:

$$e_i^G = \left(\frac{1 + \langle v, g_i \rangle}{2}\right).$$

Similar to the lighting cost we compute the weighted sum of multiple light sources using the softmax function for computing the weights:

$$E_G = \langle e^G, \text{softmax}(\beta \cdot e^G) \rangle. \quad (5.4)$$

For simplicity, we do not consider indirect glare, such as light sources that are reflected by a computer screen. Ceiling lights such as chandeliers are also excluded from this rule since light sources positioned above the field of view have a smaller impact on visual performance [Kro17].

### Accessibility

The accessibility cost  $E_A$  measures how much space is available in front of a target object to allow easy interaction and walking through the room. For example, it is necessary to provide sufficient space between a bed and a wardrobe so that the wardrobe can be easily opened. We quantify this cost by defining an interaction region  $I_j$  for each object  $F_j$  that should not intersect with the bounding box  $A_k$  of any another object  $F_k$  in the layout. For most object categories, this region is located in front of the object itself, with a width equal to that of the object and an empirically chosen depth of  $0.5m$ . An exception is made for beds, since they are usually interacted with from the sides, so we define 2 such regions on either side with a width equal to half the depth of the bed and

a depth of  $0.5m$ . Given a furniture object  $F_j$  with interaction region  $I_j$ , we define the accessibility cost  $E_a$  as

$$E_A = \sum_{k=0}^N \frac{|I_j \cap A_k|}{|I_j|}. \quad (5.5)$$

### 5.2.2 Activity Evaluation

We evaluate the ergonomic loss of a layout in the context of activities that are typically performed in rooms of a given category. Based on research on this topic [PRB<sup>+</sup>18], we select 4 such activities which we label as *Read book*, *Watch TV*, *Use computer* and *Work at desk*. To evaluate an activity, it is necessary to compute the ergonomic costs relevant to that activity (cf. Table 5.1). We furthermore use a logarithmic function to re-scale the ergonomic cost functions to more strongly punish scenes with high costs, for example

$$\bar{E}_R = -\ln(1.0 + \epsilon - E_R), \quad (5.6)$$

with the scaling functions for the other rules defined analogously. We use  $\epsilon = \exp(5)$ , so that when  $E_R = 1$ , then  $\bar{E}_R = 5$ . We found this scaling function to be beneficial for minimizing the ergonomic loss during network training.

Since the accessibility cost  $E_A$  is relevant for every activity, we decide to compute this term once for the entire layout instead of computing it separately for every activity for performance reasons. We thus define the accessibility cost for the entire layout as

$$E_{access} = \langle e^{access}, \text{softmax}(\beta \cdot e^{access}) \rangle, \quad (5.7)$$

with  $e^{access} = [\bar{E}_A(I_j)]_{j=1, \dots, N}$  being the vector containing the accessibility cost of every object and using  $\beta = 10$ . The softmax function is used to normalize the total cost of the layout such that a single badly-placed object increases the cost by roughly the same amount regardless of the number of objects in the layout.

For the activity *Read book*, proper illumination conditions are the most important factor, so we need to apply the rules for lighting and glare. Given the position  $p_j$  of seating furniture (like beds, chairs, or sofas), an associated object position  $q_j$  (a book close to  $p_j$ ) and light sources  $B$  we define

$$e_j^{book} = \frac{\bar{E}_L(p_j, B, q_j) + \bar{E}_G(p_j, B, q_j)}{2}.$$

Table 5.1: Associations of rules to activities that can be performed in an environment. Not all activities require all rules to be fulfilled.

	Reach	Visibility	Lighting	Glare	Accessibility
Read book			yes	yes	yes
Watch TV		yes		yes	yes
Use computer	yes	yes		yes	yes
Work at desk	yes	yes	yes		yes

Since we do not require all possible positions to have a good score for every activity, we once again use the softmin function to compute a weighted sum of costs for the layout. That way, if there is only one position that is suitable for an activity, it will be the only one with a large contribution to the layout cost, while having multiple suitable positions will have them contribute equally. For a set of positions  $p_j \in P$  we therefore have

$$E_{book} = \langle e^{book}, \text{softmin}(\beta \cdot e^{book}) \rangle, \quad (5.8)$$

with  $e^{book} = [e_j^{book}]_{j \in P}$  and using  $\beta = 10$ .

The other activities are defined similarly. For *Watch TV*, we require the TV to be visible from a piece of seating furniture and there should not be a light source in the field of view. We therefore compute the visibility and glare costs for positions  $p_j$  with orientation  $u_j$  (for chairs, beds, sofas) and TVs with position  $q_k$ :

$$e_{j,k}^{tv} = \frac{\bar{E}_V(p_j, u_j, q_k) + \bar{E}_G(p_j, B, q_k)}{2}.$$

Since there can be multiple TVs in a room in addition to multiple pieces of seating furniture, we need to compute the weighted sum of costs over every combination of  $p_j$  and  $q_k$ , using  $e^{tv} = [e_{j,k}^{tv}]_{j \in P, k \in Q}$ :

$$E_{tv} = \langle e^{tv}, \text{softmin}(\beta \cdot e^{tv}) \rangle. \quad (5.9)$$

The same rules are required for the activity *Use computer*, in addition to the reach rule since the seating furniture and computer should be in close proximity. We do not evaluate the lighting rule because the direction from which the light illuminates the computer is not as important, since the computer screen is already illuminated. Using  $q_k$  to denote the positions of computers we define

$$e_{j,k}^{comp} = \frac{\bar{E}_V(p_j, u_j, q_k) + \bar{E}_G(p_j, B, q_k) + \bar{E}_R(p_j, q_k)}{3}.$$

Finally, for the activity *Work at desk* we apply the rules visibility, lighting and reach. Since the viewing angle is mostly directed downward toward the desk during this activity, it is not necessary to consider direct glare caused by light sources in the room. Given chair positions  $p_j$ , table positions  $q_k$  and light sources  $B$  we compute

$$e_{j,k}^{work} = \frac{\bar{E}_V(p_j, u_j, q_k) + \bar{E}_L(p_j, B, q_k) + \bar{E}_R(p_j, q_k)}{3}.$$

To obtain the overall **ergonomic loss**  $E$  for a layout we take the average of all activity costs that are possible in the layout (e.g. if there is no computer in the scene, we do not evaluate the cost for *Use computer*):

$$E = \frac{\sum_a \delta_a E_a}{\sum_a \delta_a},$$

with  $a \in \{access, book, tv, comp, work\}$  and  $\delta_a = 1$  if the corresponding activity can be performed in the layout and  $\delta_a = 0$  otherwise.



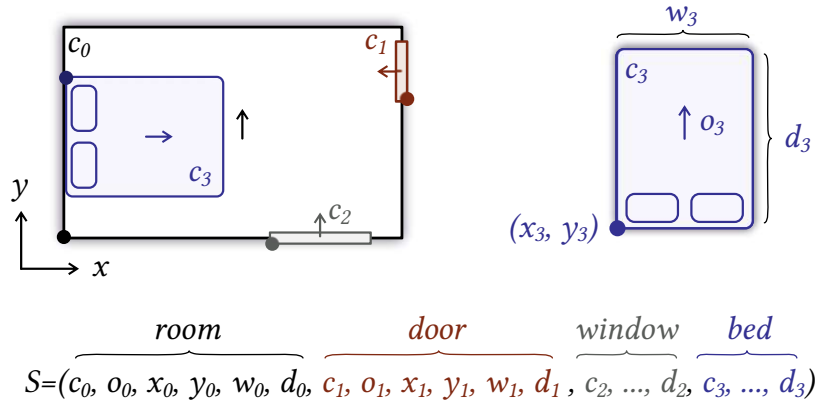


Figure 5.5: A layout is represented as a sequence  $S = (s_1, \dots, s_n)$ . Each individual token  $s_i$  in the sequence represents an attribute of a furniture object, such as its category, orientation, position or dimensions.

## 5.3 Layout Generation with Expert Knowledge

We build on top of Transformers [VSP<sup>+</sup>17] as generative model for layouts [PGK<sup>+</sup>21, WYN20, PKS<sup>+</sup>21]. In this section, we first present our model and then describe how we integrate our ergonomic loss into the training.

### 5.3.1 Generative Model

Transformers are sequence generators that originate from natural language processing. A layout is generated step-wise as a sequence of discrete tokens  $S = (s_1, \dots, s_n)$ , one token  $s_i$  at a time. Thus, we first need to define a sequence representation of our layouts.

**Sequence representation** Each furniture object is represented as a 6-tuple  $F_i = (c_i, o_i, x_i, y_i, w_i, d_i)$ , with  $c_i$  indicating the object category, such as *chair* or *table*,  $o_i$  the orientation,  $x_i$  and  $y_i$  being the x- and y-coordinates of the bottom left corner of the furniture object,  $w_i$  being the width, and  $d_i$  the depth of the furniture object (cf. Figure 5.5). Since previous work [PKS<sup>+</sup>21] has shown that randomizing the order of objects that do not admit a consistent ordering can be beneficial, we follow a similar approach. The bounding box of the room itself is represented as the furniture object  $F_0$  and is thus always the first of the ordered furniture objects, followed by the doors and windows of the layout. The order of all other furniture objects is not consistent and instead randomized during training. We concatenate the 6-tuples of the ordered furniture objects and add a special stop token to the end of the sequence to obtain the sequence  $S$ . An example can be seen in Figure 5.5.

Similar to previous work [WYN20], we use two additional parallel sequences to provide context for each token in  $S$ : a position sequence  $S^P = (1, 2, \dots, n)$  that provides the

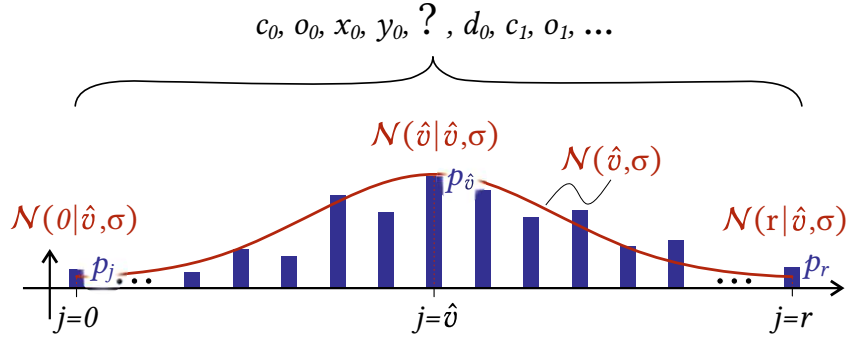


Figure 5.6: To propagate the ergonomic loss back to the token probabilities, we choose the maximum of the discrete values of the predicted token and convolve the neighborhood with a Gaussian kernel, centered at the discrete maximum. The resulting token value is a weighted sum of the discrete values in this neighborhood, weighted by the probability and distance to the kernel center of each discrete value. Please refer to Section 5.3.2 for more details.

global position in the sequence, and an index sequence  $S^I = (1, 2, \dots, 6, 1, 2, \dots, 6)$  that describes the index of a token inside the 6-tuple of a furniture object.

Our approach also supports an alternate method of providing the room shape as a binary map of the floor plan, similar to ATISS [PKS<sup>+</sup>21]. While specifying the room as part of the sequence allows the network to learn how to synthesize arbitrary rectangular rooms, using a binary map instead lets the network learn how to generate furniture layouts for more complex non-rectangular room shapes.

**Quantization** Transformers typically operate with discrete token values. By learning to predict a probability for each possible value of a token, a transformer can model arbitrary distributions over token values. To obtain discrete values, we quantize all object parameters except orientations  $o_i$  and categories  $c_i$  uniformly between the minimum and maximum values that occur in the dataset. Orientations  $o_i$  are uniformly quantized in  $[0, 2\pi)$ , adjusting the resolution to preserve axis-aligned orientations as integer values. We use a resolution of  $r = 256$ . Categories  $c_i$  do not require quantization as they are already integers. We use categorical distributions for all tokens.

**Sequence generation** Our Transformer-based sequence generator  $f_\theta$  factors the probability distribution over sequences  $S$  into a product of conditional probabilities over individual tokens:

$$p(S|\theta) = \prod_i p(s_i | s_{<i}, \theta),$$

where  $s_{<i} := s_1, \dots, s_{i-1}$  is the partial sequence up to (excluding)  $i$ . Given a partial sequence  $s_{<i}$ , our model predicts the probability distribution over all possible discrete values for the next token:  $p(s_i | s_{<i}, \theta) = f_\theta(s_{<i}, s_{<i}^P, s_{<i}^I)$  that can be sampled to obtain the

next token  $s_i$ . Here  $s_{<i}^P$  and  $s_{<i}^I$  are the corresponding partial position and index sequences that are fully defined by the index  $i$ . We implement  $f_\theta$  as a GPT-2 model [RWC<sup>+</sup>19] using the implementation included in the Huggingface library [WDS<sup>+</sup>20].

### 5.3.2 Ergonomic Loss

A loss designed by an expert, such as an ergonomic rule, defines desirable properties of layouts that may not be fully realized in a dataset. However, while minimizing the expert loss may be *necessary* to obtain a desirable layout, it is usually not *sufficient*, since a manually defined loss can usually not describe *all* desirable properties of a layout exhaustively. Our goal is thus to combine the expert loss with a data-driven generative model for layouts. However, integrating the ergonomic loss in a transformer-based generative model poses two main challenges:

**C1:** Transformers generate layouts in multiple steps, each step generating a small part of the layout such as a single object or a single object attribute. Each step, where only a partial layout has been generated, requires supervision, but the ergonomic loss cannot reliably be computed on a partial layout.

**C2:** The ergonomic loss is defined over continuous parameters, such as object positions or orientations. However, transformers typically output a probability distribution over a discrete set of values in each step, such as quantized object positions or orientations. This makes gradient propagation from the ergonomic loss to the transformer difficult.

To tackle the first challenge (**C1**), we observe that transformers are typically trained with a strategy called *teacher forcing*, where the partial sequence  $s_{<i}$  preceding the current token  $s_i$  is taken from a ground truth layout. Thus, when generating a token  $s_i$ , we can evaluate the ergonomic loss on the layout defined by  $s_{<i}, s_i, s_{>i}$ , where only  $s_i$  is generated and both the preceding tokens  $s_{<i}$  and the following tokens  $s_{>i}$  are taken from the ground truth, effectively evaluating  $s_i$  in the context of the ground truth layout.

To solve the second challenge (**C2**) we need an ergonomic loss that is differentiable w.r.t. the probabilities  $p(s_i|s_{<i}, \theta)$  predicted by our generative model. A straight-forward solution computes the expected value of the ergonomic loss  $E$  over all possible values  $v_j$  of a token  $\sum_j E(s_{<i}, v_j, s_{>i}) P(s_i = v_j|s_{<i}, \theta)$ . This solution is differentiable w.r.t. the probabilities, but requires an evaluation of the ergonomic loss for each possible value of a token, which is prohibitively expensive. Instead, we opt for a less exact but much more efficient approach, where only a single evaluation of the ergonomic loss per token is needed. We compute the ergonomic loss  $\mathcal{L}_E$  as the ergonomic loss for the expected value of a token in a small window around the most likely value of the token:

$$\begin{aligned} \mathcal{L}_E &= E(s_{<i}, \bar{v}, s_{>i}), \text{ with} & (5.10) \\ \bar{v} &= \frac{\sum_j (\mathcal{N}(v_j|\hat{v}, \sigma) P(s_i = v_j|s_{<i}, \theta) v_j)}{\sum_j (\mathcal{N}(v_j|\hat{v}, \sigma) P(s_i = v_j|s_{<i}, \theta))}, \end{aligned}$$

where  $\mathcal{N}(x|\hat{v}, \sigma)$  is the normal distribution centered at  $\hat{v}$  with standard deviation  $\sigma$ .  $\hat{v}$  is the token value with highest probability, and  $\sigma$  is set to  $1/r$  in our experiments. Figure

5.6 illustrates the approach. This loss provides gradients to all values in smooth window. Note that increasing the size of the window by increasing  $\sigma$  would propagate the gradient to a larger range of token values, but could also result in expected token values  $\bar{v}$  that are in low-probability regions of the distribution  $p(s_i|s_{<i}, \theta)$ , since the distribution may be multi-modal. The total loss function  $\mathcal{L}$  is then given by

$$\mathcal{L}(S^k) = \beta_T \mathcal{L}_T(S^k) + \beta_E \mathcal{L}_E(S^k), \quad (5.11)$$

with  $\mathcal{L}_T$  being the cross-entropy loss,  $\mathcal{L}_E$  being our proposed ergonomic loss and  $\beta_T, \beta_E$  being weights that determine the influence of the two loss terms to the overall loss. We use  $\beta_T = 1 - E(S^k)$  and  $\beta_E = E(S^k)$ , such that the cross-entropy loss term has higher influence for training samples with better ergonomic quality, while the ergonomic loss term is more important for samples with lower ergonomic quality. Essentially, we want the network to learn about the general target distribution from examples that are already considered good, while learning how to improve the ergonomic loss from bad examples. In Section 5.4.1, we discuss the influence of the weights  $\beta_T$  and  $\beta_E$  in more detail.

### 5.3.3 Implementation Details

#### Dataset

We train our models using the 3DFRONT dataset [FJG<sup>+</sup>21, FCG<sup>+</sup>21] as training data. In a pre-processing step, we parse the data to extract rooms belonging to the categories Bedroom, Dining Room, Living Room and Library. For this purpose we use the filter criteria provided by ATISS [PKS<sup>+</sup>21], consisting of a list of rooms for each category, as well as a split into training, testing and validation data. We use the rooms marked as *train* for our training sets and combine those marked as *test* and *val* for our validation sets. Depending on the use case, we apply also some additional filtering. If we want to provide the attributes of the room shape as part of the input sequence, we can only use rectangular rooms and thus filter out rooms with more complex shapes. For the Bedrooms dataset, this results in 4041 rooms for the training set and 324 rooms for the validation set. For the direct comparison with ATISS, we provide the room shape using a binary map of the floor plan, allowing us to also use non-rectangular rooms, but we need to exclude rooms that have also been filtered by the pre-processing algorithm of ATISS. This leaves in 3526 rooms for the training set and 289 rooms for the validation set.

For most furniture objects, their attributes such as the category and the transformation of the corresponding 3d model data can be directly extracted from the room data. Since separate 3d models for doors and windows are not provided with the dataset, we extract their positions and bounding box dimensions from the mesh data with corresponding labels. Since doors are only provided with each house and not attached to individual rooms, we include a door with the furniture objects of a room if its distance to the closest wall of the room is lower than a chosen threshold and its orientation is aligned with that of the wall.

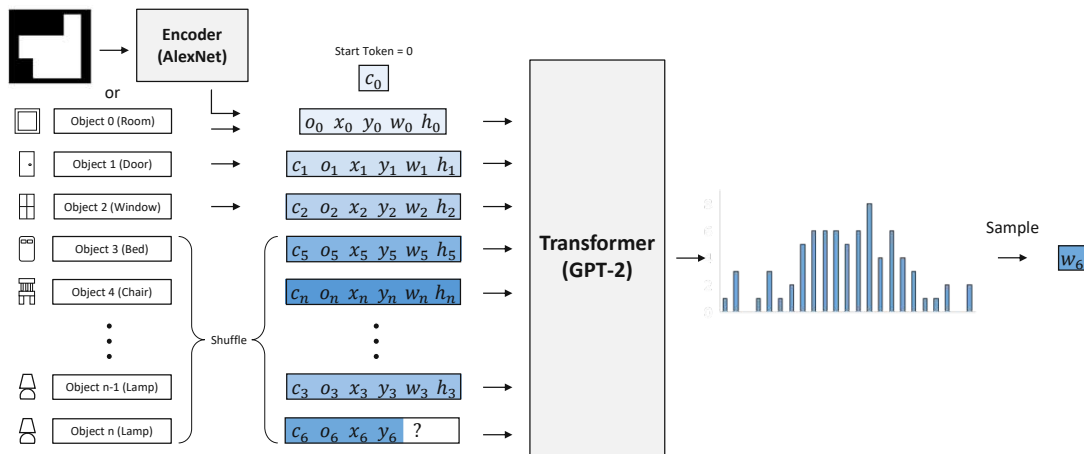


Figure 5.7: Overview of our model. A room layout consisting of individual furniture objects is mapped to a sequence of tokens which serves as the input to the transformer model. Given this sequence, the network predicts a categorical distribution for the next token from which we randomly sample the actual token value. During training, the order of objects other than the room, doors and windows is shuffled in the sequence. Furthermore, the attributes of the room can be either mapped to tokens directly (for rectangular rooms only), or by using an additional encoder network given a binary image of the floor plan as input.

Additionally, we group some of the object categories in the dataset that are very similar to each other, while filtering out some others that occur only in very few rooms, for a total of 31 categories that we use across all room types.

Since the dataset is typically lacking object categories that are necessary to properly evaluate the ergonomic loss of a layout, we augment the dataset with additional objects in the following way. For each layout, there is a 50% chance to place a furniture object of the indoor lamp category in the center of every stand and side-table object. In the same manner, a computer object is placed at the center of each desk object in a layout with a probability of 50%. Finally, every TV stand object is augmented with a TV object.

### Training.

As hyperparameters for our networks we use 12 hidden layers, 8 attention heads, embedding dimensionality of 256, dropout probability of 0.1 and a batch size of 64. Each network is trained for 150 epochs, with the number of steps per epoch being equal to the number of training samples, such that each sample is used once per epoch. During training, each training sample is randomly augmented by horizontal mirroring and/or rotation in 90° steps, in addition to applying a random permutation on the order of furniture objects other than the room, windows and doors. We use a learning rate of 0.0001 with a linear rate of decay and a warm-up period of 10 epochs. These parameters

were determined empirically in preliminary experiments. For layout synthesis, we always choose the learned network parameters of the epoch with the smallest validation loss during training.

When the shape of the room is provided as a binary map of the floor plan, we use an additional convolutional neural network to convert the binary map into a feature embedding. For this network we directly use the implementation of the AlexNet architecture [KSH12] provided in the code framework of ATISS [PKS<sup>+</sup>21]. We also experimented with a ResNet-18 architecture [HZRS15], but found that AlexNet is more successful at discriminating between mirrored floor plans. The computed feature embedding then replaces the embedding that is otherwise computed from the orientation, position and dimension tokens of the room furniture object(cf. Figure 5.7).

Our networks are trained on Google Colab, using a machine with a NVIDIA Tesla P100 GPU. When only using the cross-entropy loss, training for one epoch takes 13 seconds on average. Adding our ergonomic loss increases training times to 123 seconds per epoch on average, since we cannot make use of parallelization for layout evaluation as easily. There is room for further optimizations in this aspect.

### **Inference.**

During inference, we follow a similar approach to the strategy proposed by Sceneformer [WYN20], using top- $p$  nucleus sampling with  $p = 0.9$  for the object categories, as well as the attributes of the room, doors and windows. For the attributes of other object categories, we always pick the token with the highest probability.

The layouts synthesized by the transformer network sometimes include intersecting objects which greatly disturb the perceived realism of a layout. We therefore follow the approach of similar methods like Sceneformer and check for object intersections during inference. After the attributes of a furniture object have been generated, we check if the object can be inserted into the scene without causing large intersections. If this is not the case, we re-sample the category and other attributes of the current object. If this re-sampling approach fails too often (we choose a limit of 20 attempts experimentally), we discard the entire layout and start anew. Certain pairs of object categories are excluded from this check, e.g. chairs can be put underneath a table and thus do not cause collisions.

In terms of computation time, the intersection-detection process is the bottleneck of the inference process. If we do check for intersection during inference, it takes 1653 seconds for our models to synthesize 1000 layoutsequences, for 1.653 seconds per layout on average. If we do not perform intersection-checks between objects, we can make use of parallelization to greatly reduce inference time. In such a setup, our networks can synthesize 1000 layout sequences in 27 seconds for 0.027 seconds per scene on average.

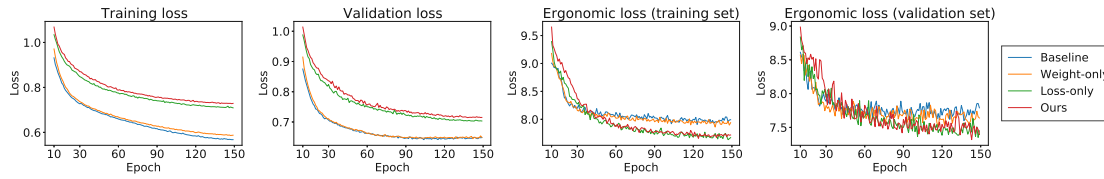


Figure 5.8: Cross-entropy loss and ergonomic loss for our model and its ablations, evaluated on the Bedrooms dataset. As training loss and validation loss we refer to the cross-entropy loss evaluated on the training and validation sets respectively. By including our proposed ergonomic loss term during training we can significantly decrease the ergonomic loss of synthesized layouts. The first 10 epochs are omitted for readability.

### Scene reconstruction.

Since our networks only generate the 2d bounding boxes of furniture objects, we use an additional post-processing step to reconstruct a 3d scene from the generated layout. For each furniture object, we select the 3d model of the same category with the smallest difference in bounding box dimensions from the models in the 3DFRONT dataset [FJG<sup>+</sup>21, FCG<sup>+</sup>21]. For categories not included in the dataset, such as doors and windows, we handpick a few suitable models from online sources [Tur22].

As a final step, the vertical position of each object is adjusted based on its category. The position of some categories like windows and chandeliers are set to a fixed height. We label some categories as supporting objects (like tables and stands) and others as supported objects (like indoor lamps and TVs). If there is an intersection between a supporting and supported object, the vertical position of the supported object is adjusted to be placed on top of the supporting object.

## 5.4 Results and Evaluation

### 5.4.1 Ablation

To evaluate the influence of our proposed ergonomic loss, we define 3 ablations of our network that are trained with different loss functions. Recall that the total loss function of our approach given in Eq. 5.11 is defined as the weighted sum of the cross-entropy loss  $\mathcal{L}_T$  and the ergonomic loss  $\mathcal{L}_E$  with weights  $\beta_T, \beta_E$ . Using these weight parameters, we define the following 3 ablations of our network:

- Baseline, with  $\beta_T = 1$  and  $\beta_E = 0$ ,
- Weight-only, with  $\beta_T = 1 - E(S^k)$  and  $\beta_E = 0$ ,
- Loss-only, with  $\beta_T = 1$  and  $\beta_E = 1$ .

In other words, the baseline model only uses the cross-entropy loss with each input sample having equal weight and is thus without any of our enhancements. The weight-only model uses the cross-entropy loss with each sample being weighted by its ergonomic loss, while the loss-only model uses the sum of cross-entropy loss and ergonomic loss with each input sample having equal weight.

Figure 5.8 depicts the cross-entropy loss and ergonomic loss evaluated on both the training and validation sets for each version, using the Bedroom dataset for training. The results show a decrease in ergonomic loss for both the loss-only model and our full model which make use of our ergonomic loss term during training. While the decrease may seem small relative to the overall loss, please keep in mind that the loss is computed for the entire scene with only one token predicted by the network. The weight-only model only yields a small decrease of ergonomic loss during training, since weighting the training samples by their ergonomic loss only reduces the influence of bad training samples without teaching the network how to improve the sample. However, this still has a noticeable effect on the synthesized scenes as we will discuss in Section 5.4.2. Please note that our loss-only model and our full model exhibit a higher cross-entropy loss for both training and validation set. This result is expected, since we aim to improve the ergonomic qualities of the synthesized layouts instead of perfectly recreating the distribution of the dataset. A qualitative comparison of scenes synthesized by our models can be seen in Figure 5.9.

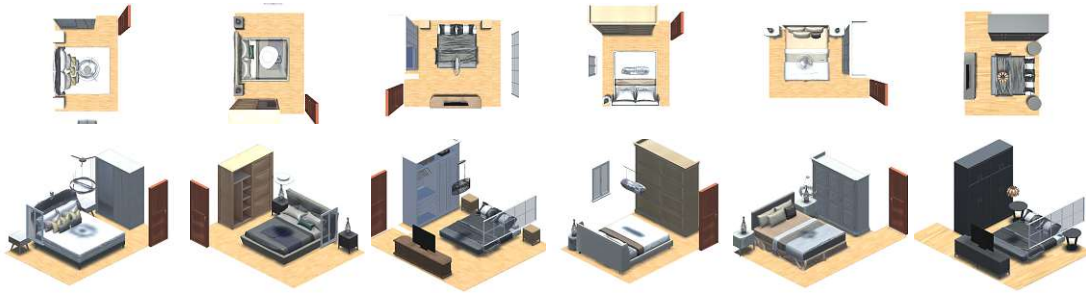
#### 5.4.2 Room-conditioned Layout Synthesis

We use our proposed model and its ablations introduced in the previous section for layout synthesis and evaluate the results in terms of both realism and ergonomic loss. In order to evaluate the realism of our generated results, we perform a perceptual study using Amazon Mechanical Turk in which we ask participants to compare pairs of Bedroom layouts with the question of which layout is more realistic on a 7-point scale. We compare layouts from 6 sources in this study: the ground truth layouts from the 3DFRONT dataset [FJG<sup>+</sup>21, FCG<sup>+</sup>21], layouts generated with our proposed model and its ablations, and another state-of-the-art method ATISS [PKS<sup>+</sup>21], which we train using the code provided on their website, modified to include windows and doors in the same manner as our model. In each layout pair, a synthesized layout is compared to a ground truth layout. A total of 330 users participated in the study. Each pair of layouts was shown 3 times to 10 different users each for a total of 30 comparisons per layout pair.

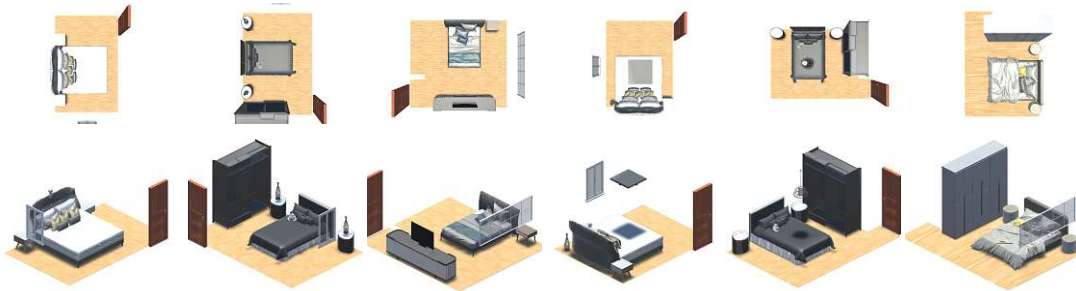
To allow for a direct comparison, we use the binary map of the floor plan and the attributes of the doors and windows from the ground truth data for each layout and only generate the rest of the furniture objects using the selected methods. For each of the 289 layouts in the validation set we generate 20 variations using both ATISS and our trained networks and create a total of 5780 sets of size 6 that contain one layout of each method generated from the same floor plan. Since ATISS does not handle any intersections between furniture objects and even some of the ground truth layouts may contain such intersections, we discard the entire set if one of its layouts contains an intersection between furniture objects larger than a threshold, which we set as 20% of



Results Ours (Full Model): Bedrooms



Results Ours (Baseline): Bedrooms



Results Ours (Weight-Only): Bedrooms



Results Ours (Loss-only): Bedrooms



Figure 5.9: Room-conditioned synthesis results for our model and its ablations.

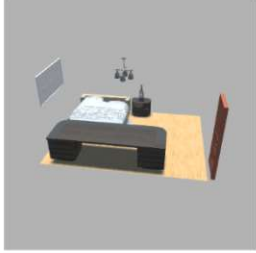
## 5. DATA-DRIVEN DESIGN OF INDOOR LAYOUTS

Below are two furnished bedrooms: **Room A** and **Room B**. We show a rotating view of each, where walls are hidden for better visibility. Carefully judge


- whether the furniture layout in Room A or Room B is more realistic, and
- which of the two rooms you would prefer to live in.

Only take into account the **layout of its furniture, windows, and doors**, not materials, colors, or furniture style.

**Room A**



**Room B**



**Think about how likely you are to encounter such a furniture layout in a real apartment.**

+++	++	+	=	+	++	+++
A much more realistic			equally realistic	B much more realistic		

**Imagine living in these rooms. Which of the two rooms serves its purpose better and in which of the two rooms would you feel more comfortable?**

+++	++	+	=	+	++	+++
A much more preferable			equally preferable	B much more preferable		

Waiting for choice

If you have any questions about this HIT, encounter a problem, or have suggestions on how we could improve it, please let us know:

Figure 5.10: The interface of the user study. Participants were asked which of the 2 displayed scenes is more realistic.

the smaller bounding box area. For our networks, we perform intersection-checks during inference, only discarding a set if an intersection-free layout cannot be generated after 20 attempts. For comparison,  $\sim 75\%$  of ATISS layouts contain bounding box intersections, compared to  $\sim 48\%$  of layouts generated by our model with intersection check turned off. Additionally,  $\sim 56\%$  of the ground truth layouts also contain bounding box intersections. These numbers likely include some false positives where the bounding boxes intersect but the 3d meshes do not. We decided to be more conservative since intersections can significantly influence the perceived realism. Furthermore, since both ATISS and our model may try to generate additional windows or doors, we simple resample the category in such a case.

For the user study, we randomly select 50 sets from all sets of synthesized layouts and ask users to compare the layouts in terms of realism. In each comparison, the user is shown a pair of layouts from the same set, each represented by an animated 3d rendering with the camera rotating around the scene. Users are asked which layout is more realistic on a 7-point scale (ranging from *Scene A much more realistic* to *Scene B much more realistic*, including a neutral *Equally realistic* option). Figure 5.10 shows the screenshot of the UI. Each user sees the same scenes three times, and we use this redundancy to

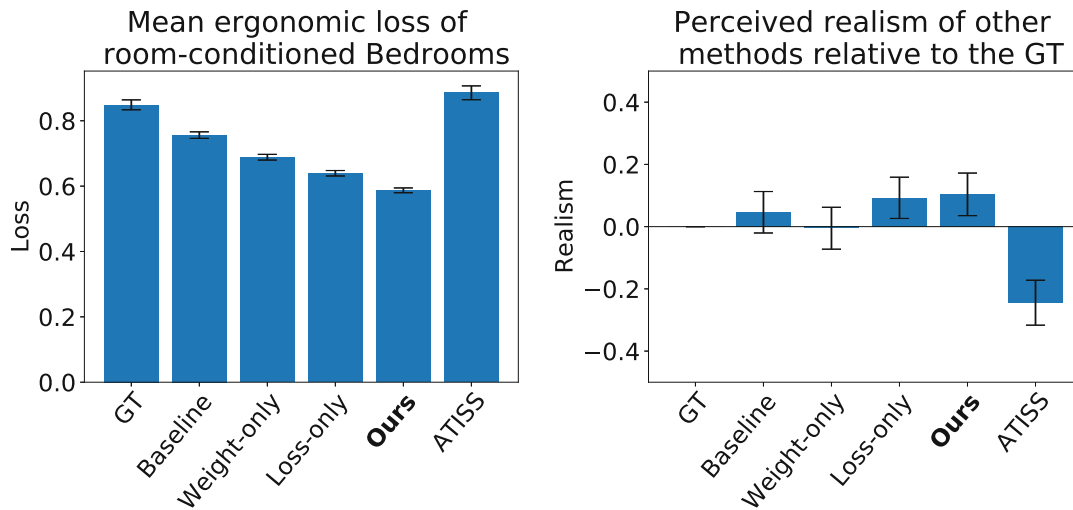


Figure 5.11: Room-conditioned layout synthesis. We synthesize 20 layout variations for each floor plan in the Bedrooms validation set and evaluate the ergonomic loss. The left chart shows the mean ergonomic loss of the synthesized layouts, with the 80% confidence interval of the mean shown in black. The realism of the synthesized layouts is evaluated in a user study. The right chart shows how the layouts synthesized using each method are perceived compared to the ground truth, with a negative value meaning that the ground truth is seen as more realistic. Our proposed approach improves the ergonomic loss of the scenes, while also being perceived as more realistic than the ground truth.

keep track of each user’s consistency. We discard users that chose options more than two points apart for the same scene in more than 10% of their comparisons. Additionally, we discard users that spent less than 10 second on average per comparison. To evaluate the results, we compute a realism score for each method, that we obtain by assigning scores from  $-1$  to  $1$  to the 7 possible user choices and averaging over all comparisons.

The left side of the Figure 5.11 shows the mean ergonomic loss of all layouts created for the user study. As can be seen, our approach performs the best at generating layouts with lower ergonomic loss, reducing the mean ergonomic loss by 30.8% compared to the ground truth data. The ablations of our model also improve the ergonomic loss to a lesser extend, including the baseline model which we attribute to our sampling strategy making it less likely to generate arrangements that are learned from outliers in the training data. On the other hand, layouts created with ATISS show the highest ergonomic loss because the layouts are perceived as less realistic than even our baseline model.

This can be seen on the right side of Figure 5.11 which shows how the users perceive the realism of synthesized layouts compared to those of the ground truth in a range of  $[-1, 1]$ , with a negative value meaning that the ground truth is seen as more realistic. The responses show that ATISS is considered significantly less realistic than the ground truth.

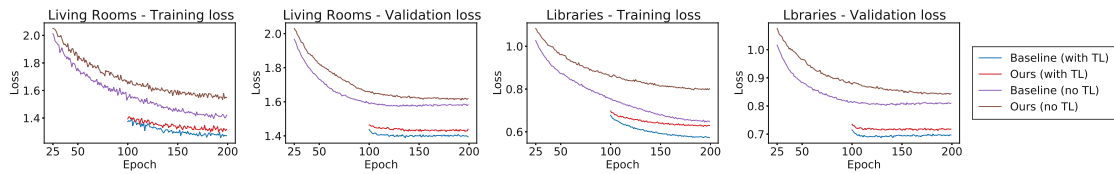


Figure 5.12: By pre-training the network on a general dataset containing samples from all room types and then fine-tuning the network for a specific room type, the validation loss can be decreased significantly, especially for small datasets.

On the other hand, the layouts generated by all our models are seen as at least equally realistic as the ground truth layouts, with users even preferring layouts created with our full model over the ground truth. This shows that our approach can not only improve the ergonomic quality in a purely quantitative sense, but also improve the perceived realism of the layouts.

A qualitative comparison is shown in Figure 5.13. While all of the methods produce plausible layouts, our approach generates, on average, layouts with fewer ergonomic issues like missing light sources or poor accessibility.

### 5.4.3 Additional Room Types

Since some room types in the 3DFRONT dataset only contain few samples (588 living rooms, 554 dining rooms and 424 libraries for training set after our pre-processing), we make use of a transfer learning strategy. We first train a base model containing training samples of all room types for 100 epochs using a learning rate of  $1e-4$ . This base model is then fine-tuned for each room type using a learning rate of  $2e-5$  to prevent overfitting to the smaller datasets.

To evaluate the effectiveness of this approach, we train networks from scratch using only the training data from each individual room category and compare the cross-entropy loss to that of our networks which are first trained on a general set of training data before being fine-tuned for a room category. Figure 5.12 shows that the transfer learning strategy already yields a lower training and validation loss after the first epoch of fine-tuning. While the training loss for networks that are trained from scratch eventually approaches that of the pre-trained network, the validation loss remains higher throughout. As can be seen, for small training datasets, transfer learning proves to be a good strategy for improving the training process.

Figure 5.14 shows some synthesized results of the room types mentioned above in addition to bedrooms. All of the shown results are created using unconditional synthesis, meaning that we do not provide any information about the shape of the room or any furniture objects contained in the scene, so the attributes of the room, windows and doors are also sampled by the trained model.

## 5.5 Discussion

### 5.5.1 Limitations

Our proposed approach has a number of limitations. Designing layouts is a complex high dimensional problem that includes modalities including selecting 3D furniture model that fit well together stylistically [WYA<sup>+</sup>20, LKS15]; architectural elements such as room shapes walls and floor plans [WFT<sup>+</sup>19]; and various other aspects of lighting and illumination conditions [VPGV20]. While important, such methods are orthogonal to our layout synthesis focused scope.

Furthermore, while our ergonomic loss functions are derived from ergonomics literature, they are only theoretical models and have not been evaluated in a real-life setting. We think that the problem of translating the vast number of ergonomic rules and interior design guidelines into differentiable functions to quantify the ergonomic quality of indoor layouts can be a promising topic of further research.

While we have demonstrated that our approach of incorporating expert knowledge into the Transformer training process produces promising results, we think that this is only the first step in combining data-driven and rule-based learning using state-of-the-art deep-learning models such as Transformers. We believe that future research in this direction can assist with making data-driven learning approaches more applicable to domains where large amounts of high-quality data with desired properties are not readily available.

### 5.5.2 Conclusions

In this chapter we presented a novel method for the synthesis of indoor layouts, which combines data-driven learning and manually designed expert knowledge. To our knowledge, we are the first to propose such a solution to the problem. The main benefit of our approach is that it allows emphasizing features that might be underrepresented in the data or might not be contained at all. At the same time, we maintain the benefits of a data-driven approach which is important for layout generation which is high-dimensional and ill-defined. Manually crafting all design rules needed to synthesize comparable results would be very difficult and time consuming. Hence, combining both expert knowledge and a distribution learned from data gives us the benefits from both worlds.

As a technical contribution, we proposed a modern Transformer network that can be trained using a loss function composed of cross-entropy and additional knowledge. We have shown that weighting the two loss terms on a per-sample basis leads to results that fulfill the additional objective well and still maintain a high degree of realism. Further, we introduced expert knowledge in the form of cost functions derived from ergonomics, whose goal is to improve layouts to be more usable and comfortable for humans.

We described the details of our implementation (our code is available on GitHub), and we evaluated the method thoroughly. We showed numerical quantitative results and

performed a perceptual study (with 330 participants on Amazon Mechanical Turk) where our model out-performs recent related work. We also used our system to synthesize a large set of realistically looking results. Our method is meant to help professionals and amateurs in the future to address the problem of interior layout design.

### Acknowledgments

This research was funded by NJIT's faculty startup funds, gifts from Adobe, and by Austrian Science Fund FWF P29981. In addition, the authors would like to thank Stefan Pillwein for his help with paper production.

## Results Ours (Full Model): Bedrooms



## Results Ours (Baseline): Bedrooms



## Results ATISS: Bedrooms

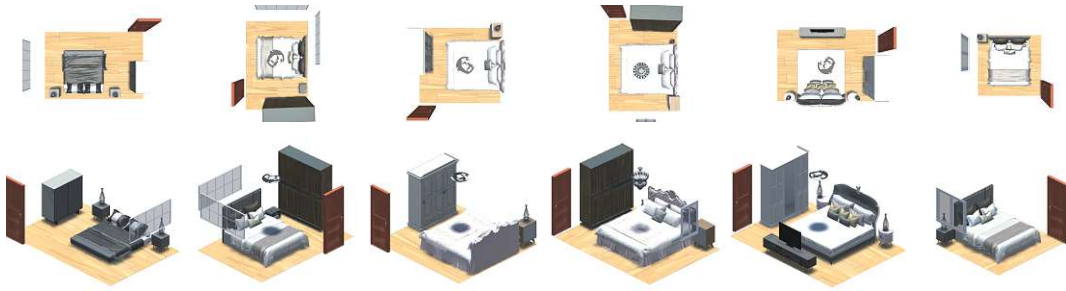


## Ground Truth: Bedrooms



Figure 5.13: Conditional synthesis results as described in Section 5.4. All methods in a column receive the same room boundary, windows, and doors as input condition. Our approach (top row) produces on average layouts with less ergonomic issues like missing light sources (e.g. missing chandeliers in Baseline columns 1, 3 and 6) or poor accessibility (e.g. chair and wardrobe blocking the path in ATISS column 5 and 6).

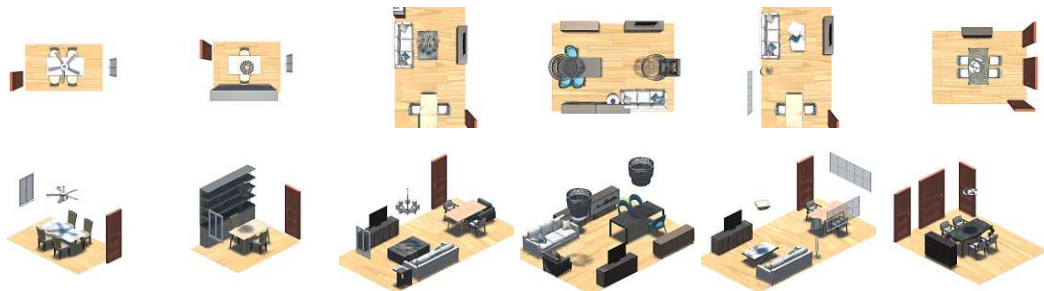
Results Ours: Bedrooms



Results Ours: Living Rooms



Results Ours: Dining Rooms



Results Ours: Libraries

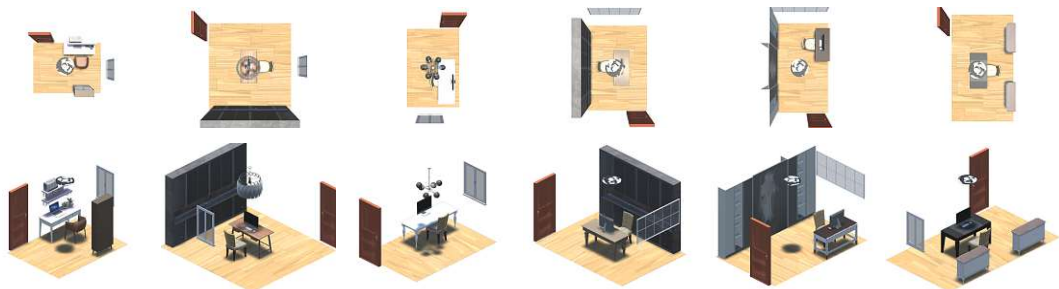


Figure 5.14: Layouts created by our models for different room types. Since the attributes of the rooms were represented as part of the input sequences during training, all layout elements including rooms, doors, and windows can be generated by the network. Our method can generate furniture arrangements typical for each room type even with small training sets.



# Conclusion

## 6.1 Summary

In Chapter 3 we presented a method for the interactive design of body-supporting surfaces driven by a given pose of the human body as well as the pressure distribution on the body's surface that is computed on-the-fly using our proposed algorithm. Our method is intended to help designers create appropriate surfaces digitally without requiring additional empirical design passes and fabrication of physical prototypes on the one hand, and to ensure physical plausibility on the other hand. Further, due to simplifications made during the pressure computation, we achieve interactive rates in the range of a few seconds, which is a necessity for interactive design applications.

The main contribution of this work is an interactive modeling system that utilizes captured body poses and computes an importance field that is proportional to the pressure distribution on the body for a given pose. This distribution indicates where the body should be supported in order to easily hold a particular pose, which is one of the measures of comfortable sitting. We evaluated our proposed approach through tests conducted with the help of design students and presented a number of results from these sessions. We furthermore demonstrated the physical viability of the surfaces designed with our approach by fabricating a prototype of one such design.

To address one of the drawbacks of the above approach, namely the requirement of a manually designed initial surface, we presented an automated computational framework for the generation of functional body supporting furniture in Chapter 4. Similar to the approach described in Chapter 3, we utilize the pressure distribution on the body as measure of comfort while sitting, but furthermore consider the joint moments acting on the body as an additional factor. The computation of the pressure distribution is also extended by incorporating frictional forces for increased accuracy. Our proposed computational method achieves much faster computation times compared to sophisticated FEM methods while still providing adequate accuracy, as demonstrated in our evaluation.

Additionally, we proposed an algorithm for automated generation of generic template meshes that utilizes the given comfort measure to optimally support the body. Since the resulting surface is dependent only on the input pose, it serves as an easy-to-use tool even for novice users or can be used by experts to generate an initial design for further editing in a manner of seconds. Our method is intended for computer-aided design of personalized furniture that can be then used for fabrication with modern digital manufacturing methods.

Shifting focus from individual pieces of furniture to entire arrangements, we presented a novel method for the synthesis of indoor layouts in Chapter 5, combining data-driven learning with manually designed rules based on expert knowledge such as ergonomics. Since the problem of layout synthesis is high-dimensional and ill-defined, we utilize a data-driven approach that allows us to learn the most common design rules by example, eliminating the effort of manually handcrafting each individual rule. At the same time, we can emphasize features that might be underrepresented or even completely absent in the data by defining just a small set of design rules, in our case based on ergonomics, with the goal of improving the learned representation of the data. Hence, combining both expert knowledge and a distribution learned from data gives us the benefits from both worlds.

As a technical contribution, we proposed a deep learning approach based on the modern Transformer network architecture that extends the commonly used cross-entropy loss function with additional differentiable cost functions based on expert knowledge. In our case, we derived these cost functions based on ergonomics literature with the goal of improving the ergonomic qualities of the synthesized layouts. We have demonstrated that weighting the two loss terms on a per-sample basis leads to results that fulfill the additional objective well while at the same time still maintaining a high degree of realism.

### 6.2 Outlook and Future Work

While we have made great strides towards ergonomics-driven computational design of furniture and indoor layouts with the research presented in this thesis, there is still much opportunity for future work in this direction. In Chapters 3 and 4 we presented approaches for the interactive and automated design of seating furniture based on human poses. One of the primary requirements of these methods was to achieve low computation times that allow for interactive design. As such, we have made several simplifications to the computation of the pressure distribution on the body that is used to guide the surface optimization process, such as the assumption that both the human body and the seating surface behave like rigid objects. While this assumption yields adequate accuracy for general purpose design of seating furniture, as we have shown in our evaluations, an even higher degree of accuracy might be required in other fields, such as medical applications. However, due to the complexity of the human body, accurate simulation would be infeasible for interactive design purposes, as computations times range in the span of several hours even with modern FEM methods. One possible direction for solving

this problem could be to allow for various degrees of accuracy during different stages of the design process, starting with fast but inaccurate simulation for the early design phase where many modifications are made, up to slow but accurate simulation in the final stages where fine-tuning of the surface becomes the priority.

Additionally, our methods only consider seating furniture made from stiff material and thus ignore the possibility of surface deformation during use. For the design of surfaces made from soft materials, such as sofas, bean chairs or inflatable furniture, the deformation behavior of the material needs to be considered not just for functional purposes (comfort while sitting) but also aesthetic reasons. However, considering soft materials in the surface optimization process adds a high degree of complexity, as the deformation behavior needs to be considered under different load distributions. The deformation of the surface also influences how pressure is distributed on the body, further increasing the computational cost of the optimization process. Furthermore, the structural properties of the furniture also become an important issue, as soft materials are less resistant to wear and tear and may become damaged more easily over longer periods of use. These additional layers of difficulty make the task of designing furniture made of soft materials an interesting topic for future research.

In Chapter 4, we presented a method for the automated generation of seating furniture that depends only on the input pose. This eliminates much of the manual effort needed to create an initial design that is suitable for the given pose, but there is still much room for further improvement. While the fixed template we use for the generation of the surface is well suited for most common sitting poses, it offers limited flexibility and cannot support less usual poses like lying sideways. Making the template generation more flexible and adaptable to the input pose, possibly including the ability to create multiple design suggestions per pose, is a challenging problem that could greatly expand the versatility of our design tool. There is also the possibility of developing a hybrid method in which the template model can be interactively modified by the user, which would require a suitable parametrization of the template. While this approach adds some manual effort back into the design process, it has the advantage of offering more control over the initial design.

Furthermore, there is the limitation that our template model can only be used to synthesize seating furniture for one single person. While it is possible to create a surface that is usable by multiple people through manual effort, by first creating individual objects and then combining them through interactive editing, this solution is far from ideal. There are different ways to approach this problem, each with their own unique challenges. A straightforward approach would be an algorithm that can automatically combine multiple surfaces into a single object, but it may be difficult to achieve this in a robust manner that allows handling of arbitrary furniture configurations as input, since each surface needs to be modified depending on the number and relative positions of the other input surfaces. Another possible method would be to extend the template model to be able to support multiple input poses, either in an automated manner or by interactive editing, as mentioned above. This approach might be more robust, but also makes the

problem of finding a suitable parametrization of the template even more challenging.

The data-driven layout synthesis approach described in Chapter 4 proposes the idea of incorporating additional manually defined rules based on expert knowledge into the training process of a deep-learning model. While we have demonstrated the viability of this approach, there is much room for further improvement, especially from a technical perspective. We have focused on models based on the Transformer architecture since recent research has demonstrated that it is well suited for layout synthesis tasks in particular. Within the constraints of this choice, there exist the options of exploring alternative ways of representing indoor layouts or using different distributions to model the furniture parameters, possibly offering even better opportunities to integrate expert knowledge into the learning process. Outside of these constraints, one may look at the possibility of utilizing other types of deep-learning models, such as autoencoders, convolutional neural networks or more general graph neural networks.

While we have shown that the integration of expert knowledge into the learning process works well for the task of indoor layout synthesis, we believe that this approach can offer benefits for other applications as well. Deep learning has been successfully applied to a vast variety of tasks, including synthesis of images and audio, detection and recognition of people, objects or text in visual data, navigation of vehicles and robots, prediction of physical phenomena, and much more. The necessity of large amounts of high-quality training data is a common problem in machine learning — for some applications, this can be alleviated through generation of synthetic training data or augmentation of existing examples, but this may not be feasible for all types of data. While we cannot guarantee that our approach can be universally adapted to every possible task, we believe it can serve as a useful alternative for some applications where synthesis or augmentation of training data is not possible.

Finally, while the work presented in this thesis focuses on computational design of furniture and its arrangements, the algorithms and tools described in Chapters 3 and 4 do not directly interact with those of Chapter 5. Yet, there are applications that could greatly benefit from a combined approach. As an example, when designing for a library, seating furniture should provide both privacy and optimal lighting conditions to allow visitors to focus on the reading experience. On the other hand, seating designed for a sports bar should enable conversation while at the same time offering a good view of the displays installed in the establishment. These constraints need to be taken into account when designing both the layout and individual pieces of furniture.

A simple way to achieve this would be to bring information from one domain into the other, for example by first synthesizing a general layout and then designing each individual piece of furniture, taking into account how the ergonomic cost of the layout can be improved by selecting and positioning the input poses used to synthesize and optimize the surface. One may also consider the opposite approach of taking particular properties of designed seating furniture into account during the evaluation of a layout. Ideally though, the flow of information should be bidirectional, so that both the furniture and layout design tasks can benefit from each other. While it is difficult to consolidate

an interactive design process with the training stage of a deep learning model, there exists the possibility of utilizing the inference stage of the model for a more interactive approach. Since our deep learning model is capable of rearranging or making additions to partial layouts, it is possible to pass information between the model and the designer in an iterative process. While our current model is limited in this regard by the small number of attributes used to represent each individual piece of furniture, building upon this approach could offer new opportunities for the design of furniture and indoor layouts in the future.



# Appendix

## A.1 Surface Fitting Details

For simplicity we will explain how to solve the optimization problem starting with point-to-point distances only. Given a mesh  $M^s$  of subdivision level  $s$ , the vertices of the next subdivision level  $s + 1$  can be computed using the subdivision matrix  $\mathbf{S}^s$ :

$$\mathbf{V}^{s+1} = \mathbf{S}^s \mathbf{V}^s,$$

or, starting from the initial control mesh  $M^0$  using

$$\mathbf{V}^{s+1} = \mathbf{S}^s \mathbf{S}^{s-1} \dots \mathbf{S}^1 \mathbf{S}^0 \mathbf{V}^0 = \bar{\mathbf{S}}^s \mathbf{V}^0.$$

Using the matrix  $\bar{\mathbf{S}}^s$ , it is possible to compute the optimal control mesh vertices  $\mathbf{V}^*$  by solving a system of linear equations

$$\hat{\mathbf{S}}^s \mathbf{V}^* = \mathbf{V}^b,$$

where  $\hat{\mathbf{S}}^s$  is the matrix whose  $k$ th row is the row of  $\bar{\mathbf{S}}^s$  corresponding to the vertex  $\mathbf{v}_k^s$ , and  $\mathbf{V}^b$  is the matrix containing the sample point coordinates  $\mathbf{v}_k^b$  of the body mesh in each row.

To modify this approach to work with the tangent distance, we need to separate the x-, y- and z-coordinates. Let

$$\mathbf{x}^* = \begin{bmatrix} \mathbf{V}_1^* \\ \mathbf{V}_2^* \\ \mathbf{V}_3^* \end{bmatrix}$$

denote the column vector of concatenated x-, y- and z-coordinates of  $\mathbf{V}^*$  and let  $\hat{\mathbf{N}}$  denote the matrix whose  $k$ th row contains coordinates of normal vector  $\mathbf{n}_k$ . Let  $\hat{\mathbf{N}}_1$  denote the

matrix containing the x-coordinates of the normals  $\mathbf{n}_k$  in its diagonal, with  $\hat{\mathbf{N}}_2$  and  $\hat{\mathbf{N}}_3$  defined analogically. Then we can solve the following system of linear equations to find  $\mathbf{x}^*$ :

$$\begin{bmatrix} \hat{\mathbf{N}}_1 \hat{\mathbf{S}}^s & \hat{\mathbf{N}}_2 \hat{\mathbf{S}}^s & \hat{\mathbf{N}}_3 \hat{\mathbf{S}}^s \end{bmatrix} \mathbf{x}^* = \hat{\mathbf{N}}^T \mathbf{V}^b.$$

The full system of equations, including the parameter  $\tau$  to blend between point-to-point and tangential distance, the Laplacian  $\mathbf{L}$  as a smoothing term controlled by weighting parameter  $\sigma$ , and the weights  $\gamma$  that allow constraining vertices to their original location is given by

$$\begin{bmatrix} (1-\tau)\mathbf{W}\hat{\mathbf{S}}^s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (1-\tau)\mathbf{W}\hat{\mathbf{S}}^s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & (1-\tau)\mathbf{W}\hat{\mathbf{S}}^s \\ \tau\mathbf{W}\hat{\mathbf{N}}_1\hat{\mathbf{S}}^s & \tau\mathbf{W}\hat{\mathbf{N}}_2\hat{\mathbf{S}}^s & \tau\mathbf{W}\hat{\mathbf{N}}_3\hat{\mathbf{S}}^s \\ \sigma\mathbf{L} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma\mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma\mathbf{L} \\ \gamma\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \gamma\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \gamma\mathbf{I} \end{bmatrix} \mathbf{x}^* = \begin{bmatrix} (1-\tau)\mathbf{W}\mathbf{V}_1^b \\ (1-\tau)\mathbf{W}\mathbf{V}_2^b \\ (1-\tau)\mathbf{W}\mathbf{V}_3^b \\ \tau\mathbf{W}\hat{\mathbf{N}}^T\mathbf{V}^b \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \gamma\mathbf{V}_1^0 \\ \gamma\mathbf{V}_2^0 \\ \gamma\mathbf{V}_3^0 \end{bmatrix},$$

with  $\mathbf{W}$  containing the vertex weights  $\rho_k$  in its diagonal.

## A.2 Jacobians for Inverse Kinematics

In our kinematic model, each physical joint  $j$  has three rotational degrees of freedom (Euler angles) for rotations about the local frame axes and therefore three corresponding entries in the joint angle vector  $\boldsymbol{\theta}$ . Only the hip joint, representing the root of all other joints, has three additional translational degrees of freedom for translations of the whole body in global  $x$ ,  $y$  and  $z$  direction.

In order to explain how to compute each entry  $J_{i,k}^{d_p}$  expressing the change of the value  $E_i^{d_p}$  w.r.t. the parameter  $\theta_k$ , we have to introduce the function  $\delta(j,k)$  that returns 1 if the joint  $j$  is influenced by the parameter  $\theta_k$  and 0 otherwise. Note that a joint  $j$  is influenced by a parameter  $\theta_k$  not only when  $\theta_k$  directly corresponds to  $j$ , but also when  $j$  is a direct or indirect child of the joint the parameter  $\theta_k$  corresponds to. Finally, the function  $w(i,j)$  returns the skinning weight from the joint  $j$  on the vertex  $\mathbf{v}_i^b$  and  $\mathbf{u}(i,j)$  returns the contribution point for the vertex  $\mathbf{v}_i^b$  from the joint  $j$  in the skinning process.

Let  $\mathbf{a}_k$  be the axis and  $\mathbf{p}_j$  be the position of a rotational joint  $j$  corresponding to parameter  $\theta_k$ . The computation of the derivative of the position of the vertex  $\mathbf{v}_i^b$  w.r.t. the parameter  $\theta_k$  then comes down to

$$\frac{\partial \mathbf{v}_i^b}{\partial \theta_k} = \mathbf{a}_k \times \left( \sum_{j=1}^{n_\theta} \delta(j,k) w(i,j) (\mathbf{u}(i,j) - \mathbf{p}_j) \right).$$



If  $\theta_k$  corresponds to a translational joint with translation axis  $\mathbf{a}_k$  (in our case only the hip joint), the derivatives can be computed significantly easier with

$$\frac{\partial \mathbf{v}_i^b}{\partial \theta_k} = \mathbf{a}_k.$$

Finally, we compute the entries for all Jacobians according to

$$\begin{aligned} J_{i,k}^{d_p} &= \frac{\partial e_i^{d_p}}{\partial \theta_k} = \frac{r_i}{\|\mathbf{v}_i^b - \mathbf{v}_i^s\|} \left( (\mathbf{v}_i^b - \mathbf{v}_i^s)^T \frac{\partial \mathbf{v}_i^b}{\partial \theta_k} \right), \\ J_{i,k}^{d_t} &= \frac{\partial e_i^{d_t}}{\partial \theta_k} = r_i \left( \mathbf{n}_i^s \cdot \frac{\partial \mathbf{v}_i^b}{\partial \theta_k} \right), \\ J_{i,k}^p &= \frac{\partial e_i^p}{\partial \theta_k} = - \frac{a \cdot r_i \left( \mathbf{n}_i^s \cdot \frac{\partial \mathbf{v}_i^b}{\partial \theta_k} \right)}{\pi \left( (a \cdot (\mathbf{n}_i^s \cdot \mathbf{v}_i^b - d_i^s) + b)^2 + 1 \right)}. \end{aligned}$$

### A.3 Reaction Force Computation

The equilibrium constraints of our reaction force computation model can be formulated as the matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_b^F & \mathbf{0} & v^R \alpha_b^R \mathbf{T}^R \\ [\mathbf{a}]_b^F & \mathbf{I}_b^M & v^R \alpha_b^R [\mathbf{a}]_b^R \mathbf{T}^R \\ \mathbf{I}_j^F & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_j^M & \mathbf{0} \end{bmatrix},$$

with  $\mathbf{I}_b^F$  (respective  $\mathbf{I}_b^M$ ) being the  $3 \times 3$  identity matrix if the force  $\mathbf{f}$  (respective moment  $\mathbf{m}$ ) corresponding to the column is active inside the body segment  $b$  corresponding to the row, and  $\mathbf{0}$  otherwise. Similarly,  $\mathbf{I}_j^F$  (respective  $\mathbf{I}_j^M$ ) is the  $3 \times 3$  identity matrix if the force  $\mathbf{f}$  (respective moment  $\mathbf{m}$ ) corresponding to the column is active on the joint  $j$  corresponding to the row, and  $\mathbf{0}$  otherwise.  $\mathbf{T}^R$  denotes the transformation matrix from tangent space to world space. Finally,  $[\mathbf{a}]_b^F$  (respective  $[\mathbf{a}]_b^R$ ) denotes the skew-symmetric matrix

$$[\mathbf{a}]_b = \begin{bmatrix} 0 & a_3 & -a_2 \\ -a_3 & 0 & a_1 \\ a_2 & -a_1 & 0 \end{bmatrix},$$

if the force  $\mathbf{f}$  (respective reaction force  $\mathbf{r}$ ) corresponding to the column is active inside the body segment  $b$ , or  $\mathbf{0}$  otherwise. The scalars  $\alpha_b^R$  are the actual linear blend skinning weights of each vertex-body segment connection, and the scalar values  $v^R$  are additional user provided weights, which allow to further control the importance or unimportance of particular surface regions. In particular, we use them to exclude body parts like face, chin, or parts of the abdomen.

The vector  $\mathbf{a}$  denotes the moment arm vector—the vector pointing from the COM of  $b$  to the point affected by the force  $\mathbf{f}$  (or reaction force  $\mathbf{r}$ ) corresponding to the column. Finally, the right side of the system,  $\mathbf{z}$ , is a  $(6n_B + 6n_J)$  column vector:

$$\mathbf{z} = \begin{bmatrix} -\mathbf{g}_b \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

The terms of the energy function can be formulated as matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I}^M & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \lambda v^R \mathbf{W}^R \mathbf{T}^R \end{bmatrix},$$

where  $\mathbf{I}^M$  is the identity matrix with  $3n_M$  rows and  $\lambda \geq 0$  is a weight that assigns more or less importance to the minimization of the moments, which can be interpreted as the stiffness of the joints.  $\mathbf{W}^R$  contains the weights that determine how the reaction forces are distributed on the surface and also acts as a regularizer, as without it the best solution is likely to have a very small number of extreme reaction forces.

## A.4 Surface Optimization Gradient

In order to improve the efficiency of the solving the surface optimization problem, we evaluate the gradient of the corresponding energy function.

*Laplacian smoothing distance:* To compute the gradient value for the Laplacian smoothing distance metric, we first rewrite the corresponding term as:

$$S_L = \sum_i^{n^v} \sum_k^{x,y,z} \left( \mathbf{v}_{ik} - \frac{\sum_{j \in N_1(i)} \mathbf{v}_{jk} w_j}{\sum_{j \in N_1(i)} w_j} \right)^2 \lambda_S^l$$

For each vertex, each dimension can be computed separately as the squared difference from the average position of its neighbors. To evaluate the gradient, we need to compute the partial derivatives for each variable of the objective function (i.e. the x, y and z coordinates of each vertex position). For the x coordinate of an arbitrary vertex (with its 1-ring neighborhood  $N_1(i)$ ), its respective part of the gradient value is computed as.

$$\frac{\partial S_L}{\partial x} = 2\lambda_S^l \left( x - \frac{\sum_{j \in N_1(i)} x_j w_j}{\sum_{j \in N_1(i)} w_j} \right)$$

For each variable, we also have to consider its occurrence as neighbor of another variable. Therefore, for each variable's gradient value, we also accumulate the following term:

$$\frac{\partial S_L}{\partial x_{n1}} = \frac{-2\lambda_S^l w_{n1} \left( x - \frac{\sum_{j \in N_1(i)} x_j w_j}{\sum_{j \in N_1(i)} w_j} \right)}{\sum_{j \in N_1(i)} w_j}$$

*Angle based differences:* To compute the gradient value of a variable corresponding to the angle based differences, we consider its occurrences in the respective computations. For each face, its four interior angles are considered. To compute an angle, the corresponding vertex and the edges to its two adjacent vertices are required. Each vertex is adjacent to multiple faces in the surface mesh. This means, to compute the (angle based) gradient value for a variable, we need to consider the angles of all adjacent faces of the vertex. For each face, a vertex position is relevant for three interior angles. The respective parts of the gradient values are computed as follows (for a single variable):

$$\begin{aligned} \frac{\partial \alpha_{abc}}{\partial a_x} &= -\frac{\frac{\mathbf{e}_{bc,x}}{m_{bc}m_{ab}} - \frac{\mathbf{e}_{ab,x}S_{abc}}{m_{bc}m_{ab}^3}}{\sqrt{1 - \langle \mathbf{t}_{ab}, \mathbf{t}_{bc} \rangle}} \\ \frac{\partial \alpha_{dab}}{\partial a_x} &= -\frac{\frac{\mathbf{e}_{da} + \mathbf{e}_{ab}}{m_{ab}m_{bc}} - \frac{\mathbf{e}_{da,x}S_{dab}}{m_{ab}^3m_{da}} + \frac{\mathbf{e}_{ab,x}S_{dab}}{m_{ab}m_{da}^3}}{\sqrt{1 - \langle \mathbf{t}_{da}, \mathbf{t}_{ab} \rangle}} \\ \frac{\partial \alpha_{cda}}{\partial a_x} &= -\frac{\frac{-\mathbf{e}_{cd,x}}{m_{cd}m_{da}} - \frac{\mathbf{e}_{da,x}S_{cda}}{m_{cd}m_{da}^3}}{\sqrt{1 - \langle \mathbf{t}_{cd}, \mathbf{t}_{da} \rangle}} \\ S_{abc} &= \sum_i^{x,y,z} (b_i - a_i)(b_i - c_i) \end{aligned}$$

- $a, b, c$  and  $d$  are the corner vertices of the face.
- $\mathbf{e}_{ab} = (\mathbf{b} - \mathbf{a})$  refers to the edge between vertices  $a$  and  $b$ .
- $m_{ab} = \|\mathbf{e}_{ab}\|_2$  is the magnitude, i.e. the euclidean norm of the edge vector.
- $\mathbf{t}_{ab} = \frac{\mathbf{e}_{ab}}{m_{ab}}$  is the normalized edge direction vector.

The gradient values for the face interior angles are accumulated for each variable and utilized separately for the *regular faces* and *maximum angles* error metrics.

*Vertex distance:* The vertex distance error metric from the objective function's data term can be written as

$$D_V = \lambda_D^V \sum_{i=1}^{n^V} \left( (\mathbf{v}_{ix} - \tilde{\mathbf{v}}_{ix})^2 + (\mathbf{v}_{iy} - \tilde{\mathbf{v}}_{iy})^2 + (\mathbf{v}_{iz} - \tilde{\mathbf{v}}_{iz})^2 \right) w_i$$

The corresponding part of the gradient value for each variable is simply computed from its partial derivative:

$$\frac{\partial D_V}{\partial \mathbf{v}_{ix}} = 2\lambda_D^V w_i (\mathbf{v}_{ix} - \tilde{\mathbf{v}}_{ix})$$

*Plane distance:* The data term's plane distance metric is computed for the four corner vertices for each face that corresponds to a supporting plane in the original model. Vertices that are adjacent to more than one of these planes, have multiple plane distance values. Therefore, the gradient values for each variable has to be accumulated for each face. A variable's value corresponding to a single face  $F_j$  (with  $\mathbf{v}_i \in F_j$ ) is computed as:

$$\frac{\partial D_P}{\partial \mathbf{v}_{ix}} = 2\lambda_D^P w_j \mathbf{n}_x^{P_j} \langle \mathbf{v}_i - \mathbf{c}^{P_j}, \mathbf{n}^{P_j} \rangle$$

The gradient values for all error metrics are further scaled by the corresponding global factors for the data and smoothing terms. The objective function's gradient evaluation results in vector of length  $3n^V$ , where each value corresponds to the sum of the error metric gradient values for an individual variable.

# List of Figures

1.1	The design of furniture and indoor spaces is a difficult task that requires expert knowledge from a variety of disciplines. In this thesis, we present different computational design approaches that incorporate several aspects of the design task, including ergonomic, functional or aesthetic requirements. Left: A fabricated multi-purpose seating surface designed for various types of sitting and leaning using the interactive approach described in Chapter 3. Middle: Design for a seating surface that was synthesized for the given input pose using the automated method presented in Chapter 4. Right: A living room furniture layout synthesized using the data-driven approach described in Chapter 5. . . . .	2
2.1	Examples of related work on the topics of physically-informed and pose-driven design. Left: An interactive tool by Umetani et al. [UIM12] provides suggestions for the design of stable furniture. Right: The approach proposed by Zheng et al. [ZLDM16] allows reshaping of existing 3D models to fit the dimensions of an input pose. Images sourced from the respective papers. .	10
2.2	Indoor scenes synthesized by existing transformer-based deep learning models which are closely related to our approach described in Chapter 5. Left: Sceneformer [WYN20]. Right: ATISS [PKS <sup>+</sup> 21]. Images sourced from the respective papers. . . . .	12
3.1	Multi-purpose sitting surface designed for various types of sitting and leaning using our method. Left: three renderings. Right: fabricated result inspected by design students. . . . .	16
3.2	Overview of the interactive design system. Given initial design and a set of poses captured by a motion capture device, our system estimates a pressure distribution on the bodies in the given poses. The artist can then create a social scenario using the given poses and provide a initial control mesh for a surface. Our system then computes an optimized smooth subdivision surface and its control mesh using our surface fitting algorithm. In further design steps, the computed control mesh can be edited interactively and used as input again to generate a new design. Our pose relaxation algorithm also makes it possible to adapt the input poses to the computed subdivision surface if necessary to ensure that all poses can be supported well. . . . .	16
		93

3.3	Illustration of the forces acting on a rigid body. Left: reaction forces split into the normal and tangential components. Right: definition of pressure as normal force acting per unit area. If the same forces are acting on a larger area, the pressure is lower. . . . .	18
3.4	Results of physical simulation and comparison to ours. Top row: sitting pose—please note that for visualization purpose we render the results on a T-pose. Bottom row: lying T-pose. Please notice the different range for the error image. . . . .	19
3.5	Pose relaxation using inverse kinematics. The designer can further relax the input poses in order to let them to adapt to the current surface (please refer to Section 3.5) . . . . .	23
3.6	Design process. Left: recording of poses using a motion capture device. Center: interactive design using RHINOCEROS and GRASSHOPPER. Right: GRASSHOPPER canvas. . . . .	24
3.7	Variations using the control mesh shown in Fig. 3.7a created with varying values of the smoothing parameter $\sigma$ which weighs the Laplacian operator. Using this parameter the designer can balance between the importance of the input body map and smoothness of the surface. Top row: arms of the input body have not been considered to be supported. Middle row: arms are supported. Bottom: contact area and pressure on the seat. Refer to Section 3.4 for more details. . . . .	26
3.8	Chair optimized for 2 persons with different body types. Top: optimized for thin person. Bottom: optimized for overweight person. . . . .	27
3.9	Several results created by design students using our method. The top row shows three design variations using the same input poses achieved by fixing different control vertices oder changing design parameters. Please refer to Section 5.4 for more details. . . . .	28
3.10	Two additional designs of sitting landscapes for multiple persons. . . . .	29
3.11	A series of designs using asymmetric input poses created with the control mesh shown in Fig 3.7a. . . . .	30
4.1	Left: seating surface generated by our automated algorithm and suited to support the input pose . Right: the surface fitting algorithm described in Chapter 3 applied to the generated surface. . . . .	34
4.2	Left: a moment $\mathbf{m}_1$ as a cross product of the moment arm $\mathbf{a}_1$ and the force $\mathbf{g}_1$ . Body segments have anatomical values (e.g., $\mathbf{g}_1$ ) assigned from [PEA83]. Middle: a link-segment-skeleton with 21 segments and a polygonal surface mesh. Right: The mesh is rigged using linear blend skinning [MTLT89].	36
4.3	Physics of sitting: if contact with a support surface is given on the buttocks, back, and feet, the moments of the body are minimized and the forces are in equilibrium. If the contact on the feet is lost, the contact to the back is lost automatically due to the missing friction force on the feet. . . . .	37

4.4	Computational human body model. Left: simplified friction model, right: free-body diagram of the skeleton model. Please refer to Section 4.2.2 for the details. . . . .	40
4.5	Comparison to the FEM (cf. Section 4.2.2). Left: results of a lying T-pose. Right: a sitting pose. Please note that we plot the pressure and shear distributions of the sitting pose on a T-pose mesh for better visualization purpose. . . . .	41
4.6	Body part mapping in the advanced template model. The rows of the model are mapped to individual body parts. Within a row, the segment in each column is mapped to a subset of the corresponding body vertices. The leg segments are mapped independently to the corresponding body parts. . . . .	44
4.7	Plane fitting and mesh generation process. Left: A plane is fitted to the body segment marked in red. The vectors show the local coordinate system on the plane consisting of plane normal vector $\mathbf{n}^P$ , the projection of skeleton bone direction $\mathbf{d}^{body}$ onto the plane $\mathbf{d}^{up}$ and the vector $\mathbf{d}^{side}$ which is orthogonal to both. Right: Template mesh during the third step of the mesh generation process. Row and column intersection lines $\mathbf{l}^{ir}$ and $\mathbf{l}^{ic}$ are shown in yellow and magenta respectively. . . . .	46
4.8	Error handling on a pose where one foot rests on top of the other. Left: surface model generated without error handling showing intersections. Right: corrected surface mesh. . . . .	48
4.9	Finalized seating surface results after the optimization process. Left: seating surface before optimization. Center/Right: results after optimization . . . . .	49
4.10	Results of our method. Left: control mesh generated with our method. Center: the fitting algorithm described in Chapter 3 applied to our control mesh. Right: the same fitting algorithm applied to a flat patch serving as the control mesh. . . . .	50
4.11	Forward-leaning poses and leaning back while standing are also supported by our method. . . . .	53
4.12	Our method fails to generate a valid surface from unsupported poses like lying on the side or placing the feet underneath the body. . . . .	53
4.13	Mapping of the contact pressure of Poses 1-6 onto their corresponding generated surfaces. Each column corresponds to one pose. Red color indicates high pressure values. . . . .	55
4.14	Additional results generated using our algorithm. The surface in the bottom row is created by manually editing and combining multiple control meshes. . . . .	56
5.1	Our proposed LayoutEnhancer combines data-driven learning from potentially imperfect data with expert knowledge. Generated layouts are biased to follow rules laid out in the expert knowledge, effectively reducing the impact of data imperfections. See Figure 5.2 for examples of imperfections that are avoided due to the inclusion of expert knowledge. . . . .	58
		95

5.2	LayoutEnhancer can learn to improve issues found in imperfect data like <i>ergonomic issues</i> (left room): (i) a window directly behind the TV causes glare on sunny days, making it difficult to watch due to a big contrast in brightness. (ii) Insufficient illumination for reading a book without a light source behind or beside the bed; and <i>geometric issues</i> (right room): (i) desk is intersecting with the bed and the closet; (ii) closet is covering the door.	58
5.3	Ergonomic rules implemented in our system. We chose these guidelines as they are essential in most indoor scenarios, like reading a book, watching TV, or working at the desk or the computer. We convert the rules to scalar cost functions and evaluate them using activities (cf. Section 5.2).	59
5.4	Human activity in the room based on the example of <i>Watch TV</i> . For all possible sitting locations $p_j$ an avatar is sampled and the ergonomic rules for visibility and glare are evaluated. The final contribution is the weighted sum of costs over every combination of a sitting possibility $p_j$ and all TVs $q_k$ . Please refer to Section 5.2 for more details.	60
5.5	A layout is represented as a sequence $S = (s_1, \dots, s_n)$ . Each individual token $s_i$ in the sequence represents an attribute of a furniture object, such as its category, orientation, position or dimensions.	65
5.6	To propagate the ergonomic loss back to the token probabilities, we choose the maximum of the discrete values of the predicted token and convolve the neighborhood with a Gaussian kernel, centered at the discrete maximum. The resulting token value is a weighted sum of the discrete values in this neighborhood, weighted by the probability and distance to the kernel center of each discrete value. Please refer to Section 5.3.2 for more details.	66
5.7	Overview of our model. A room layout consisting of individual furniture objects is mapped to a sequence of tokens which serves as the input to the transformer model. Given this sequence, the network predicts a categorical distribution for the next token from which we randomly sample the actual token value. During training, the order of objects other than the room, doors and windows is shuffled in the sequence. Furthermore, the attributes of the room can be either mapped to tokens directly (for rectangular rooms only), or by using an additional encoder network given a binary image of the floor plan as input.	69
5.8	Cross-entropy loss and ergonomic loss for our model and its ablations, evaluated on the Bedrooms dataset. As training loss and validation loss we refer to the cross-entropy loss evaluated on the training and validation sets respectively. By including our proposed ergonomic loss term during training we can significantly decrease the ergonomic loss of synthesized layouts. The first 10 epochs are omitted for readability.	71
5.9	Room-conditioned synthesis results for our model and its ablations.	73
5.10	The interface of the user study. Participants were asked which of the 2 displayed scenes is more realistic.	74
		96



5.11 Room-conditioned layout synthesis. We synthesize 20 layout variations for each floor plan in the Bedrooms validation set and evaluate the ergonomic loss. The left chart shows the mean ergonomic loss of the synthesized layouts, with the 80% confidence interval of the mean shown in black. The realism of the synthesized layouts is evaluated in a user study. The right chart shows how the layouts synthesized using each method are perceived compared to the ground truth, with a negative value meaning that the ground truth is seen as more realistic. Our proposed approach improves the ergonomic loss of the scenes, while also being perceived as more realistic than the ground truth.	75
5.12 By pre-training the network on a general dataset containing samples from all room types and then fine-tuning the network for a specific room type, the validation loss can be decreased significantly, especially for small datasets.	76
5.13 Conditional synthesis results as described in Section 5.4. All methods in a column receive the same room boundary, windows, and doors as input condition. Our approach (top row) produces on average layouts with less ergonomic issues like missing light sources (e.g. missing chandeliers in Baseline columns 1, 3 and 6) or poor accessibility (e.g. chair and wardrobe blocking the path in ATISS column 5 and 6).	79
5.14 Layouts created by our models for different room types. Since the attributes of the rooms were represented as part of the input sequences during training, all layout elements including rooms, doors, and windows can be generated by the network. Our method can generate furniture arrangements typical for each room type even with small training sets.	80



# List of Tables

3.1	Results of the questionnaire given to the design students after using our system. Please refer to Section 3.6.2 for the particular questions. . . . .	25
4.1	Quantitative comparison between the results when using a flat patch as input and our method for the poses shown in Figure 4.13 (P1-P6) and the pose shown in Figure 5.1 (P7). We measure the average distance from the body vertices to the closest surface vertex, average and maximum joints moments, as well as average and maximum contact pressure. . . . .	54
5.1	Associations of rules to activities that can be performed in an environment. Not all activities require all rules to be fulfilled. . . . .	63



# List of Algorithms

3.1	Surface Fitting . . . . .	22
-----	---------------------------	----



# Bibliography

- [ACLW<sup>+</sup>19] Esra Ataer-Cansizoglu, Hantian Liu, Tomer Weiss, Archi Mitra, Dhaval Dholakia, Jae-Woo Choi, and Dan Wulin. Room style estimation for style-aware recommendation. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 267–2673. IEEE, 2019.
- [Bas18] Manuel Bastioni. Manuel Bastioni official page, 2018.
- [BKP<sup>+</sup>10] M Botsch, L Kobbelt, M Pauly, P Alliez, and B Levy. *Polygon Mesh Processing*. Ak Peters Series. Taylor & Francis, 2010.
- [BP07] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics*, 26(99):72, jul 2007.
- [BRTT08] Alexandra Brintrup, Jeremy Ramsden, Hideyuki Takagi, and Ashutosh Tiwari. Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms. *Evolutionary Computation, IEEE Transactions on*, 12:343 – 354, 07 2008.
- [Bus04] Samuel Buss. Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Transactions in Robotics and Automation*, 17, 2004.
- [CB12] Francis D. K. Ching and Corky Binggeli. *Interior Design Illustrated*. Wiley, 2012.
- [CC78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, nov 1978.
- [CWH<sup>+</sup>07] K.-S.D. Cheng, Wenping Wang, Hong Qin, K.-Y.K. Wong, Huaiping Yang, and Yang Liu. Design and Analysis of Optimization Methods for Subdivision Surface Fitting. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):878–890, sep 2007.
- [CZW<sup>+</sup>22] Aditya Chattopadhyay, Xi Zhang, David Paul Wipf, Himanshu Arora, and René Vidal. Structured graph variational autoencoders for indoor furniture layout generation. *ArXiv*, abs/2204.04867, 2022.

- [DAdG04] R. Dumas, R. Aissaoui, and J.A. de Guise. A 3D Generic Inverse Dynamic Method using Wrench Notation and Quaternion Algebra. *Computer Methods in Biomechanics and Biomedical Engineering*, 7(3):159–166, jun 2004.
- [DKEV03] Michiel P. De Looze, Lottie F.M. Kuijt-Evers, and Jaap Van Dieën. Sitting comfort and discomfort and the relationships with objective measures. *Ergonomics*, 46(10):985–997, 2003.
- [DSP06] Kevin G. Der, Robert W. Sumner, and Jovan Popović. Inverse kinematics for reduced deformable models. *ACM Transactions on Graphics*, 25(3):1174, jul 2006.
- [DYZ<sup>+</sup>20] Xinhan Di, Pengqian Yu, Hong Zhu, Lei Cai, Qiuyan Sheng, Changyu Sun, and Lingqiang Ran. Structural plan of indoor scenes with personalized preferences. volume 12538 LNCS, 2020.
- [FAK<sup>+</sup>14] R. Fluit, M.S. Andersen, S. Kolk, N. Verdonschot, and H.F.J.M. Koopman. Prediction of ground reaction forces and moments during various activities of daily living. *Journal of Biomechanics*, 47(10):2321–2329, jul 2014.
- [FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [FCG<sup>+</sup>21] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.
- [FCSF17] Qiang Fu, Xiaowu Chen, Xiaoyu Su, and Hongbo Fu. Pose-Inspired Shape Synthesis and Functional Hybrid. *IEEE Transactions on Visualization and Computer Graphics*, 23(12):2574–2585, dec 2017.
- [FCW<sup>+</sup>17] Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.
- [FJG<sup>+</sup>21] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, pages 1–25, 2021.
- [FRS<sup>+</sup>12] Matthew Fisher, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan. Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012.



- [FSL<sup>+</sup>15] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. Activity-centric scene synthesis for functional 3d scene modeling. *ACM Transactions on Graphics (TOG)*, 34(6):1–13, 2015.
- [GGV11] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *CVPR 2011*, pages 1529–1536. IEEE, IEEE, jun 2011.
- [GK97] Etienne Grandjean and Karl H.E. Kroemer. *Fitting The Task To The Human, Fifth Edition: A Textbook Of Occupational Ergonomics*. Taylor & Francis, 1997.
- [GPS02] H Goldstein, C P Poole, and J L Safko. *Classical Mechanics*. Addison Wesley, 2002.
- [Hal95] Susan J Hall. *Basic biomechanics*. McGraw-Hill Humanities, Social Sciences & World Languages, 1995.
- [Her58] Hans T. E. Hertzberg. Seat comfort. *Annotated bibliography of applied physical anthropology in human engineering*,” WADC Technical Report, pages 56–30, 1958.
- [HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [IMT99] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, pages 409–416, 1999.
- [JHR<sup>+</sup>15] Amaury Jung, Stefanie Hahmann, Damien Rohmer, Antoine Begault, Laurence Boissieux, and Marie-Paule Cani. Sketching Folds. *ACM Transactions on Graphics*, 34(5):1–12, 2015.
- [JO16] Alec Jacobson and Others. {gptoolbox}: Geometry Processing Toolbox, 2016.
- [Jon69] J. C. Jones. Methods and results of seating research. *Ergonomics*, 12(2):171–181, 1969.
- [KCGF14] Vladimir G Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. Shape2pose: Human-centric shape analysis. *ACM Transactions on Graphics (TOG)*, 33(4):120, 2014.
- [KL14] Changgu Kang and Sung-Hee Lee. Environment-Adaptive Contact Poses for Virtual Characters. *Computer Graphics Forum*, 33(7):1–10, oct 2014.
- [KLTZ16] Z Sadeghipour Kermani, Zicheng Liao, Ping Tan, and H Zhang. Learning 3d scene synthesis from annotated rgb-d images. In *Computer Graphics Forum*, volume 35, pages 197–206. Wiley Online Library, 2016.

- [Kro17] Karl H.E. Kroemer. *Fitting the Human: Introduction to Ergonomics / Human Factors Engineering, Seventh Edition*. CRC Press, 2017.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.
- [KTOK82] Ken Kamijo, Harutoshi Tsujimura, Hideo Obara, and Masaaki Katsumata. Evaluation of seating comfort. *SAE transactions*, pages 2615–2620, 1982.
- [Kuo98] A. D. Kuo. A Least-Squares Estimation Approach to Improving the Precision of Inverse Dynamics Computations. *Journal of Biomechanical Engineering*, 120(1):148, 1998.
- [Law18] Neil D Lawrence. mocap: Matlab Motion Capture Toolbox, 2018.
- [LBRM18] Kurt Leimer, Michael Birsak, Florian Rist, and Przemyslaw Musialski. Sit & relax: Interactive design of body-supporting surfaces. In *Computer Graphics Forum*, volume 37, pages 349–359. Wiley Online Library, 2018.
- [LCMS16] Bokyung Lee, Minjoo Cho, Joonhee Min, and Daniel Saakes. Posing and acting as input for personalizing furniture. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, page 44. ACM, 2016.
- [LGWM17] Kurt Leimer, Lukas Gersthofer, Michael Wimmer, and Przemyslaw Musialski. Relation-based parametrization and exploration of shape collections. *Computers & Graphics*, 67:127–137, 2017.
- [LGWM22] Kurt Leimer, Paul Guerrero, Tomer Weiss, and Przemyslaw Musialski. Layoutenhancer: Generating good indoor layouts from imperfect data. 2022.
- [LKS15] Zhaoliang Lun, Evangelos Kalogerakis, and Alla Sheffer. Elements of style: learning perceptual shape style similarity. *ACM Transactions on graphics (TOG)*, 34(4):1–14, 2015.
- [LLL<sup>+</sup>19] Wei Liang, Jingjing Liu, Yining Lang, Bing Ning, and Lap-Fai Yu. Functional workspace optimization via learning personal preferences from virtual experiences. *IEEE transactions on visualization and computer graphics*, 25(5):1836–1845, 2019.
- [LM20] Kurt Leimer and Przemyslaw Musialski. Reduced-order simulation of flexible meta-materials. In *Symposium on Computational Fabrication, SCF '20*, New York, NY, USA, 2020. Association for Computing Machinery.

- [LM22] Kurt Leimer and Przemyslaw Musialski. Analysis of a reduced-order model for the simulation of elastic geometric zigzag-spring meta-materials. *Computers & Graphics*, 102:187–198, 2022.
- [LPL<sup>+</sup>17] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. BendSketch: Modeling Freeform Surfaces Through 2D Sketching. *ACM Transactions on Graphics*, 36(4):1–14, 2017.
- [LPX<sup>+</sup>19] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)*, 38(2):1–16, 2019.
- [LTPSR08] Miguel López-Torres, Rosa Porcar, José Solaz, and Tomás Romero. Objective firmness, average pressure and subjective perception in mattresses for the elderly. *Applied Ergonomics*, 39(1):123 – 130, 2008.
- [Lue83] Rani Karen Lueder. Seat Comfort: A Review of the Construct in the Office Environment. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 25(6):701–711, dec 1983.
- [LWOM20] Kurt Leimer, Andreas Winkler, Stefan Ohrhallinger, and Przemyslaw Musialski. Pose to seat: Automated design of body-supporting surfaces. *Computer Aided Geometric Design*, 79:101855, 2020.
- [LXW<sup>+</sup>11] Yang Liu, Weiwei Xu, Jun Wang, Lifeng Zhu, Baining Guo, Falai Chen, and Guoping Wang. General planar quadrilateral mesh design using conjugate direction field. *ACM Trans. Graph.*, 30(6):140:1–140:10, December 2011.
- [LZWT20] Andrew Luo, Zhoutong Zhang, Jiajun Wu, and Joshua B. Tenenbaum. End-to-end optimization of scene layout. 2020.
- [MK05] Martin Marinov and Leif Kobbelt. Optimization methods for scattered data approximation with subdivision surfaces. *Graphical Models*, 67(5):452–473, sep 2005.
- [MSL<sup>+</sup>11] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. *ACM transactions on graphics (TOG)*, 30(4):1–10, 2011.
- [MTLT89] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proceedings on Graphics interface '88*, pages 26–33, dec 1989.
- [Neu18] Perception Neuron. Perception Neuron by Noitom | Perception Neuron motion capture for virtual reality, animation, sports, gaming and film, 2018.

- [NISA07] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Fibermesh: designing freeform surfaces with 3d curves. *ACM Transactions on Graphics*, 26:41, 2007.
- [NNL<sup>+</sup>12] Kageyu Noro, Tetsuya Naruse, Rani Lueder, Nobuhisa Nao-i, and Maki Kozawa. Application of zen sitting principles to microscopic surgery seating. *Applied Ergonomics*, 43(2):308 – 319, 2012. Special Section on Product Comfort.
- [PEA83] Stanley Plagenhoef, F Gaynor Evans, and Thomas Abdelnour. Anatomical Data for Analyzing Human Motion. *Research Quarterly for Exercise and Sport*, 54(2):169–178, jun 1983.
- [Pet00] Jörg Peters. Patching Catmull-Clark meshes. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIG-GRAPH '00*, pages 255–258, New York, New York, USA, 2000. ACM Press.
- [PGK<sup>+</sup>21] Wamiq Para, Paul Guerrero, Tom Kelly, Leonidas J Guibas, and Peter Wonka. Generative layout modeling using constraint graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6690–6700, 2021.
- [PH10] Jenny Pynt and Joy Higgs. *A History of Seating, 3000 BC to 2000 AD: Function Versus Aesthetics*. Cambria Press, 2010.
- [Pil05] John F Pile. *A history of interior design*. Laurence King Publishing, 2005.
- [PKS<sup>+</sup>21] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [PLBM20] Stefan Pillwein, Kurt Leimer, Michael Birsak, and Przemyslaw Musialski. On elastic geodesic grids and their planar to spatial deployment. *ACM Trans. Graph.*, 39(4), jul 2020.
- [Pop10] Valentin L. Popov. Coulomb’s Law of Friction. In *Contact Mechanics and Friction*, pages 133–154. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [PRB<sup>+</sup>18] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba. Virtualhome: Simulating household activities via programs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8494–8502, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society.
- [QZH<sup>+</sup>18] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5899–5908, 2018.

- [RCH<sup>+</sup>04] Gordon Robertson, Graham Caldwell, Joseph Hamill, Gary Kamen, and Saunders Whittlesey. *Research methods in biomechanics*. 2004.
- [RP08] Matthew P Reed and Matthew B Parkinson. Modeling variability in torso shape for chair and seat design. In *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 561–569. American Society of Mechanical Engineers, 2008.
- [RWC<sup>+</sup>19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [SBK<sup>+</sup>16] Maxim Smulders, Karlien Berghman, M Koenraads, J.A. Kane, Kashyap Krishna, Terence Carter, and Udo Schultheis. Comfort and pressure distribution in a human contour shaped aircraft seat (developed with 3d scans of the human body). *Work*, 54:1–16, 08 2016.
- [SLMI11] Greg Saul, Manfred Lau, Jun Mitani, and Takeo Igarashi. SketchChair: an all-in-one chair design system for end users. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction - TEI '11*, page 73, New York, New York, USA, jan 2011. ACM Press.
- [Smi09] Michael Smith. *ABAQUS/Standard User's Manual, Version 6.9*. Simulia, 2009.
- [Sta95] H a M Staarink. Sitting posture, comfort and pressure: assessing the quality of wheelchair chushions. *Journal of Rehabilitation Sciences*, 1995.
- [SZGP05] Robert W. Sumner, Matthias Zwicker, Craig Gotsman, and Jovan Popović. Mesh-based inverse kinematics. *ACM Transactions on Graphics*, 24(3):488, jul 2005.
- [Tur22] Turbosquid. 3D Model Collection. <https://www.turbosquid.com/>, 2022. [Online; accessed Jan-2022].
- [UIM12] Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra. Guided exploration of physically valid shapes for furniture design. *ACM Transactions on Graphics*, 31(4):1–11, jul 2012.
- [Vli14] Nathaniel Vlietstra. *Comparing methods for full body inverse dynamics analysis of a standing long jump*. PhD thesis, Grand Valley State University, 2014.

- [VPGV20] Nick Vitsas, Georgios Papaioannou, Anastasios Gkaravelis, and Andreas-Alexandros Vasilakis. Illumination-guided furniture layout optimization. In *Computer Graphics Forum*, volume 39, pages 291–301. Wiley Online Library, 2020.
- [VSP<sup>+</sup>17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [Way22] Wayfair. Room Planner. <https://www.wayfair.com/RoomPlanner3D>, 2022. [Online; accessed Jan-2022].
- [WDS<sup>+</sup>20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- [WFT<sup>+</sup>19] Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019.
- [Win19] Andreas Winkler. Pose-driven generation and optimization of seating furniture. Master’s thesis, Research Unit of Computer Graphics, Institute of Visual Computing and Human-Centered Technology, Faculty of Informatics, TU Wien, Favoritenstrasse 9-11/E193-02, A-1040 Vienna, Austria, October 2019.
- [WLD<sup>+</sup>18] Tomer Weiss, Alan Litteneker, Noah Duncan, Masaki Nakada, Chenfanfu Jiang, Lap-Fai Yu, and Demetri Terzopoulos. Fast and scalable position-based layout synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 25(12):3231–3243, 2018.
- [WLW<sup>+</sup>19] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.
- [WSCR18] Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

- [WWT<sup>+</sup>18] Yu-Chian Wu, Te-Yen Wu, Paul Taelle, Bryan Wang, Jun-You Liu, Pin-sung Ku, Po-En Lai, and Mike Y. Chen. Activeergo: Automatic and personalized ergonomics using self-actuating furniture. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 558:1–558:8, New York, NY, USA, 2018. ACM.
- [WYA<sup>+</sup>20] Tomer Weiss, Ilkay Yildiz, Nitin Agarwal, Esra Ataer-Cansizoglu, and Jae-Woo Choi. Image-driven furniture style for interactive 3d scene modeling. In *Computer Graphics Forum*, volume 39, pages 57–68. Wiley Online Library, 2020.
- [WYN20] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. *arXiv preprint arXiv:2012.09793*, 2020.
- [YDF92] Myung Hwan Yun, Lynn Donges, and Andris Freivalds. Using force sensitive resistors to evaluate the driver seating comfort. *Advances in Industrial Ergonomics and safety IV*, pages 403–410, 1992.
- [YYT<sup>+</sup>11] Lap Fai Yu, Sai Kit Yeung, Chi Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011*, v. 30,(4), July 2011, article no. 86, 30(4), 2011.
- [YZY<sup>+</sup>21] Haitao Yang, Zaiwei Zhang, Siming Yan, Haibin Huang, Chongyang Ma, Yi Zheng, Chandrajit Bajaj, and Qixing Huang. Scene synthesis via uncertainty-driven attribute synchronization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5630–5640, 2021.
- [ZFBV12] R. Zenk, M. Franz, H. Bubb, and P. Vink. Technical note: Spine loading in automotive seating. *Applied Ergonomics*, 43(2):290 – 295, 2012. Special Section on Product Comfort.
- [ZHD96] Luian Zhang, Martin G. Helander, and Colin G. Drury. Identifying Factors of Comfort and Discomfort in Sitting. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 38(3):377–389, sep 1996.
- [ZHPY21] Yongqi Zhang, Haikun Huang, Erion Plaku, and Lap-Fai Yu. Joint computational design of workspaces and workplans. *ACM Transactions on Graphics (TOG)*, 40(6):1–16, 2021.
- [ZLDM16] Youyi Zheng, Han Liu, Julie Dorsey, and Niloy J. Mitra. Ergonomics-Inspired Reshaping and Exploration of Collections of Models. *IEEE Transactions on Visualization and Computer Graphics*, 22(6):1732–1744, jun 2016.

- [ZSW10] Mirko Zadavec, Alexander Schiftner, and Johannes Wallner. Designing quad-dominant meshes with planar faces. *Computer Graphics Forum*, 29(5):1671–1679, 2010.
- [ZWK19] Yang Zhou, Zachary While, and Evangelos Kalogerakis. Scenegrphnet: Neural message passing for 3d indoor scene augmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7384–7392, 2019.
- [ZYM<sup>+</sup>20] Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. Deep generative modeling for scene synthesis via hybrid representations. *ACM Transactions on Graphics (TOG)*, 39(2):1–21, 2020.



# CV

## **KURT LEIMER**

Phone: +43 6766394880

Mail: kurtleimer@hotmail.com

## **EDUCATION**

Ongoing PhD studies (expected graduation 2023), Computer Science, Vienna University of Technology, Austria

Supervisors: Przemyslaw Musialski, Michael Wimmer

Dipl.-Ing., Visual Computing, Vienna University of Technology, Austria, 2016

Master Thesis: Co-Analysis and Parameterization of 3D Shape Collections for Shape Synthesis

Supervisors: Michael Wimmer, Przemyslaw Musialski

BSc, Media Informatics and Visual Computing, Vienna University of Technology, Austria, 2013

Bachelor Thesis: External Sorting of Point Clouds

Supervisors: Michael Wimmer, Claus Scheiblauer

## **EMPLOYMENT**

Data Scientist, Andritz AG, Vienna, Austria, 2022-present

UG Student Research Assistant, New Jersey Institute of Technology, Newark, NJ, USA, 2021-2022

Research Assistant, Vienna University of Technology, Austria, 2016-2021

## **TEACHING**

Master Thesis Co-Supervisor, Vienna University of Technology, Austria, 2018-2019

Thesis: Andreas Winkler, Pose-Driven Generation and Optimization of Seating Furniture

Lab Exercise Supervisor, Vienna University of Technology, Austria, 2018-2019

Courses: Geometry for Computer Science, Modeling in Computer Graphics

## PUBLICATIONS

Kurt Leimer, Paul Guerrero, Tomer Weiss, Przemyslaw Musialski  
*LayoutEnhancer: Generating Good Indoor Layouts from Imperfect Data*  
ACM SIGGRAPH Asia 2022 Conference Papers (SA '22 Conference Papers), 2022

Kurt Leimer and Przemyslaw Musialski  
*Analysis of a Reduced-Order Model for the Simulation of Elastic Geometric Zigzag-Spring Meta-Materials*  
Computers & Graphics, Volume 102, Pages 187-198, 2022

Kurt Leimer and Przemyslaw Musialski  
*Reduced-Order Simulation of Flexible Meta-Materials*  
SCF '20: Symposium on Computational Fabrication, 2020

Stefan Pillwein, Kurt Leimer, Michael Birsak, Przemyslaw Musialski  
*On Elastic Geodesic Grids and Their Planar to Spatial Deployment*  
ACM Trans. Graph., 39, 4, 2020

Kurt Leimer, Andreas Winkler, Stefan Ohrhallinger, Przemyslaw Musialski  
*Pose to Seat: Automated Design of Body-Supporting Surfaces*  
Computer Aided Geometric Design, 79, 2020

Kurt Leimer, Michael Birsak, Florian Rist, Przemyslaw Musialski  
*Sit & Relax: Interactive Design of Body-Supporting Surfaces*  
Computer Graphics Forum (Proc. PG2018), 37(7), October 2018

Kurt Leimer, Lukas Gersthofer, Michael Wimmer, Przemyslaw Musialski  
*Relation-Based Parametrization and Exploration of Shape Collections*  
Computers & Graphics, 67:127-137, October 2017

## PRESENTATIONS

SIGGRAPH Asia 2022, Daegu, South Korea  
Paper Presentation  
*LayoutEnhancer: Generating Good Indoor Layouts from Imperfect Data*

SCF '20: Symposium on Computational Fabrication, Online Conference  
Paper Presentation  
*Reduced-Order Simulation of Flexible Meta-Materials*

SIGGRAPH 2020, Online Conference  
Poster Session  
*Simulation of Flexible Patterns by Structural Simplification*

GCD 5 - Symposium on Geometry and Computational Design 2018, Vienna, Austria  
Paper Presentation  
*Sit & Relax: Interactive Design of Body-Supporting Surfaces*

Pacific Graphics 2018, Hong Kong  
Paper Presentation  
*Sit & Relax: Interactive Design of Body-Supporting Surfaces*

SIGGRAPH 2016, Anaheim, CA  
Poster Session  
*Relation-Based Parametrization and Exploration of Shape Collections*

## **AWARDS AND HONORS**

Best Paper Honorable Mention - Pacific Graphics 2018, Hong Kong  
Kurt Leimer, Michael Birsak, Florian Rist, Przemyslaw Musialski  
*Sit & Relax: Interactive Design of Body-Supporting Surfaces*

## **OTHER PUBLICATIONS (POSTERS, ETC.)**

Kurt Leimer and Przemyslaw Musialski  
Simulation of Flexible Patterns by Structural Simplification (Poster)  
In: ACM SIGGRAPH 2020 Posters

Kurt Leimer, Michael Wimmer, Przemyslaw Musialski  
Relation-Based Parametrization and Exploration of Shape Collections (Poster)  
In: SIGGRAPH '16: ACM SIGGRAPH 2016 Posters. 2016