

# Pose-Driven Generation and Optimization of Seating Furniture

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Visual Computing**

eingereicht von

**BSc. Andreas Winkler**

Matrikelnummer 01129264

an der Fakultät für Informatik  
der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dipl.-Ing. Dr.techn. Michael Wimmer  
Mitwirkung: Przemyslaw Musialski, MSc. Dr.techn.  
Kurt Leimer, MSc.

Wien, 21. Oktober 2019

---

Andreas Winkler

---

Michael Wimmer



# Pose-Driven Generation and Optimization of Seating Furniture

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Visual Computing**

by

**BSc. Andreas Winkler**

Registration Number 01129264

to the Faculty of Informatics

at the TU Wien

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

Assistance: Przemyslaw Musialski, MSc. Dr.techn.  
Kurt Leimer, MSc.

Vienna, 21<sup>st</sup> October, 2019

---

Andreas Winkler

---

Michael Wimmer



# Erklärung zur Verfassung der Arbeit

BSc. Andreas Winkler  
Address

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, 21. Oktober 2019

---

Andreas Winkler



# Acknowledgements

First and foremost, I want to thank my family and friends for their love and support during my entire studies. It has been a long journey that I couldn't have possibly completed without them.

I would like to thank my supervisors Kurt Leimer and Przemyslaw Musialski for their guidance, assistance and expertise during my work on this thesis. Likewise, I'm thankful for all the other researchers around the world, whose insights were essential for the completion of this work.

Last, but not least, my thanks go out to all the great people I had the pleasure of working with during my time at the TU Wien. I want to thank the professors and university staff for their teachings and assistance, as well as all my former colleagues for their support.



# Kurzfassung

Moderne Designsysteme sind in der Lage, Entwürfe für Sitzmöbel für verschiedenste Anwendungen zu erstellen, sowohl für allgemeine Zwecke als auch für spezielle Umgebungen. Das primäre Ziel dieser Systeme ist es, möglichst komfortable Sitzgelegenheiten zu erschaffen. Um optimalen Komfort für bestimmte Personen in besonderen Umgebungen zu gewährleisten, ist ein personalisierter Designprozess erforderlich. Komfort wird im Allgemeinen als ein subjektives Empfinden einer Person definiert. Objektive Metriken werden definiert um das Empfinden einer Person möglichst genau für eine bestimmte Sitzfläche zu approximieren. Computerunterstützte Designsysteme erschaffen spezialisierte Sitzflächen mit Hilfe interaktiver Algorithmen, bei denen professionelle Designer oder der Endbenutzer selbst in den Designvorgang einbezogen werden.

Im Rahmen dieser Diplomarbeit wird eine computerunterstützte Design Umgebung präsentiert, welche personalisierte Sitzflächen generiert. Durch Approximation des Komforts einer Person, auf Basis der Verteilung des Oberflächendrucks, wird eine Sitzfläche an eine Person in einer bestimmten Sitzpose angepasst. Das vollständig automatisierte System ist in der Lage komfortable Sitzgelegenheiten für variable Körperformen und Sitzposen zu erzeugen. Ein generisches Modell wird in einem mehrstufigen Prozess an beliebige Körperformen und Sitzposen angepasst. In einem nichtlinearen Optimierungsprozess wird das Oberflächenmodell weiter verbessert um sowohl optimalen Komfort als auch visuelle Qualität zu gewährleisten.

In weiterer Folge dient das präsentierte System als automatisierte Lösung um Eingabedaten für weitere Design und Optimierungssysteme zu erzeugen und damit manuellen Arbeitsaufwand zu reduzieren.



# Abstract

Modern furniture design systems provide seating solutions for various applications, ranging from general purpose solutions to specific environments. The central goal of furniture design is to create comfortable seating surfaces. To provide optimal comfort for a specific person and environment, personalized furniture design is required. As comfort is generally seen as the user's subjective feeling, objective comfort measures are defined that approximate a person's comfort for a given seating surface. Computational furniture design systems create seating solutions for a given scenario using interactive algorithms. Specialized seating surfaces often require extensive manual design effort.

In this thesis, a computational furniture design framework to generate personalized seating surfaces is proposed. Utilizing a notation of sitting comfort based on equal pressure distribution, our algorithm generates seating surface models fitted to a person in a specific pose. We introduce an automated furniture design framework able to create comfortable seating surfaces for specific body shapes and poses. We developed a generic template model capable of supporting a large variety of sitting poses and human body shapes that is matched to an input pose in multi stage fitting process. Furthermore, we introduce a non-linear mesh optimization algorithm for further functional and visual improvements.

In addition, the proposed framework serves as a fully automated solution to create specialized control meshes usable as input meshes in other design frameworks, thus eliminating the need for manual design effort.



# Contents

|   |             |
|---|-------------|
| <b>Kurzfassung</b>  | <b>ix</b>   |
| <b>Abstract</b>   | <b>xi</b>   |
| <b>Contents</b>   | <b>xiii</b> |
| <b>1 Introduction</b>                                       | <b>1</b>    |
| 1.1 Motivation . . . . .                                    | 1           |
| 1.2 Aim of the work . . . . .                               | 2           |
| 1.3 Structure of the work . . . . .                         | 3           |
| <b>2 Background</b>   | <b>5</b>    |
| 2.1 Furniture design . . . . .                              | 5           |
| 2.2 Comfort . . . . .                                       | 11          |
| 2.3 Related work in personalized furniture design . . . . . | 15          |
| 2.4 Optimization . . . . .                                  | 21          |
| <b>3 Pose-Driven Design</b>                                 | <b>27</b>   |
| 3.1 Goals and motivation . . . . .                          | 27          |
| 3.2 Overview . . . . .                                      | 29          |
| 3.3 Human body representation . . . . .                     | 30          |
| 3.4 Sitting poses . . . . .                                 | 32          |
| 3.5 Comfort measures . . . . .                              | 34          |
| <b>4 Template Model</b>                                     | <b>37</b>   |
| 4.1 Requirements . . . . .                                  | 37          |
| 4.2 Initial template model . . . . .                        | 38          |
| 4.3 Advanced template model . . . . .                       | 50          |
| <b>5 Optimization</b>                                       | <b>65</b>   |
| 5.1 Goals . . . . .   | 65          |
| 5.2 Optimization problem overview . . . . .                 | 66          |
| 5.3 Variables . . . . .                                     | 66          |
| 5.4 Quality measures . . . . .                              | 67          |
|   | xiii        |

|          |                                       |            |
|----------|---------------------------------------|------------|
| 5.5      | Constraints . . . . .                 | 69         |
| 5.6      | Energy function . . . . .             | 70         |
| 5.7      | Gradient . . . . .                    | 72         |
| 5.8      | Optimization results . . . . .        | 75         |
| <b>6</b> | <b>Implementation</b>                 | <b>77</b>  |
| 6.1      | Overview . . . . .                    | 77         |
| 6.2      | MATLAB . . . . .                      | 78         |
| 6.3      | Rhino 3D . . . . .                    | 87         |
| <b>7</b> | <b>Results</b>                        | <b>89</b>  |
| 7.1      | Overview . . . . .                    | 89         |
| 7.2      | Visual results . . . . .              | 90         |
| 7.3      | Evaluation . . . . .                  | 99         |
| <b>8</b> | <b>Conclusion</b>                     | <b>105</b> |
| 8.1      | Contributions . . . . .               | 105        |
| 8.2      | Limitations and future work . . . . . | 106        |
|          | <b>List of Figures</b>                | <b>109</b> |
|          | <b>List of Tables</b>                 | <b>111</b> |
|          | <b>List of Algorithms</b>             | <b>113</b> |
|          | <b>Bibliography</b>                   | <b>115</b> |

# Introduction

In the average 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.

## 1.1 Motivation

To provide optimal sitting comfort for specific applications, personalized furniture is required. The goal of personalized furniture design is to create the right piece of furniture for a specific person in a specific situation or environment. The central task is therefore to find an optimal fit between a person and the seating surface.

While personalized furniture design is not a novel concept, the computational design of seating furniture is a more recent topic. Modern furniture design systems aim to create or optimize specific pieces of furniture, for a specific person and environment. For optimal sitting 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.

Personalized furniture design applications are commonly limited to specific situations and environments. Providing seating surfaces for arbitrary seating poses is a difficult task

and often requires manual design effort to guide the computational design process. Our primary motivation for this thesis is to provide a personalized furniture design solution capable of supporting a person in a general sitting pose, while eliminating most manual design effort.

### 1.2 Aim of the work

The central goal of this work is to provide a solution for computational personalized 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 personalized seating surfaces, 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:

1. The created models must provide a high level of sitting comfort. The initial focus lies on a specialized environment. The primary objective is therefore to create a seating surface which optimally supports a given body shape for a specific pose.
2. 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.
3. Unique aesthetics and appealing shapes are generally desired in furniture and product design. We aim to achieve a high level of visual quality and create furniture models that are visually pleasing.

In order to design a framework that fulfills these goals we face a number of challenges, that we aim to overcome with the following contributions:

- We propose a **pose-driven design approach**, utilizing a comfort model based on pressure distribution to approximate sitting comfort for a person in a given pose. A suitable representation of a human body shape and pose is required in order to create personalized seating surfaces.
- A generic furniture model needs to be defined which is capable of supporting a large variety of poses and body shapes, while fulfilling the functional and visual requirements. We propose a **template seating surface model** that is capable of supporting individual parts of a person's body. A RANSAC based surface fitting algorithm is developed to ensure an optimal fit between the surface model and the body shape.

- Appealing aesthetics are desired for the resulting furniture models. Therefore a **mesh optimization algorithm** is proposed that is capable of improving the model's visual quality while preserving its functionality as supporting surface.

## 1.3 Structure of the work

The remaining chapters of this thesis are structured as follows:

- In Chapter 2, detailed insights into the background and concepts relevant to this thesis are provided, including furniture design, sitting comfort and optimization techniques. Related work in the field of personalized furniture design is presented.
- Next, the methodological approach for this thesis is presented in detail. Chapter 3 introduces the pose-driven furniture design approach that is followed for this work. This includes our chosen notation of comfort as well as the representations for human body shapes and sitting poses.
- In Chapter 4, we present the algorithms and techniques used to create the template model. This chapter revolves around surface fitting and mesh generation.
- In Chapter 5, a mesh optimization approach used to refine the visual and functional quality of the model is proposed. The optimization problem, including its constraints and objective function are discussed.
- Chapter 6 gives a detailed view of the system architecture of the software framework. The implementation of the core components of the system is explained in detail.
- Visual examples of seating surfaces generated by the proposed framework are presented in Chapter 7. We show a variety of 3D seating furniture results generated for different poses and body shapes.
- Chapter 8 concludes this thesis with a summary and critical reflection of our contributions and its limitations, as well as a number of proposals for future work.



# Background

In this chapter, the fundamental concepts relevant for this thesis are presented. We describe the general goals and concepts of furniture design in detail. In addition, we delve into the field of sitting comfort and its history of research. Furthermore, we evaluate the state of the art in the field of computational furniture design and present selected related research in detail. As last part of this chapter, we provide a short introduction to optimization techniques.

## 2.1 Furniture design

From a general perspective, the central goal of this work is the design of 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. In this work the focus lies on computerized product design of comfortable seating furniture. Depending on the application, a number of goals and design principles are considered. In this section, various decisions and goals that arise in the process of digital furniture design are discussed.

Throughout this chapter, the terms *optimal* or *fitting* are frequently used in context of seating furniture. As there is no universal definition on optimal or fitting furniture, most researchers have their own understanding of these terms. These concepts are explained in detail in Sections 2.1.5 and 2.2.

### 2.1.1 General purpose or specialized solutions

An important question is whether a designed seating surface is aimed for a general application or to be used in a specific situation only. For one, this includes the physical dimensions and shape of the seating surface. For instance, a seating surface to be used in vehicles will be subject to different limitations and constraints than a dentist's chair.

Designing a generic seating surface usable in both applications will most likely only be possible as a tradeoff with a loss of optimality for both applications.

With equal importance, specialization also applies to the designated users of seating furniture. The physical shape and size can vary greatly between human persons. As a result, it is a virtually impossible task to design a seating surface that optimally fits a wide range of human body shapes. For instance a seat tailored to a tall overweight person might provide only limited support and comfort to a child.

While specialized solutions seem more appropriate in most situations, it is clear that this is unfeasible in a lot of applications. In any public place or transport, 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. However, optimally supporting even the most extreme cases of human body shapes with a general seating solution is not possible with current methods.

In contrast, some applications allow the use of specialized furniture. Designing furniture for a specific purpose or for a specific type of body shapes, allows for much more refined and precise solutions. While designing general purpose furniture is an important task, the focus on this work lies on the design of personalized seating furniture.

### 2.1.2 Object centered design

Following a specialized design approach, further decisions in the design process are required: What is the primary target of the design? Object centered furniture design is referring to applications where the seating surface is already given or restricted by tight constraints. The general task therefore revolves not necessarily around designing the seating surface, but rather around understanding its optimal use. Research in this area has focused on recognizing seating surfaces from images [GGVG11] and finding human poses to optimally use them [KCGF14], [KL14]. Acquiring information about the semantics of human-object interaction can provide important insights for furniture design or assist in evaluation of existing furniture.

### 2.1.3 Human centered design

On the contrary, in a human centered design process the attention is shifted to the needs and requirements of a person. The goal is to find a seating surface that optimally matches the requirements of a person, ranging from physical properties such as shape or size [RP08] to semantic constraints. In this context we can distinguish research based on how human requirements are included in the design process.

What we will refer to as *function driven* design, is research that focuses their attention on functional requirements for human interaction with the seating surface. These requirements refer to semantics of human-object interaction or describe general activities such as sitting at a desk or lying in a bed. Research in function driven design aims to find

seating surfaces which match a general class of poses or guidelines [ZLDM16, BRTT08].

In *pose-driven* design the goal is to fulfill much stricter human requirements. Research in this area considers a given pose as optimal for a specific situation and aims to design a seating surface to match a person in that pose as close as possible [FCSF17], [LBRM18]. As the physical dimensions of a human body in a specific pose can differ between persons, pose-driven design also has to take body shapes into account. Pose-driven design is one of the core concepts of this thesis. Related work in this area will be presented in detail in Section 2.3

Human- and object centered methods are often combined to provide better results. For instance, a seating surface that was matched to a human body could be further improved by adjusting the person's pose for an even more refined match to the surface [LBRM18].

#### 2.1.4 Interactive design

Interactivity is an important factor in personalized furniture design. Most modern furniture design systems are created to provide some degree of user interaction. The level of interactivity varies between systems, depending on their application and the intended user group.

For example, Saul et al. [SLMI11] proposed a furniture design system intended for end users which allows them to design chairs from free-form shapes. Interactivity is a core element in their research. Lee et al. [LCMS16] designed their system around VR technology to allow users to personalize furniture via poses and voice commands.

On the contrary, some researchers focused on automated systems that aim at the design of well-fitting furniture in an automated process. User interaction is mostly limited to customizing input data and parameters [ZLDM16, VDVD<sup>+</sup>11].

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].

One of the main benefits of interactive design is the possibility to incorporate professional designers and/or the actual end user in the process. This would shift the design process closer to regular product design and provide a better control over the results, and therefore produce more professional designs or results closer to a specific user's needs. As an obvious downside, an interactive design process requires additional manual effort, which can be time consuming and result in additional costs.

Automated systems will allow for a faster design process by limiting or entirely foregoing manual design input and therefore reducing the required effort with the obvious tradeoff of limited feedback from professional designers or end users. The quality of the results of automated systems therefore greatly relies on the quality of input data and the accuracy of the algorithm itself.

### 2.1.5 Design goals

The central goal of personalized furniture design is to create an optimal piece of furniture for a specific situation. While the exact definition of an optimum clearly varies between different applications, most research on furniture design share a number of common design goals:

- **Physical validity:**

While most research in furniture design is focused on the shape of the seating surfaces, some researchers have also included physical validity as a goal in their design process. This ranges from simple physics simulations to evaluate balance and stability [SLMI11] to more complex evaluation of structure and materials [HMBB17]. Umetani et al. [UIM12] provide a furniture design system which evaluates physical validity during an interactive design process.

- **Aesthetics:**

Traditionally, one of the most important goals of furniture designs is the creation of visually pleasing results. While the well known *form follows function* principle states that the function of an object dictates its form, unique aesthetics are often desired in product design. A common goal is therefore to create a fully functional seating surface that is also visually appealing.

- **Comfort:**

The indisputably most important goal in seating furniture design is to create a comfortable seating surface. While this is a general goal, no universally agreed definition of comfort exists. There is a long history of research in this field with a small number of common definitions which modern furniture design is based on for the most part. Section 2.2 provides a more detailed description of research on comfort.

### 2.1.6 Fabrication

While digital assets, such as 3D models of furniture are in demand for a number of applications, in the most common scenario, the desired end product of personalized furniture design is a physical piece of seating furniture. Therefore, a lot of researchers in furniture design also aim to provide a solution for the fabrication of their created furniture. Some researchers shifted their focus entirely on the fabrication aspect. Depending on the application or the designated users, the exact requirements on fabrication differ through various research. In general, efficient material usage and cost efficiency are highly sought after.

Fan and Schodek [FS07] researched the use of shape memory polymer as material

in furniture design. *Transformable Furniture* was introduced, made of carbon fiber reinforced SMP composite panels, providing consumers a new way of customizing furniture. Shaped memory polymer (SMP) stays rigid at a cooled state and can be reshaped by a cycle of heating, deforming and cooling. Prototype furniture consisting of multiple connected panels was introduced allowing users to assemble and shape a custom piece of seating furniture according to their personal preference.

For personalized furniture design that is aimed at the end user, fast and easy assembly is commonly a desired goal. The SketchChair [SLMI11] system produces furniture designs that can be fabricated from different flat materials using a variety of production equipment such as CNC mills, hand saws or laser cutters, as well as simple prototypes from paper or cardboard at a smaller scale (Figure 2.2). The furniture designs consisting of flat pieces are designed for quick and easy assembly by the end user and also have the advantage of cost-effective shipping. Optionally, the assembled chair frames can be reinforced by aluminum corner braces to improve stability and also covered by a softer seating surface to improve sitting comfort.

Lau et al. [LOMI11] introduce an automatic system to convert digital furniture models into a fabricatable parts (Figure 2.1). From arbitrary 3D furniture models, the system is able to identify the primitive parts and structure of the model to generate a list of parts and connectors (i.e., screws, nails) required to assemble the corresponding physical furniture piece. While this research is focused on the fabrication of cabinets and tables rather than seating furniture, the general transition from digital models to fabricatable parts is an interesting concept in furniture design.

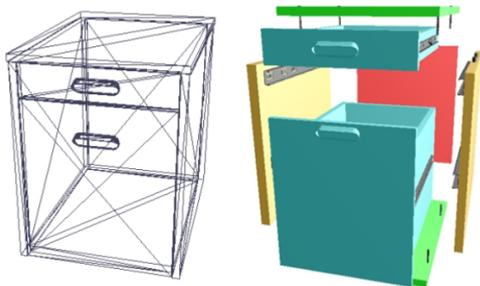


Figure 2.1: 3D furniture model converted into individual parts using the algorithm by Lau et al. [LOMI11]



Figure 2.2: SketchChair [SLMI11] promotional image showing individual parts and an assembled chair.

Schwartzburg and Pauly [SP13] developed a computational design system to generate 3D models consisting of interlocking planar pieces (Figure 2.3). The computed pieces

can be fabricated from flat panels and easily assembled by sliding the pieces into each other along straight slits without the need of glue, screws or other means of support. The researchers provide an interactive system to design and optimize furniture pieces in a way that ensures meeting the requirements for fabrication, stability and assembly.

In a different approach by Haeusler et al. [HMBB17], the use of 3D-printing to produce structural elements in furniture is evaluated. The researchers introduced a custom optimization method for highly complex structural nodes at furniture scale and successfully produced a full scale prototype from 3D printed plastic nodes and timber members (Figure 2.4).



Figure 2.3: Chair built from interlocking parts created by the system from Schwartzburg and Pauly [SP13].

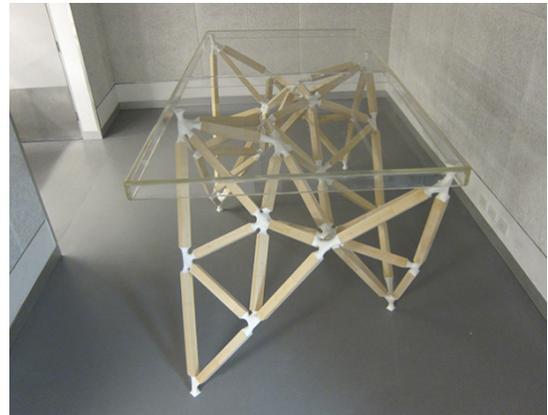


Figure 2.4: Table assembled from 3D printed structural nodes and timber elements (Haeusler et al. [HMBB17]).

Grujovic et al. [GZZ<sup>+</sup>17] evaluate the benefits and possibilities of 3D-printing techniques in the wood industry. Various techniques and materials for use in furniture design are reviewed, considering mechanical strength, material durability, cost and customization. The researchers have successfully created custom wood clamp tools which can be used to hold furniture parts together to assist the assembly of custom furniture objects allowing complex designs to be easily assembled. Experiments proved 3D-printed connectors (fixtures, fittings, joiners) were suitable as structural elements in lightweight furniture.

Furniture fabrication implicitly introduces the challenges of efficient material usage. Koo et al. [KHL17] introduce the concept of *waste minimizing furniture design*. The researchers propose a system which dynamically analyzes material space layout during the design process to guide a user to adjust and improve the design towards a more efficient material usage. During this process, the original design intent is maintained via the use

of design constraints. While this tight coupling between 2D material usage and 3D design can provide wastage reduction by 10% to 15%, the researchers acknowledge that their current system does not consider stability or durability of the produced furniture.

## 2.2 Comfort

Comfort is the central goal in most furniture design systems, regardless of application: A person using the furniture should feel comfortable. While this goal is universally agreed on, there is no common definition of comfort. Instead, comfort is a highly subjective phenomenon, depending primarily on the person involved as well as the environment and other lesser factors. Research on comfort predates digital furniture design by a large margin. Depending on the field of applications, researchers developed various definitions and measures to represent comfort in their field of work.

This section will provide brief insights in the history of comfort and give an overview of comfort definitions and measures in modern furniture design.

### 2.2.1 Definition

Throughout decades of research, experts have attempted to get a better understanding about the meaning of comfort. In his research on the development of military plane seats, Hertzberg [Her58] defined comfort as the "absence of discomfort". Later research (Shackel et al. [SCS69], Richards [Ric80]) expanded this definition: Rather than seeing comfort and discomfort as a binary state, the researchers placed them on the respective ends of the same continuous scale.

Other researchers questioned this unidimensional, continuous nature of comfort. Branton [Bra69] conceptualized comfort as a neutral feeling, with only two discrete stages possible: presence or absence. However, the popularity and success of the usage of graded scales for comfort/discomfort, shifted the general believe in favor of a unidimensional scale.

In a different approach, Zhang et al. [ZHD96] argue that comfort and discomfort are different entities. The researchers associate discomfort primarily with physiological and biomechanical factors, whereas comfort is primarily linked to aesthetics.

With respect to the debate in literature surrounding the differences between sitting comfort and discomfort and their relation to one another, de Looze et al. [DLKEVD03] summarized the notations about comfort which were *not* under debate:

1. Comfort is a subjectively-defined state of feeling of the person involved.
2. Comfort is affected by physical, physiological and psychological factors.
3. Comfort is a reaction to the environment.

Following the practical studies of Zhang et al. (1996) [ZHD96], Helander and Zhang (1997) [HZ97] and Paul et al. (1997) [Pau97], de Looze et al. [DLKEVD03] formulated a theoretical model treating comfort and discomfort as different, complementary entities. The proposed model describes the underlying factors of comfort and discomfort at different levels:

(1) **human**, (2) **product/seat** and (3) **environment/context**. A visual representation of the model is shown in Figure 2.5.

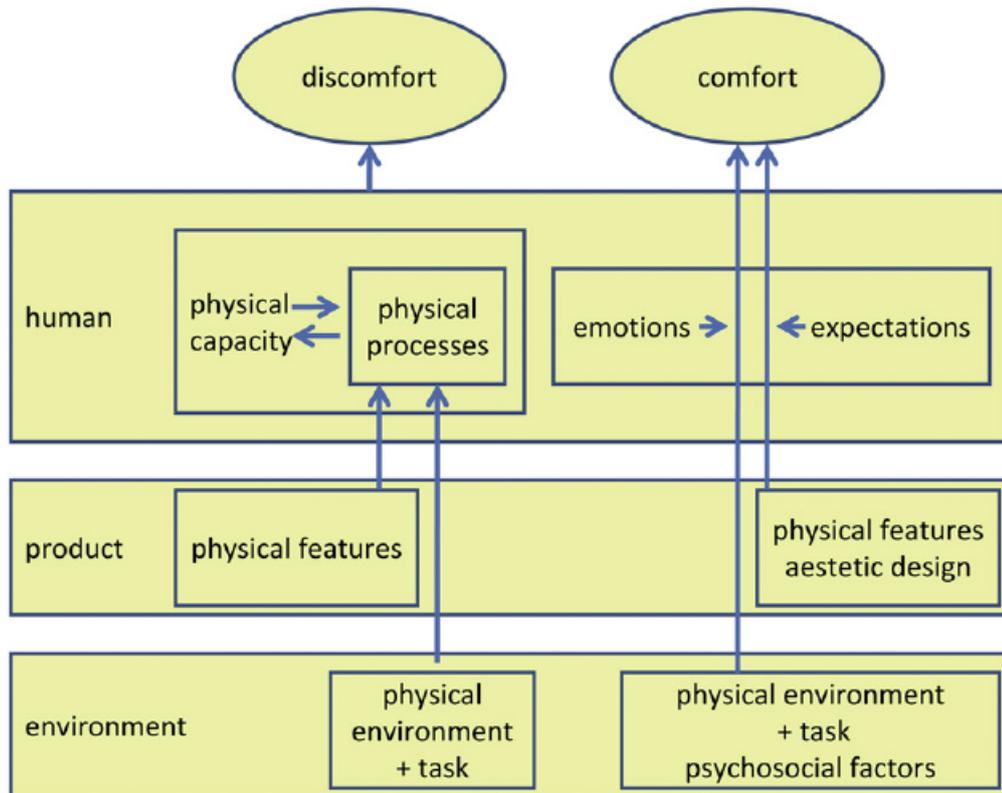


Figure 2.5: Model of comfort and discomfort by de Looze et al. [DLKEVD03]

The left side of the model describes factors concerning discomfort, which are mostly the result of physical processes. The physical characteristics of the seat, the environment or the task (in the context of a work environment) expose a sitting person to external factors (forces, pressure), which lead to a disturbance of the internal state of an individual (muscle activation, internal forces and pressure, etc.). As a result, further physiological, chemical and biomechanical responses of the human body are provoked, establishing the perception of discomfort.

The factors concerning comfort are portrayed on the right side of the model: Comfort at environment level is influenced by not only physical but also psycho-social factors

such as social support or job satisfaction. At a product level, the physical features and aesthetic design of the seat are the basis for comfort. At the human level, the feeling of comfort is affected by individual feelings and emotions.

### 2.2.2 Objective comfort measures

Identifying the factors of comfort and discomfort is only an initial step. In order to properly utilize the notion of comfort in furniture design, measures are required to quantify, rate and compare sitting comfort.

Historically, the most elementary way to determine comfort or discomfort of a seat is to keep note of the subjective feelings of its users. Early methods simply measured the length of time a person can spend in a sitting position on a given seat before they start to feel uncomfortable [Her58, Jon69]. In most studies, participants are directly asked to rate the comfort of a seat on a scale.

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 [DLKEVD03]: Objective methods are less dependent on a large number of subjects, less prone to measurement error and can be applied earlier in the design process. For the application in modern furniture design, the nature of digital design and automation makes it clear that the use of objective measures is highly beneficial.

Therefore, researchers have aimed to find a relation from subjective feelings to objective measures for comfort and discomfort. De Looze et al. [DLKEVD03] identified a variety of objective measures for comfort or discomfort from literature in medicine and ergonomics:

- **Posture and movement**

Various studies showed a slight correlation between discomfort and lumbar spine postures [SC99] as well as trunk flexion [EC87]. Bishu et al. [BHRS91] investigated spinal profile in detail and suggested a range of spinal angles where discomfort is minimal. Other studies [JL94] suggest that the spatial fit between a persons back and the back rest profile is related to comfort. While some trends could be identified, no statistically significant relation of postural deviations between people and their reported subjective discomfort.

- **Muscle activity**

A number of studies evaluated measurements of muscle activation and muscle fatigue by electromyography. An increase of back and shoulder muscle activation over time was suggested to lead to an increase of discomfort over time. High muscle activation has shown a tendency of leading to discomfort, most likely as a result of muscle fatigue. Very low muscle activation is also related to discomfort, probably caused by a hampering of blood circulation from static muscle activity. Overall, very few statistical evidence for a relation between muscle activity and comfort ratings of participants was collected.

- **Spinal load**

Measurements of stature loss (spinal shrinkage) as well as estimations of spinal loading forces have been the focus of a number of studies on sitting comfort. Eklund and Corlett. [EC87] suggested that seats with lower stature shrinkage and spinal forces correlated with low discomfort. A similar tendency was only found within specific subgroups of subjects by Michel and Helander [MH94].

- **Pressure distribution**

Throughout the reviewed literature, there is significantly more statistical evidence for a relation of pressure distribution to comfort and discomfort than for other objective measures. Numerous studies report correlations between measurements of pressure at the back rest or seat surface and comfort and discomfort ratings.

Yun et al. [YDF92] report a statistical correlation between uniform pressure distribution at the lower back and buttock area and local discomfort. Kamiyo et al. [KTOK82] suggest that car seats with a variance of pressure along the body's shape were generally considered comfortable. The measured pressure levels also highlight the importance of lumbar support for comfortable seating.

On the contrary, there exist large studies on chair comfort which found no statistical evidence regarding pressure distribution. Lee et al. [LFT93] concluded that the correlations are not high enough to make big design decisions based on pressure distribution alone. Other studies report significant correlation but lack the statistical evidence to back up their claim.

De Looze et al. [DLKEVD03] concluded that from the evaluated objective measures, pressure distribution showed the most clear association with the subjective ratings. Especially in car seat design, significant statistical evidence was provided backing the importance of pressure distribution as objective measure. De Looze concluded that the inclusion of pressure measurements besides subjective ratings could be valuable in the design process of other seats, such as office chairs, as well.

Zenk et al. [ZFBV12] concluded that the pressure distribution of the seat is correlated with the intervertebral pressure, suggesting a relation between experienced discomfort and the pressure in the spinal disc. In other studies it was suggested that low peak pressures and high contact areas [NNL<sup>+</sup>12] as well as average pressure [LTPSR08] are related to overall comfort.

### 2.2.3 Comfort in modern furniture design

In personalized furniture design the evaluation of comfort is a crucial task, as the primary goal of a personalized piece of furniture is to provide optimal comfort in a specific situation. The most direct and accurate way to receive feedback on comfort is through subjective rating. In the context of furniture design, this means to fabricate the seating surface and have subjects use and evaluate the furniture piece. Fabrication is mostly a very time consuming and expensive process, therefore subjective testing will in most cases only make sense for final evaluation. For feedback on comfort during the design

process, this is entirely infeasible.

Identifying objective comfort measures from subjective ratings is a necessary step in personalized furniture design. Therefore one of the key tasks in furniture design is to incorporate fitting objective comfort measures in the design process. From the subjective nature of comfort and discomfort it is clear that there can be no objective comfort measures that are universally applicable. Depending on the situation some assumptions about comfort have to be made and the appropriate comfort measures are chosen. The practical approach is to derive objective measures from the given environment and constraints. This is often achieved by simulating physical measures such as pressure distribution or muscle activation, for a person in a specific pose on a seating surface. Another approach is to follow known guidelines for an optimal fit. For instance, ergonomic standards, which were formulated from a large number of subjective ratings, dictate certain variables for the design of furniture in a desk environment.

In 1982, Bardsley and Taylor [IBMT82] developed an assessment chair, for the purpose of identifying a person’s seating requirements to assist in the design of seats for physically disabled patients. This assessment chair is highly configurable via a number of mechanical and hydraulic systems as well as exchangeable parts such as support surfaces. By fine tuning the chair’s configuration to a person’s subjective comfort requirement, an optimal seating configuration can be determined, which in turn can give insights about the statistical importance of the respective chair configurations.

In essence, this assessment chair aims to find a solution for the same problem which personalized furniture design systems are attempting to solve in a computerized manner: To find an optimal fit between a specific person’s body and a furniture model characterized through its configuration option. The objective measures used in modern personalized furniture design applications and how they are utilized in the respective algorithms is described in detail in the next section (2.3).

## 2.3 Related work in personalized furniture design

In previous sections, an overview of general principles and goals of personalized furniture design was presented. Various tasks were described, along with examples of respective approaches in historic and modern research. In this section, modern furniture design systems are presented in detail.

### 2.3.1 Pose-inspired shape synthesis

In 2017, Fu et al. [FCSF17] introduced a shape synthesis approach with the goal of creating hybrid shapes usable by humans. Shape synthesis refers to the creation of new 3D models as a combination or variation from existing shapes.

The workflow of the algorithm is visualized in Figure 2.6. The framework utilizes a database of pre-segmented man made shapes (e.g. chairs, bicycles, beds) which are mapped to human poses (e.g. sitting, lying) in a preprocessing step. In addition,

semantic information, such as relation between poses and interactive parts of the shapes is stored. The algorithm starts with a human pose as input and then explores a group of shapes for components that match the input pose in terms of human-object interaction. Relation graphs are used to represent the structure of the selected shapes. The graphs are utilized by a combination algorithm which merges the divided components to form the synthesized shapes. Created compositions can also be used in further synthesis steps to create models for multiple operators and poses.

While this work is not limited to furniture shapes, it serves as an example for pose-driven design. Contrary to other approaches, the pose matching works mostly based on semantics and skeletal transformations and its quality greatly depends on the segmentation and labeling. The actual body shape or comfort is not considered.

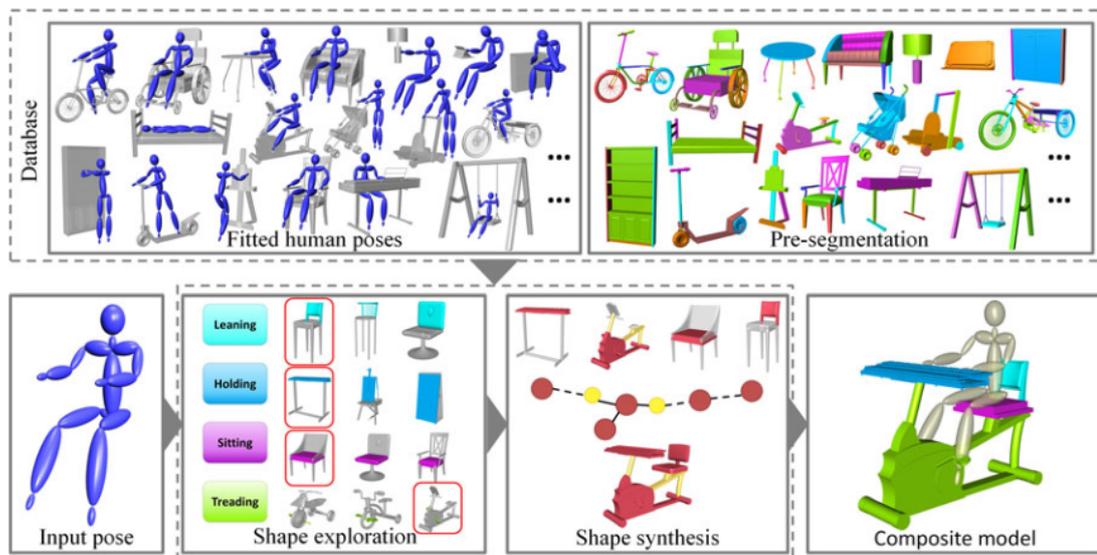


Figure 2.6: Overview of the pose-inspired shape synthesis algorithm by Fu et al. [FCSF17]

### 2.3.2 Posing and acting as input for personalizing furniture

Lee et al. [LCMS16] proposed a novel user centric furniture design process. The researchers aim to make digital design interfaces accessible for casual users by using poses and gestures, speech commands and augmented reality technology rather than conventional tools as seen in most computer-aided design (CAD) applications (Figure 2.8). The researchers conducted multiple studies analyzing how users utilize their body to specify distances and dimensions with gestures as well as the words they use describing their actions.

Utilizing the insights from the studies, the researchers developed a novel customization interface prototype utilizing a command language of full body gestures and voice commands to let users specify dimensions and positions of furniture objects. Using a head

mounted display showing a 3D visualization, as well as skeleton tracking techniques, several experiments were conducted where users could design various pieces of furniture in a room. Overall, the research successfully shows how a user's body can be utilized in personalized furniture design and provides insightful information about various methods of user interaction in furniture design systems.

### 2.3.3 Ergonomics-inspired reshaping of models

Zheng et al. [ZLDM16] introduce an interactive system that selects and adapts seating furniture for user-specified human body and input poses. In order to provide a high level of sitting comfort, the researchers follow ergonomics guidelines collected from reference work. For optimal comfort, a number of constraints are enforced for the parameters of the chair, such as height, width and depth of the chair seat. Furthermore, flat uncontoured seats, lumbar support, head support and arm rests are desired.

The algorithm starts with a user-specified human skeleton body and a geometric furniture model with multiple components as input. From analyzing the contact regions between the human body and the seating surface, geometric contact constraints are extracted from a set of ergonomic guidelines. Second, a deformation algorithm utilizes the geometric constraints to reshape the chair model to fit the human body shape. This optimization process, based on edit propagation, adjusts size, width and height of the shape components according to the constraints while preserving the shape structure. After deformation, the algorithm compares the deformed geometry to its initial state to analyze the shape-to-body deformation costs. These cost measurements allow the system to effectively rank and classify objects in regards to human poses. The proposed system can also be used for human-centric content retrieval: Given an specified body shape, the best suitable objects can be found and deformed to optimally match the body shape. The framework also supports shapes that are used by multiple people at the same time.

While the optimization process is based on ergonomics, the researchers note that the system does not precisely account for ergonomics and is therefore not yet suited to produce production-ready furniture models.

### 2.3.4 Automatic and personalized ergonomics

An entirely different approach at personalized furniture design is presented by Wu et al. in 2018 [WWT<sup>+</sup>18]: ActiveErgo is an attempt to automate personalized ergonomics in a desk environment. Instead of the design and fabrication of a personalized chair, the researchers designed an active furniture with the goal of optimizing a person's comfort during regular usage according to ergonomic guidelines (Figure 2.7). The system monitors a person's posture and automatically adjusts itself and guides the user into an optimal sitting position.

The active furniture system consists of a Kinect sensor for skeleton sensing and posture measurement, a motorized desk and computer monitor which can be controlled remotely adjusted in height (desk) or position and viewing angle (screen) and visual sensors to determine the exact position and height of the chair.

According to ergonomic guidelines the researchers consider a number of restrictions on the user's posture for optimal sitting position, specifying optimal head tilt, vertical view angle as well as positioning and alignment of upper arms, forearms, thighs and knees. As variable parameters in the environment the researchers consider chair and desk height, keyboard, chair and monitor position as well as the monitor's vertical viewing angle. After the optical sensor recognizes a user's posture, the system calculates the proper desk height which is automatically adjusted. The optimal chair height and positions are computed accordingly and visual instructions are used to guide the user in making the proper adjustments. The monitor's height and viewing angle are automatically adjusted matching the user's head position and angle.

The presented automated framework is not directly used for the design of general seating furniture, but rather for the design of a personalized desk environment. Nonetheless, the use of active furniture, could potentially be applied to other comfort measures and generic sitting applications beyond desk environments.

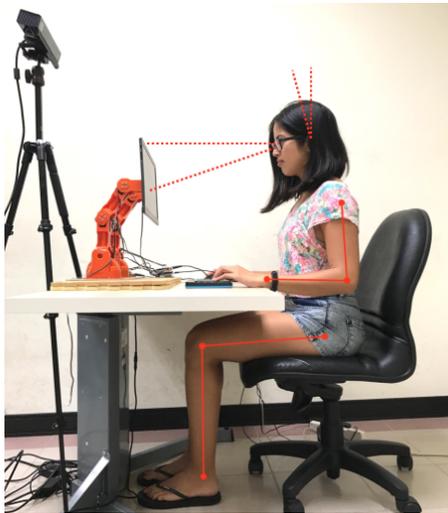


Figure 2.7: Optimal furniture arrangement and sitting position in ActiveErgo. [WWT<sup>+</sup>18]

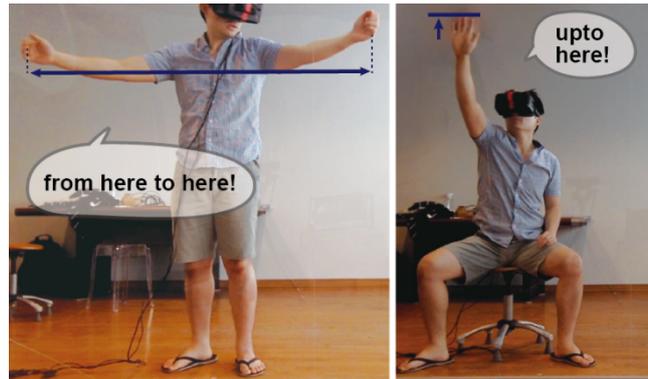


Figure 2.8: User input for furniture design via gestures and voice commands in the BodyMeter prototype. [LCMS16]

### 2.3.5 Interactive design of body-supporting surfaces

In 2018, Leimer et al. [LBRM18] present Sit&Relax, a pose-driven interactive furniture design approach. The algorithm presented in this thesis utilizes various concepts from the work of Leimer et al. and is designed to work in tandem with their proposed algorithm. Therefore, this section explains the basic concepts and implementation of the Sit&Relax algorithm in detail. An overview of the framework is presented in Figure 2.9.

The researchers propose a novel, interactive design method for pose-driven design of seating surfaces based on pressure distribution. Starting with motion captured input poses, the framework generates a seating surface from a user defined base shape that is

optimally fit to one or more given body shapes.

### Interactive design process

The input poses used in the design process can be selected from existing poses or recorded via a motion capture system. An interactive application allows the user to design an initial control mesh for the seating surface and place one or more poses into the scene. For varying body types, custom body shape models can be chosen. Further customization options allow users to fixate parts of the surface and to choose which body parts should be supported.

### Pressure distribution

In order to generate a fitting seating surface, proper comfort measures need to be considered. The researchers aim to find a plausible distribution of pressure acting on the body when resting in a given pose. According to De Looze et al. [DLKEVD03], the goal is to support the body at areas with high relative pressure while keeping absolute pressure peaks low by using a high contact area.

The researchers base their calculations on a strongly simplified physical pressure model considering only the normal component of the reaction force divided by the area the force is acting on. In order to greatly reduce computation requirements, the human body surface is considered rigid. While this model is a highly simplified approximation, comparisons with more accurate physical simulations have shown errors in a low range, which led the authors to claim that the chosen model is well-suitable for its intended purpose.

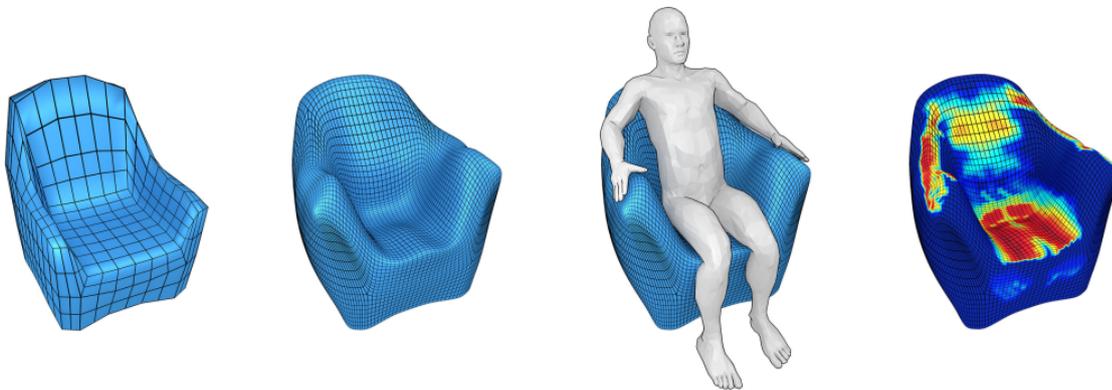


Figure 2.9: Intermediate steps in the Sit&Relax [LBRM18] framework: (1) Control mesh. (2, 3) Subdivided model fitted to a human body shape. (4) Visualization of the contact areas and pressure.

### Surface fitting

The approximated reaction forces, which are computed for a given input pose are utilized in the next step to fit the initial shape (used as control mesh) to the body. For this task,

a Catmull-Clark subdivision algorithm is applied to the chosen input shape to generate a detailed and smooth surface. The goal of the surface fitting step is to find the optimal vertex positions of the original control mesh so that its subdivided mesh has minimal distance to the input body mesh in the given pose.

In the iterative optimization process an energy function is minimized which is defined as a combination of point-to-point tangential distances, a Laplacian smoothing term, and a penalty term which prevents vertices of the seating surface from intersecting with the target body.

As an additional optimization step, the researchers included a pose relaxation step. After the surface is initially fitted, the human pose is slightly adjusted with the goal of minimizing the distance between the high priority regions on the body mesh and the corresponding parts on the seating surface. At the same time, penetration of the surface is avoided and large changes of joint angles are penalized.

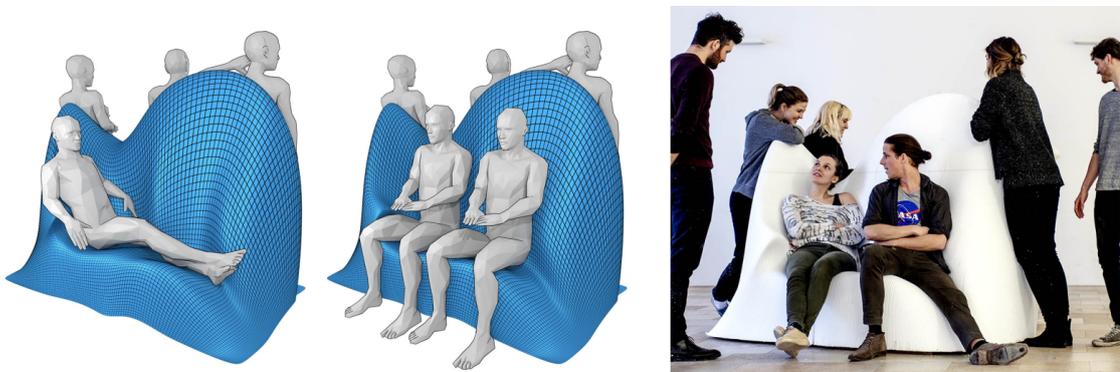


Figure 2.10: Multi-purpose surface created for sitting and leaning poses [LBRM18]. *Left*: rendered seating surfaces with two possible uses. *Right*: fabricated surface with design students.

### Results

The presented framework produces visually pleasing furniture models optimized for comfort for a given body shape in a specific pose. The framework supports the design of multi-purpose sitting surfaces, which are optimized to be usable in multiple different poses. The researchers have successfully fabricated furniture prototypes milled from Styrofoam using an industrial robot arm (Figure 2.10).

The biggest limitation according to the authors is the strongly simplified physical model for the body surface and pressure distribution. With the assumption that the human body is rigid, possible deformations of the human body when in contact with the seating surface are not considered. In the proposed framework, human body softness is approximated by allowing the surface to penetrate the body by a short margin.

The accuracy of the created results is mostly dependent on the quality of the control meshes used as input. The system is designed for interactivity and aims to allow designers to create appealingly looking shapes which still fulfill all functional requirements. More

precise control meshes lead to better fitted results. To create an optimal seating surface, a significant amount of manual design effort is required. As part of this thesis, we show that the accuracy of the results can be greatly improved by utilizing more suitable control meshes.

## 2.4 Optimization

In previous sections, the basics of personalized furniture design as well as practical applications in research were presented. In this section, the focus lies on the basics of mathematical optimization and an overview of optimization techniques relevant to the proposed algorithm in this thesis. In addition an overview of basic mesh optimization and smoothing techniques is given.

In general, mathematical optimization describes finding the best available (according to some criterion) values of an objective function within a defined domain. In the most basic case, optimization of a real function is performed by systematically choosing input values from the defined set and evaluating the function value in order to minimize or maximize it.

**Local optimization** is defined as finding local minima or maxima of a given function. Local minima are defined as points where the function value is smaller than or equal to function values of all nearby points, but not necessarily to all function values from the input domain. Finding local minima is a relatively trivial task using classical local optimization methods.

**Global optimization** has its focus on finding global extrema of a given set of function values. A global minimum is a point where its function value is equal to the lowest function value of all feasible points. Comparatively, finding global minima of a function is a significantly more challenging task.

**Principal component analysis (PCA)** is a basic method of multivariate data analysis [WEG87]. PCA is a mathematical dimension-reduction procedure where a large set of variables is approximated by a smaller set of linear combinations called principal components.

A set of values of potentially correlated variables is transformed into a set of values of uncorrelated variables called principal components. Starting with the largest possible variance, each succeeding component is orthogonal to the previous one and has the highest possible variance.

In the context of this thesis, principal component analysis is used to fit two-dimensional surfaces to 3-dimensional position data. In other words, PCA is used to find a plane which best represents the shape of a group of vertices in 3D space.

**Random sample consensus (RANSAC)** is a simple iterative method for fitting a model to observed data, capable of detecting outliers [FB81]. RANSAC is a non-deterministic

technique to estimate parameters of a model by random sampling of observed data, containing both outliers and inliers. The algorithm iteratively operates in two steps:

1. A minimal data subset, sufficient to determine the model parameters, is randomly selected from the input data set. A fitting model is then computed from the selected data subset.
2. The entire data set is evaluated for consistency using the previously defined model.

The consensus set is the set of inliers obtained for the fitting model. The algorithm is repeated until a threshold is reached or a fixed iteration count was exceeded. The result of the algorithm is the fitted model with the largest consensus set.

In the proposed framework, the RANSAC algorithm is utilized as surface fitting method. From a set of vertices in 3D, three points are randomly selected to form a plane which is then evaluated for inliers. The utilization of RANSAC and PCA techniques are described in greater detail in chapter 4.

### 2.4.1 Mesh smoothing

For the creation of functional and visually pleasing furniture models, operations on mesh geometry are required. Depending on the type of geometry used, different requirements for optimization exist. In the context of this thesis, the focus lies on connected polygonal meshes. In most applications triangular- or quadrilateral meshes are used.

A common application for mesh optimization techniques is to improve the visual quality of a model by smoothing out hard edges or noise in the surface. In general, smoothing algorithms attempt to adjust a mesh surface by modifying individual vertex positions in a way that preserves the original topology and general shape of the model.

Local smoothing techniques individually adjust the geometric position of each vertex in order to improve its local neighborhood of vertices according to some criteria. To obtain overall improvements in the mesh, the local smoothing operation is performed a number of times over all vertices of the mesh. For a large number of grid points the individual local operations are required to be computationally inexpensive.

The simplest and most commonly used local mesh smoothing operations is **Laplacian smoothing** [Lo85]. In this method each vertex of a surface mesh is moved to the geometric center of its neighbors (Figure 2.11, 2.12). While this method is computationally inexpensive, there is no guaranteed improvement in quality and even invalid results are possible. Smart variants of Laplacian smoothing incorporate a quality measure in order to prevent degrading quality: A local smoothing operation is only performed if the adjusted position would lead to an increase in quality.

In practice, a variety of more complex local smoothing operations and (global) optimization based techniques are used [EÜZ09]. Frequently, local smoothing is used in a

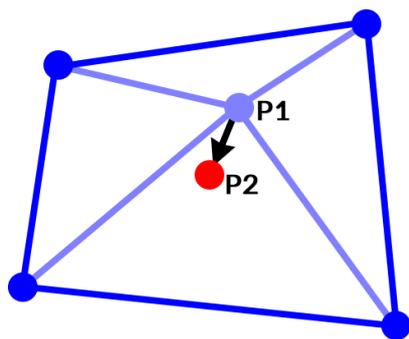


Figure 2.11: Laplacian smoothing step on a single vertex. The original point (P1) is moved to the geometric center of its neighbors (P2).

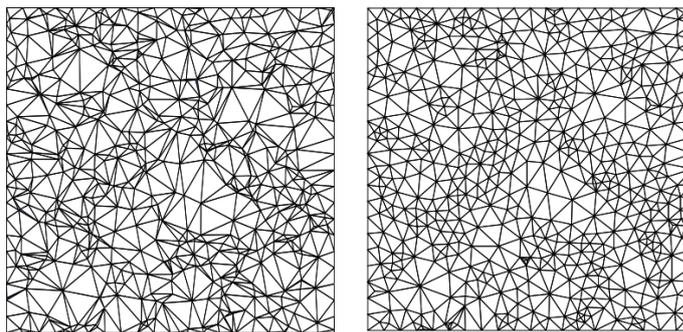


Figure 2.12: *Left*: Randomly generated triangle mesh. *Right*: The same mesh after 3 passes of Laplacian smoothing ([Fre97]).

combination with a global energy minimization in order to improve the quality of the results with the tradeoff of additional computation costs.

Freitag et al. [Fre97] evaluated a combined usage of smart Laplacian and optimization-based smoothing techniques. Optimization based techniques aim to find the minimum or maximum value of a function within a solution space using function and gradient evaluations. In the proposed algorithms, (local) smart Laplacian smoothing or optimization based smoothing is used depending on user-defined thresholds. The researchers concluded that low computational costs of Laplacian can effectively be combined with the high quality of optimization based approaches.

Erten et al. [EÜZ09] review a selection of mesh smoothing techniques based on different underlying methods. **Centroidal Voronoi tessellation (CVT)** is a special case of Voronoi tessellation, where the generating point of each Voronoi cell is equal to its center of mass. This structure has been shown to be well suited for mesh smoothing techniques and is utilized in different ways: In a proposed method by Alliez et al. [ACSYD05], each vertex is moved to a weighted average of their circumcenters (The weights depend on the physical size of the simplex.). Chen et al. [Che04] also incorporated the mesh density function to a CVT based smoothing method.

Chen [Che04] introduces the **optimal Delaunay triangulation (ODT)**, defined as "the triangulation that minimizes the interpolation error among all triangulations with the same number of vertices". In an ODT the length of all edges is aimed to be equal on the basis of the approximated function. Several proposed ODT based smoothing algorithms are claimed to be as computationally cheap as Laplacian smoothing but with reduced interpolation error.

Zhou and Shimada [ZS00] proposed a computationally simple **angle-based** smoothing approach. In their method, the local neighborhood of a vertex is formulated as a torsion spring system based on angles instead of distances as in Laplacian smoothing. In the original (local) version of the angle-based method, for each vertex, all incident pairs of adjacent angles are compared and adjusted to make them equal.

In a more computationally expensive, optimization based variant of the algorithm [XN06], the angle-based torsion spring system is formulated as a least-squares optimization problem.

A common flaw of basic mesh smoothing algorithms is that there is no discrimination between noise and salient features. Inspired from image processing, **feature-preserving** smoothing algorithms have been proposed. The general goal is to smooth and flatten most areas of the mesh while leaving corners and sharp edges untouched. Among other approaches, feature-preserving methods were introduced which aim to reduce the volume shrinking effect of Laplacian smoothing [VMM99] or alternate inward and outward diffusion of vertices in order to preserve features of the mesh [Tau95] [Tau00].

Svub et al. [SKŠ<sup>+</sup>10] introduce a simple to use algorithm which adjusts vertices based on a covariance matrix of neighborhood triangle normals. The displacement vector of the Laplacian operator is weighted with the eigenvalues and eigenvectors of the covariance matrix. The algorithm is robust and stable as well as applicable for a wide range of different meshes.

### 2.4.2 Subdivision

While smoothing algorithms aim to improve the visual quality of a mesh by smoothing out noise and flattening the surface, they do not alter the topology of the mesh. The achievable quality of smoothing results is limited by the vertex density in the original model. Therefore, smoothing techniques are usually not applicable for models with a very low number of vertices.

In order to increase the smoothness of an arbitrary models, meshes are often required to be refined into a much finer grid. For this task, subdivision algorithms are used, which aim to visually refine the surface by adding additional vertices into the mesh grid and thereby increasing its total vertex count. A subdivision surface is a representation of a smooth surface, recursively created from a coarser polygon mesh. Commonly limited by a user-defined number of iterations, a refinement scheme is recursively applied to a given mesh, splitting up faces and inserting new faces and vertices into the mesh. The mesh resulting from one iteration is a more refined version of the previous iteration's result and serves as the input mesh for the next step.

Most subdivision algorithms are based on a publication from 1978 by Catmull and Clarke [CC78]. The researchers proposed a method to recursively generate B-spline surfaces from arbitrary meshes - a technique known today as *Catmull-Clark subdivision surface*. The algorithm creates refined meshes consisting only of non-planar quadrilaterals (See Figure 2.13).

Jos Stam [Sta98] proved that Catmull-Clark subdivision surfaces could be exactly computed at an arbitrary iteration step without recursive subdivision.

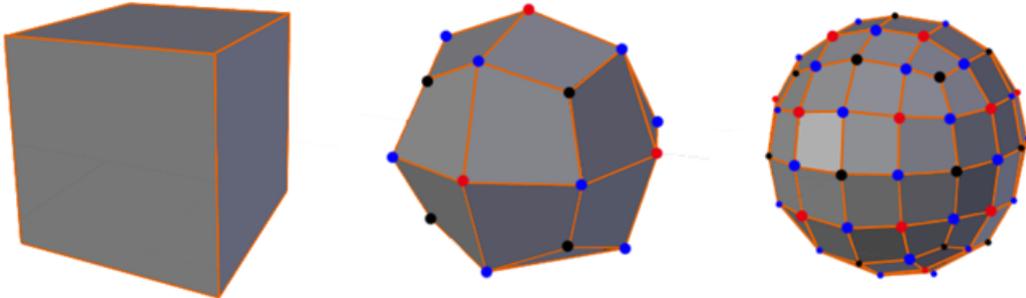


Figure 2.13: Catmull-Clarke Subdivision Surface applied to a cube [CPH15]. From left to right: Original mesh; mesh after one iteration; mesh after two iterations.

While subdivision algorithms create smooth and visually appealing surfaces, the increase in vertex density is undesirable in numerous applications as it leads to a direct increase of computation costs for most algorithms operating on vertices or faces. In practice, subdivision algorithms can be used combined with smoothing and optimization techniques.



# Pose-Driven Design

In this thesis we propose a new personalized furniture design framework. Now that background information and relevant techniques for the proposed system have been presented, the following chapters will provide a detailed look into the ideas and motivation that inspired the system proposed in this thesis, as well as concepts, techniques and methods that were utilized.

In this chapter we recall the goals and motivation for the development of this furniture design framework and provide a quick overview of the methodology behind it. This includes a clear definition of the goals and requirements as well as a description of all the tasks that need to be completed in order to fulfill these goals. In addition, we give an overview of the algorithms that are needed and go into more detail about additional resources that are required for this framework.

## 3.1 Goals and motivation

Before the proposed methods and techniques for the algorithms are presented, we need to clearly define the required goals for this framework. As we recall from the introduction chapter, the core motivation for this thesis is to provide a computerized personal furniture design system. Furthermore, the goal is to follow a human centered pose-driven design approach.

As the first goal, we discuss **furniture design**: The central task of the proposed algorithm is to produce a digital model of a seating furniture piece. While fabrication of physical seating furniture was discussed in the previous chapter, the focus of the proposed algorithm lies primarily on digital models. Physical stability of the models or material usage is not considered. However, we do not completely disregard the possibility of fabrication. Instead, the digital models are designed with the intention of potentially allowing fabrication in the future. The created digital furniture models should represent a connected surface mesh usable as seating surface.

Arguably the most important goal is the **personalization** aspect of the system. The furniture design process is centered around a specific person in a specific pose. At the same time, the system needs to be able to support a wide range of different human body shapes and sizes as well as different sitting poses. Therefore, an important task is to develop an adjustable and highly configurable design system which is able to produce seating furniture for large variety of body shapes and poses. This includes analyzing and prioritizing the various positions and transformations of individual body parts of a human skeleton in various sitting poses.

The degrees of **automation and interactivity** are further important factors of the proposed furniture design system, distinguishing it from other work. A common approach at furniture design is to provide a design interface to the user with a high level of interactivity to allow users to manually shape and personalize furniture pieces. However, significant disadvantages of this methodology exist in some applications, which we want to address in this work.

For one, user interaction is time consuming and requires manual effort. Our chosen automated approach aims to limit the required manual design effort to a minimum. Second, relying on interactive input during the design process assumes that the user has a certain level of knowledge about furniture design as well as the system and its controls. Depending on the application, it can be difficult for non-experts to create functional and visually pleasing furniture pieces. Therefore, the goal for this work is to create an automated furniture design process, while the personalization aspect of the system is handled by the selection of the input poses and body shapes.

As a direct result of the aforementioned design decision, there are less options for (expert) users to adjust and further personalize results. In order to overcome this limitation, we have to consider higher requirements for visual quality and accuracy for the initial results.

Another major goal of the proposed framework is directly related to the chosen design methodology for user interaction. The furniture models created by this system should not only be usable as seating surfaces, but also serve as input shapes for further processing in the **Sit&Relax** framework by Leimer et al. [LBRM18]. As mentioned in Section 2.3.5, the framework relies on user defined input shapes used as control meshes for an optimization process. A goal for the system proposed in this thesis is to provide surfaces models which serve as an optimal control mesh for the Sit&Relax framework for the given pose and body shape. While the algorithm by Leimer et al. is able to fit simple control meshes to a human body with good precision, practical results show significant differences in quality with using generic primitives compared to manually crafted control meshes that were created to roughly fit the pose. Our proposed system was inspired by the assumption that using a generic model that is pre-fitted to a pose as input control mesh can further increase the quality and precision of the results, without the need for manual fitting.

## 3.2 Overview

This section gives an overview about the approach that was chosen in order to fulfill these goals. The proposed system consists of a number of components required to solve the individual tasks (Figure 3.1):

- **Template furniture model**

A highly configurable generic furniture model was developed. The model was defined as a simple connected surface mesh consisting of quadrilaterals, where the individual faces can be matched and fitted to corresponding body parts according to comfort measures. The model is parametrized through the position and orientation of individual planes supporting the faces of the model, aiming for an optimal overall fit between the entire body and the seating surface.

The template model is designed to be able to fit a large variety of body shapes and poses with the goal of providing a solution for generic sitting applications. In separate development stages, two different versions of the template model have been developed: a simple initial model, limited to basic sitting poses as well as a more complex model, capable of supporting a greater variety of poses.

- **Surface fitting**

A surface fitting algorithm was developed in order to fit the generic template model to the chosen human body shapes. In our algorithm, the individual faces of the template model are fit separately to corresponding body parts, while preserving the overall structure of the model. In order to create suitable seating furniture, comfort measures are utilized in the fitting algorithm. The chosen measures are based on pressure distribution as proposed by Leimer et al. [LBRM18]. Each vertex of the body shape is weighted based on its importance for equal pressure distribution. The fitting step's goal is to minimize the distance of the surface to important vertices in order to maximize the total sum of supported vertex weights. Several different surface fitting methods have been evaluated for this task. The proposed framework utilizes a RANSAC based surface fitting approach to determine the optimal position of the planes.

- **Mesh optimization**

The initially determined configuration of the template model is further optimized in a non-linear optimization step. The initial surface mesh is refined by an iterative optimization process aiming to increase the visual quality of the model by smoothing out edges and increasing the planarity and regularity of the mesh. At the same time, transformations on supporting areas of the model are penalized in order to preserve a maximal amount of comfort.

- **Weighted fitting**

Subsequent to finding an optimal configuration for a given pose, the created model is automatically set up to be directly used within the Sit&Relax framework for further optimization, where it is subdivided and iteratively fit to the body shape.

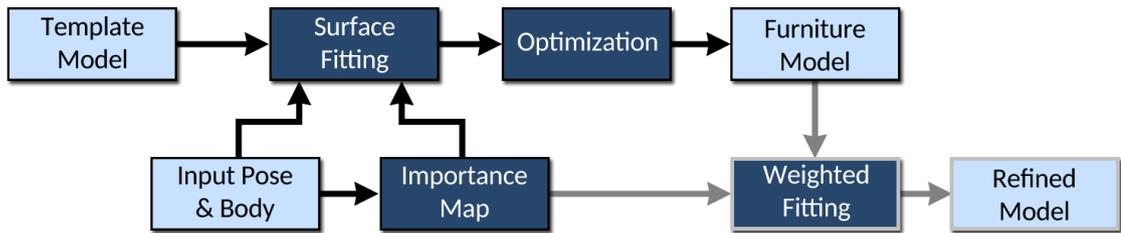


Figure 3.1: An overview of the proposed system: The framework starts with an *input pose and body* which are used to compute an *importance map* using the algorithms from Leimer et al. [LBRM18]. The proposed template model is utilized along with the importance map and the body to perform the initial *surface fitting* process. The matched template model configuration is then further refined in a *non-linear optimization* process to create the finalized personalized *furniture model*. In an optional step, the refined model is used as control mesh in the *weighted fitting* algorithm by Leimer et al. to create a further improved subdivided model.

### 3.3 Human body representation

As the proposed furniture design framework follows a human centered approach, an accurate **representation of a human body** is essential. This includes both the surface mesh as well as the hierarchical skeleton of the human body.

In order to simulate pressure distribution at contact areas of the body, a detailed human surface model is required. In addition, the model needs to be adjustable for a variety of body shapes. A suitable representation of human body shapes is provided as a Blender plug-in by Manuel Bastioni [Bas18], which allows us to configure and export human body models in regards to gender, mass and stature.

While the goal of this thesis is to provide a personalized seating solution, acquiring an accurate 3D representation of a specific body shape would require extensive effort as well as expensive 3D scanning hardware. Therefore we settle on a generic solution using the aforementioned parametrized models to get a close resemblance to a specific person's body.

We assume that the body surface models are detailed and accurate enough for the

requirements in the proposed framework. Choosing a more detailed representation would not necessarily improve the accuracy of our framework for a number of reasons:

- As the human body models represent a *generic* body shape, a further refined human body model does not improve the accuracy for a *specific* person.
- The proposed comfort measures by Leimer et al. are based on a highly simplified physical model and a number of assumptions which would limit the accuracy independently of the chosen body model.
- A large number of vertices in the human body model would directly increase the required computation effort in most stages of the proposed algorithm.

In addition to 3D models of body shapes, a suitable **representation of sitting poses** is required. Any movement of the human body can be seen as a transformation on the joints of its underlying skeleton, whereas pose is a static state of these transformations. In order to portray human sitting poses, we need a digital representation of a human skeleton.

For this task, we look into common notations for digital animation files formats [MM<sup>+</sup>01]. Commonly, a skeleton is composed of a number of bones in a hierarchical structure. A bone is the smallest entity of the skeleton that can be individually transformed in an animation. Each bone can be subject to a number of transformations, such as scale, rotation or translation. A dynamic animation is composed of a number of frames, where in each frame the transformations of the individual bones of the skeleton are described. We can directly apply this notation to static poses as well: A sitting pose corresponds to a single frame of an animation. For realistic sitting animations, only rotate-transformations are relevant. Translations are only required to describe an initial default transformation of the skeleton. A human skeleton model can be moved into a sitting pose, by rotating its bones into the appropriate position.

Analogous to the human body shape representation, a sufficiently detailed **skeletal hierarchy** must be chosen. To portray accurate transformations, the skeleton should be a close approximation of actual joints in a human body and detailed enough to represent human sitting poses. For some body parts, such as limbs, it is far easier to keep the chosen representation close to its human counterpart, for more complex structures such as the spine, an approximation will be necessary. The skeleton model that was chosen for the proposed framework is a hierarchical structure consisting of 21 bones and shown in Figure 3.2. For an accurate portrayal of sitting poses a detailed representation of the spine is beneficial and allows for refined deformations along the back.

In order to represent human bodies in sitting poses, the body shape models need to be transformed according to the underlying skeleton. For this goal the body shapes need to be manually mapped to the bones of the skeleton. In this *rigging* process, for every vertex, specific weights are assigned for different bones of the skeleton. This mapping

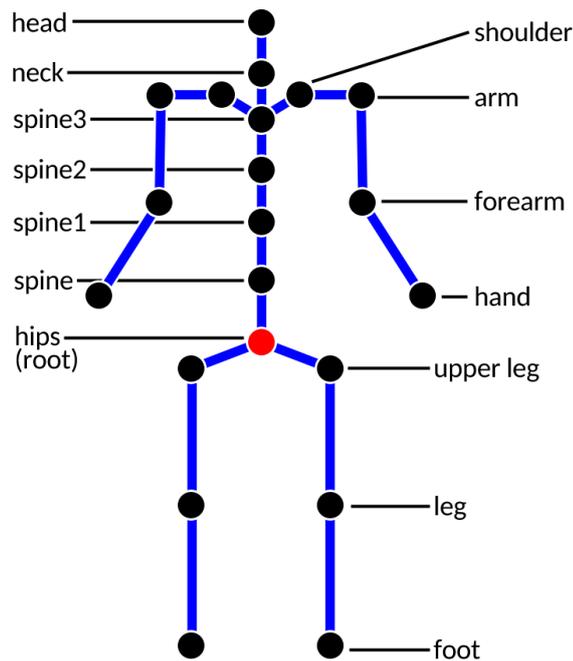


Figure 3.2: The human skeleton model used in the proposed framework, consisting of 21 individual bones.

tells us which vertices of the body shape should move with the corresponding bones of the skeleton. Once the body shape and skeleton is properly set up, we can freely combine any shape variation of the body with arbitrary poses. The skeleton and rigged body shape models used for the proposed system are taken from the work of Leimer et al. to ensure the results are consistent and compatible with the framework.

### 3.4 Sitting poses

As mentioned above, sitting poses are represented as animation frames describing the rotation of the skeleton's joints. As the goal of this work is to create a usable furniture design system, using realistic and accurate sitting poses is highly beneficial. Manually creating realistic animation data for human poses is difficult and tedious task and often takes skilled artists large amounts of time using 3D modeling software.

Therefore, a logical decision would be to rely on recorded poses using motion capture tools. On one hand, the use of motion capture techniques eliminates time consuming posture configuration effort as the poses are directly recorded from actors. On the other hand, motion capturing requires significant setup time and cost. Cheap and simple to use motion capture technology exists but usually with a tradeoff in accuracy. In general, both manual animation editing or motion capture techniques are suitable to acquire poses for this approach.

For this thesis, a collection of motion captured poses were provided by Leimer et al. consisting of various sitting, lying or leaning poses. The poses were recorded with the users fully supported in a comfortable position. Examples of various sitting poses are shown in Figure 3.3.

The proposed framework was designed with the goal of supporting a large variety of

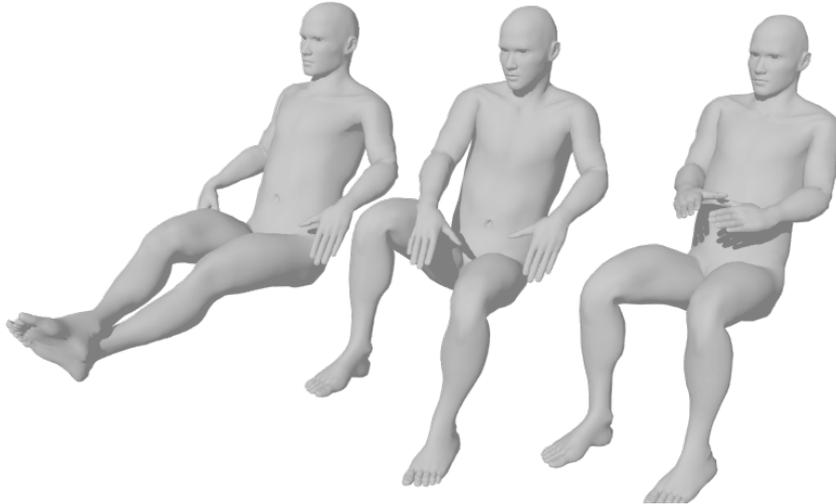


Figure 3.3: Rendered images of human body models [Bas18] in various motion captured poses provided by Leimer et al. [LBRM18].

sitting poses. Therefore, it was necessary to evaluate a large number of poses during the development process in order to create a robust system. For the design of the algorithm, a tradeoff between accuracy and variety is necessary. In an ideal case, a generic algorithm could provide accurate solutions for all sitting poses. This is infeasible in practice as there can be large variations between individual poses. Tuning the algorithm to be able to support extreme cases would usually result in a general loss of accuracy or the inability to support a different group of special poses.

While variety is important, we focus on a high level of accuracy for the proposed framework. As the goal is to create comfortable seating surfaces, we disregard most leaning and lying poses and focus on various sitting poses. While it is not possible to clearly classify poses into disjoint groups, it is beneficial to define which general classes of poses are supported by the system. For all available sitting poses, we attempt to identify and classify special cases that are impossible to solve following the nature and design of the algorithm. In general, we can classify potential input poses into a number of groups:

### 1. Regular sitting poses

An ideal pose for this algorithm is a person sitting in an upright or backwards leaning position with a straightened back and his or her limbs in a natural position. This also includes minor variations such as an angled back or legs in a lifted position.

#### 2. **Special sitting poses**

This category consists of sitting poses which feature various unnatural or uncommon positioning of certain body parts. These poses are part of the allowed input set but require additional special case handling in the optimization process. This category includes poses with crossed legs or where the person is leaning forward.

#### 3. **Impossible poses**

Certain groups of poses can not be properly supported by the algorithm and are excluded from the possible input set. This includes all leaning poses as well as all sitting or lying poses where the body is oriented sideways. Furthermore, we consider all sitting poses where the person’s feet are located below their body unsuitable.

## 3.5 Comfort measures

The end result of the proposed framework is a digital model of a personalized piece of furniture that is comfortable for a specific person in a certain pose. In the previous section, the goals of supporting body shape models and sitting poses were discussed. In this context, a number of questions remain: Under which circumstances is a body part considered supported and how is this related to comfort?

As described in Section 2.2, comfort is a subjective feeling of a person using a seating surface. Subjective evaluation of comfort would require fabrication of the chair models and is infeasible for most digital furniture design systems. In order to include comfort in the furniture design process, objective comfort measures are required. From several studies and evaluation of comfort measures in previous research, we conclude that pressure distribution can be seen as the most reliable objective measure for sitting comfort.

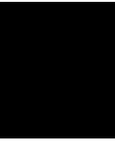
For the proposed framework, we utilize comfort measures based on **pressure distribution**, following the notation by Leimer et al. [LBRM18]. As described in more detail in Section 2.3.5, the measures are based on the physical pressure acting on the body when resting in a specific pose. Maximum comfort is assumed when the pressure is uniformly distributed on an as large area as possible. Based on a simplified physical model, the algorithm provided by Leimer et al. computes an importance map on the human body surface for a given pose, proportional to the pressure distribution.

The created **importance map** essentially tells us which regions of the body in a given pose have the highest priority for support. This leaves us with the decision how to utilize these comfort measures for the surface fitting and optimization algorithms. The general idea is to minimize the distance between the supporting surface and the important parts of the body model. The importance values are represented as per-vertex weights on the body shape model. The importance values for each vertex are normalized (i.e. divided by the surface area) to compensate for unevenly distributed vertex density in the model. As comfort measure in the proposed framework we consider the total sum of vertex

weights that are within supporting distance of the surface. The supporting distance is a global threshold defining how close a body vertex has to be to the surface to be considered supported. This threshold is used to compensate for inaccuracies in the body shape geometry.

The human body is considered rigid in the physical model used to compute the importance map. To simulate the softness of the body, we define a *softness threshold* value for each body part. The softness threshold acts in the opposite direction of the supporting distance and defines how far the supporting surface is allowed to intersect with the shape of a particular body part.





# Template Model

In this chapter, the structure and design of the template furniture model is explained in detail. This generic model is the main contribution of this thesis and core resource in the design process. The following sections summarize and further clarify the goals and requirements for this model. Furthermore, the methodology behind the generic template model is presented in detail.

In separate stages of the development process two variants of the template model were created: In the initial approach, a simplistic model was designed, consisting of a simple hierarchy of planar quadrilaterals which are matched to a human body. The second model, based on the initial one, utilizes a more complex hierarchy of non-planar quads for a more refined fit of the body shape. The concepts and techniques which are utilized for the development of both models are presented in this chapter.

## 4.1 Requirements

As we recall the overall goals of the framework in this section, the specific requirements for the template model are defined. From the previously mentioned general goals, we derive the requirements for the template model:

- The general goal is to create a seating surface model that closely fits a human body in a specific pose. A basic mesh structure is required that is able to represent a seating surface. A **quadrilateral mesh** is chosen for simplicity as well as suitability for the task. The basic structure of the model is derived from real-world configurable chair models which are generally composed of rectangular shapes. We conclude that it is easier to fit rectangular than triangular faces to shape of a human body.

In order to provide a close match to a human body, the template model is required to

support individual body parts. The mesh resolution of the template model surface needs to be high enough to provide a close fit to the body shape and individual limbs.

- The model needs to be **configurable and variable** in order to match all poses and body shapes. The parameter space of the template model needs to be complex enough to account for any joint orientation in sitting poses as well as variable length and sizes of individual body parts. The template model needs a sufficient number of degrees of freedom in order to allow suitable configurations for all poses and body shapes from the input domain.
- In addition, the resulting surface must be usable as **control mesh** in the Sit&Relax framework. The weighted fitting algorithm from Leimer et al. utilizes Catmull Clark subdivision surface which requires a single manifold surface mesh. Therefore, we require the template model to consist of a single connected surface rather than individual geometric objects.
- The furniture design process proposed in this thesis mostly follows the *form follows function* paradigm: Creating a comfortable seating surface is the primary goal. 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**. The quadrilateral faces of the mesh should be as planar and rectangular as possible. Furthermore, the vertex positions in the mesh should be close to a regular grid. Our chosen notation of visual quality is mostly based on subjective evaluation of visual examples of seating surfaces. While research on visual quality of low-poly surface meshes is limited, we derived the concepts of planarity and regularity as visual quality measure from research on the creation of free form architectural surfaces ([LXW<sup>+</sup>11],[ZSW10]).
- Physical constraints are not evaluated in detail, however the model was designed with **physical plausibility** in mind as well as the possibility for fabrication. The created furniture model must not contain self-intersecting geometry. In addition, the template model geometry should be connected to the ground and provide an (estimated) stable base.

## 4.2 Initial template model

The initial template model was designed with the goal of fulfilling the above requirements with simplicity in mind. The structure of the the template model was mostly inspired by

existing configurable chair models, which are often used for patients in hospitals or a doctor's office.

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 such as connectivity and planarity.

#### 4.2.1 Model structure design

The coordinate system used for all geometric shapes used in the proposed framework is based on the following notation:

**X-Axis: Forward/Backward.**

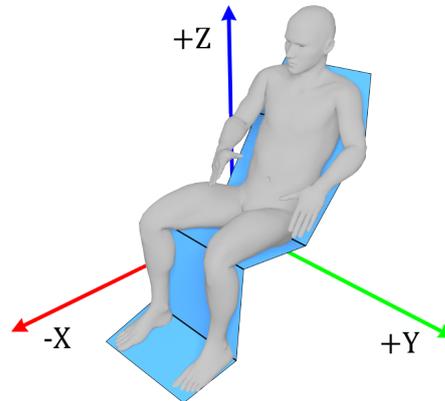
The negative X-direction is the facing direction of a person sitting in an upwards position.

**Y-Axis: Left/Right.**

Left refers to the positive and right to the negative Y direction.

**Z-Axis: Up/Down.**

The positive Z-Axis represents the upwards direction in the scene.



For the design of such a template model structure, a number of considerations arise. We recall the chosen representation of the human skeleton: The hierarchy consists of 21 body parts that can be individually oriented in a pose. For the template model, a corresponding surface structure is needed that is able to match this hierarchy. For this task we need to determine a mapping between body parts and faces of the template model which involves a number of decisions:

The initial decision is to determine which number of quad-faces is required. This means, we determine which body parts need individual support and which can be grouped together and mapped to a single face.

For the individual quad-faces of the template model we need to define suitable degrees of freedom. This means we need to define what transformations are allowed for the faces, similar to how elements of an adjustable chair can be moved in certain directions (i.e. the backrest of a chair can be angled backwards, which corresponds to a rotation in its Y-axis)

All those decisions have to be made in line with the defined requirements. Therefore, we need to consider the possibilities of how to enforce these constraints in a chosen template model structure. On the other hand, it is crucial to evaluate the potential limitations on the model structure that are imposed by the constraints.

For the development of the initial template model, a number of different structures were evaluated.

### Fixed hierarchical model

The first prototype was inspired by the basic structure of conventional adjustable chair models. The proposed model consists of a hierarchy of 6 connected faces, with variable length (X-axis) and angle (Y-axis). The individual faces are mapped to groups of body parts: (1)*feet*, (2)*legs*, (3)*hips and thighs*, (4)*lower back*, (5)*upper back and shoulders*, (6)*head*. The model's faces are linked in a hierarchical structure, with the shape supporting the hips and thighs serving as root element. In addition, two extra quad-faces could be freely placed along the side of the model to serve as armrests.

A number of advantages and disadvantages of the fixed hierarchical approach were determined:

- The model fulfills all visual requirements to a great degree. All faces of the model are perfectly rectangular and planar.
- The template model closely resembles existing adjustable chair models and could be fabricated from conventional parts and materials in a similar way. It might be possible to replicate certain configurations of the model with existing chair models.
- The model's seating surface is a connected mesh of quadrilateral faces and therefore suitable as control mesh for the weighted fitting algorithm by Leimer et al. with exception of the armrests.
- The hierarchical structure of the model makes it difficult to perform local fitting to individual faces of the model. Adjusting the parameters of inner shapes of the hierarchy would propagate changes to linked faces of the model. The fitting process therefore becomes a global problem and would require the application of different techniques such as inverse kinematics.
- The biggest disadvantage of the fixed hierarchical approach is the limited number of degrees of freedom for individual faces of the model. The limitation of disallowing tilt rotation (X-axis) greatly reduces accuracy and quality for irregular sitting poses. This approach is only suited for straight sitting poses and would therefore greatly limit the variety.

### Hierarchical faces model

Considering the limitations of the fixed hierarchical model, the proposed approach was modified in multiple ways with the goals of increasing its degrees of freedom while preserving the visual quality and simplicity.

The main difference to the fixed hierarchical model is that the modified variant's faces are allowed to rotate freely in multiple directions. This allows for a much greater versatility and enables the model to represent tilted surfaces to support angled posture in irregular poses. The added degrees of freedom also allow surfaces to better support asymmetrical orientation of a person's legs in a pose.

The basic hierarchy in the improved model is the same as in the base variant. This includes the related downsides such as the hierarchical dependency and propagation of changes.

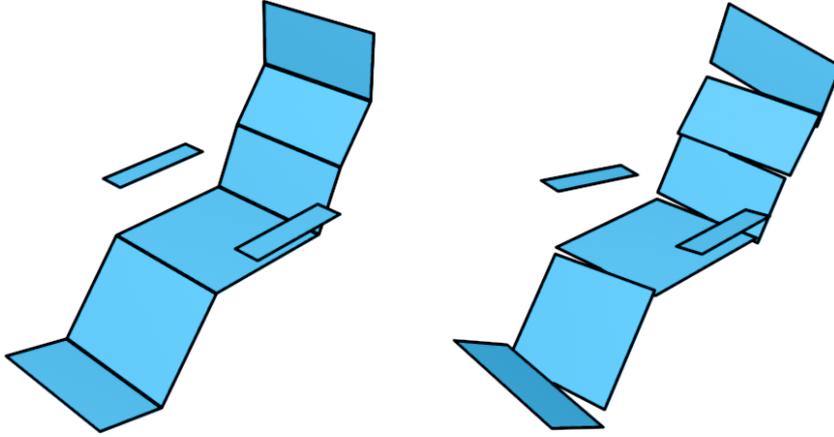


Figure 4.1: Visual comparison of the hierarchical model variants. *Left:* In the *fixed* variant, the model's faces are perfectly rectangular. *Right:* In the *hierarchical faces model* additional rotations along the X and Z axis are allowed, resulting in a loss of connectivity.

The largest downside of this modified approach is the loss of connectivity. As the base variants limited degrees of freedom would enforce connectivity for adjacent faces along their rotation axis, this feature is lost in the modified variant (See Figure 4.1). As connectivity is required for the weighted fitting algorithm and therefore a necessity, additional measures are required to connect vertices of neighboring faces. For this task, we aim to connect adjacent faces while preserving their planarity. As tradeoff we have to settle with losing regularity of the quad-faces as they are no longer rectangular. In order to connect adjacent faces, their respective planes are intersected and the corresponding vertices are joined at a point on the intersection line. This process ensures both connectivity and planarity. At the same time, it locks all affected vertices on their original planes and therefore preserves most of the contact area with the body.

However, this method introduces a number of limitations and disadvantages: Large angle differences between adjacent faces can result in greatly distorted intersection lines. While neighboring faces are ideally connected with a line orthogonal to the orientation of the underlying plane in the hierarchy, the intersection lines can be almost parallel in the worst case.

These extreme deformations lead to a number of significant issues. Distorted intersection lines imply a loss of regularity of adjacent quad-faces as they lose their rectangular shape. This does not only affect visual quality, because in the worst case quad-faces can degenerate into triangular shapes which also affects the functionality of the algorithm. Another significant downside of this plane intersection step is that the contact area with

the body shape can grow smaller, reducing the overall level of support and comfort the seating surface can provide. This effect is even more significant with distortions in the intersection lines.

In order to prevent these errors, additional constraints on the rotation angles of the faces in the model need to be introduced. While restricting the allowed rotation angles prevents most of the mentioned issues it also limits the amount of poses that can be supported by the model.

### **Free faces model**

A third approach for the initial template model was developed which follows a slightly different method. The basic structure of faces is the same as described in the fixed hierarchical model. However, the free faces model completely replaces the hierarchical structure of the previous approaches with a set of freely arranged planes. The basic idea behind this approach is to modify the position and orientation of the faces independently, rather than in a hierarchical order.

The main advantage that we seek to achieve with this method is that it allows us to perform local optimization techniques on individual planes in the model without propagating changes to adjacent faces.

The fitting process is performed in two stages. In independent surface fitting steps, the underlying planes of the model are positioned to optimally support their respective body parts. From the initial positioning of the planes, the vertices and faces forming the actual structure of the model are generated from intersection lines between neighboring planes. This method closely resembles the connectivity ensuring step in the hierarchical faces model.

As a result, the same limitations regarding intersection lines need to be taken into account. In order to restrict the rotation angle of faces according to adjacent faces, we need to include hierarchical dependencies: Surface fitting steps are performed in a fixed hierarchical order, with the orientation of the parent plane taken into account in each respective step. While this introduces error propagation, the free faces model is still significantly less limited than the hierarchical variants.

Overall, we concluded that the free faces model is better suited for local surface fitting methods as it greatly simplifies the optimization process. The free faces approach was therefore chosen for the design of the generic template model.

### **4.2.2 Surface fitting techniques**

The optimization process for the generic template model is based on surface fitting. The underlying planes of individual faces should be properly matched with their corresponding body parts in a way that maximizes the contact area and ensures maximal comfort. Our

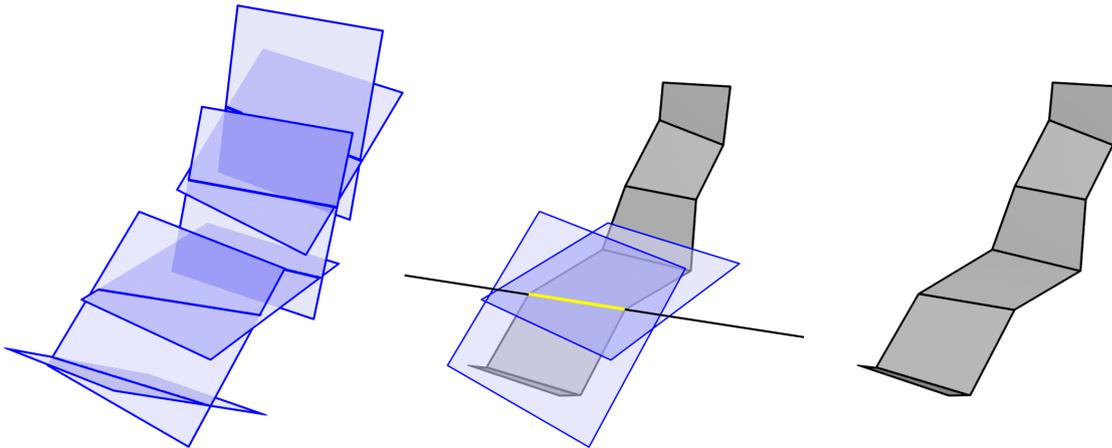


Figure 4.2: Overview of the free faces model. *Left:* The model consists of multiple planes that are positioned independently in the scene corresponding to their body parts. *Center:* (Hierarchically) adjacent planes are intersected. *Right:* The surface geometry mesh is generated according to the intersection lines.

chosen comfort measure is therefore the total sum of weights for each vertex of the body model that is within a defined supporting distance of the plane.

For this task, we evaluated a number of surface fitting techniques to find the most suitable method according to the following requirements:

1. The optimization method must be able to find the optimal plane according to the provided importance weights.
2. Intersections with the body beyond a defined softness threshold must be avoided.
3. The method should be suitable for all body parts and shapes. In an ideal case, the same algorithm is applicable for a single limb or a group of body parts.
4. In order to prevent errors and deformations, we need to be able to introduce constraints into the optimization process.

From the evaluation of various surface fitting methods we gained the following insights:

- **Principal component analysis (PCA):**

The general approach is to find an optimal plane over all vertex positions of a body part, using their importance as weights. The computed plane is then shifted along its normal to avoid intersections with the body. The PCA approach does not directly optimize for the maximal sum of vertex weights, but its results are sufficiently accurate.

PCA is fast and efficient, but has a tendency to fail on narrow body parts such as limbs: While high vertex weights are exclusively located on the bottom surface of a body part, the direction from which a surface supports the body is not accounted for. As a result, planes computed from PCA might be wrongly oriented.

- **Least squares or Z-distance:**

Least squares optimization or minimizing the vertical distance provide simple alternatives to PCA. In the same way as PCA, these methods minimize the distance between relevant vertices and the plane but do not actually maximize the contact area. Minimizing the vertical distance prevents the direction issue for narrow body parts, but at the same time leads to a great loss in accuracy for vertical planes such as the backrest.

- **Random sample consensus (RANSAC):**

A RANSAC based surface fitting approach was developed with the goal of directly maximizing the chosen comfort measure. In multiple iterations, three vertices of a body part are chosen at random (weighted by their importance) in order to form a plane. For each plane, the total sum of supported weights is computed and the best result is chosen.

The RANSAC based approach does not create deterministic results and is slower than the other methods. Its results however, prove to be significantly more accurate than those of other methods. In addition, the nature of the algorithm allows us to include constraints into the fitting process and therefore reduce potential errors. It is worth noting that RANSAC performs significantly worse on body parts with very low importance such as the headrest.

- **Average normals and position:**

A supporting plane can be computed by averaging the face normals of the body part and using the weighted mean position of its vertices as center. This minimalistic approach has very low accuracy but can be used for basic tasks such as error detection.

### 4.2.3 Constructing the surface model

For the optimization of the initial generic template model a combination of the evaluated surface fitting methods was chosen. The surface geometry of the generic template model (as seen in Figure 4.3) is constructed in a number of steps which are described in this section.

#### Preprocessing

Before a specific body shape model or pose can be utilized, a number of static constraints are defined. The chosen human skeleton representation consists of 21 individual bones. Each vertex of the human body shape model is mapped to one or more of those bones via skinning weights. For the generic template model, the goal is to support all body parts

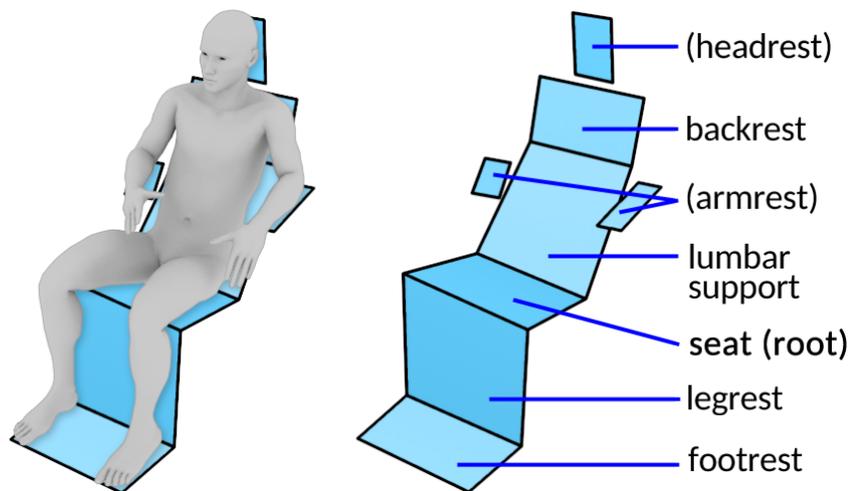


Figure 4.3: *Left:* Visual example for the geometry of the initial template model. Each face of the model lies on a plane that is fitted to its corresponding body parts. *Right:* Description of the individual faces used in the model. Armrest and headrest faces are optional and separated from the hierarchy.

of the human body shape using a much smaller number of supporting faces. Therefore, all bones of the skeleton and their corresponding body part vertices are mapped to faces of the generic template model.

In addition, we define a fixed softness threshold value for each body part, which defines how far a supporting surface is allowed to intersect with the body.

For the next steps in the template model optimization process, further input data is required. Each template model configuration is constructed for a specific pose and body shape. The chosen body shape is transformed according to the pose animation data and a corresponding importance map is computed. The importance map contains weight information for each vertex of the body shape. As the vertex density in the supplied model can vary between body parts, the importance map is normalized accordingly. From this input data and the precomputed body part mapping, the following information can be utilized in the optimization process:

- A human body shape model in a specific pose, with **position**, **normal** and **importance** weight information stored for each vertex.
- A list of 21 body parts each with a softness threshold value assigned.
- A static table of skinning weights, mapping each vertex of the body shape model to its corresponding body part(s).
- A set of planes representing the individual faces of the template model each with a list of corresponding body parts.

### Surface fitting

Utilizing the obtained data, the framework starts the surface fitting process. Each plane  $P$  of the template model is defined by its center position ( $\mathbf{c}_P$ ) and normal direction vector ( $\mathbf{n}_P$ ):  $P = (\mathbf{c}_P, \mathbf{n}_P)$

For each plane, the following steps are performed: Select all *relevant vertices* of the body mesh  $\mathbf{V}$  for the plane. We consider all vertices of the body shape relevant which match the following conditions:

- (a) The assigned skinning weights exceed a defined minimal threshold value ( $\tau^s$ ) for any of the body parts that are mapped to the current plane. In other words, we select those vertices of the body which belong to body parts that are supposed to be supported by this face.
- (b) The importance weight computed for the vertex must be higher than a defined threshold ( $\tau^r$ ). This excludes vertices with low importance from the optimization process to reduce computation time.

In addition to the relevant vertices  $\mathbf{V}_P^R$  which are used for the surface fitting process, an additional group of vertices  $\mathbf{V}_P^C$  is selected in order to account for collisions with the body shape. For collision detection, we need to consider all vertices belonging to the body part regardless of their importance weights.

$$\mathbf{V}_P^R = \{\mathbf{v} \in \mathbf{V} \mid w^s(\mathbf{v}, P) > \tau^s \wedge w^r(\mathbf{v}) > \tau^r\} \quad (4.1)$$

$$\mathbf{V}_P^C = \{\mathbf{v} \in \mathbf{V} \mid w^s(\mathbf{v}, P) > 0\} \quad (4.2)$$

- $\mathbf{V}_P^R$  is the set of relevant vertices for plane  $P$ .
- $\mathbf{V}_P^C$  is the set of collision vertices for plane  $P$ .
- $w^s(\mathbf{v}, P)$  is skinning weight value for vertex  $\mathbf{v}$  corresponding to the body parts relevant to plane  $P$ .
- $w^r(\mathbf{v})$  is the importance weight for vertex  $\mathbf{v}$ .

(2) Compute a reference direction vector from the average normals of the relevant vertices. This direction vector is necessary to get a general understanding of the orientation of the body parts and is used to prevent wrong orientations for fitted planes.

$$\bar{\mathbf{n}}_P^R = \frac{1}{\|\mathbf{V}_P^R\|} \sum_{\mathbf{n} \in \mathbf{N}_P^R} \mathbf{n} \quad (4.3)$$

- $\bar{\mathbf{n}}_P^R$  is the reference direction vector for plane  $P$ .
- $\mathbf{N}_P^R$  is the set of normal vectors corresponding to the vertices in  $\mathbf{V}_P^R$ .

(3) The selected vertices and the reference vector are further utilized to fit the template model plane to the shape of the body part. Depending on the total amount of vertex weights from the relevant set, one of two different surface fitting methods is performed. This distinction is necessary because the proposed RANSAC based approach performs poorly on smaller vertex sets and low vertex weights.

**Principal component analysis (PCA)** is used to compute an optimal plane for body parts with low importance values. Using the relevant vertex positions as input data and their importance value as weights, the PCA algorithm computes the surface normal direction of a fitted plane. The weighted average of the vertex positions is chosen as center ( $\mathbf{c}_P$ ):

$$\mathbf{c}_P = \frac{\sum_{v \in \mathbf{V}_P^r} \mathbf{v} w^r(\mathbf{v})}{\sum_{v \in \mathbf{V}_P^r} w^r(\mathbf{v})} \quad (4.4)$$

The plane normal direction ( $\mathbf{n}_P$ ) is compared with the previously computed reference vector and flipped if necessary. In order to avoid intersections with the body shape model, the plane position is shifted in opposite direction of its normal so that all intersecting vertices are within the defined softness threshold distance  $t^s$ .

$$D = \left\{ \langle \mathbf{v} - \mathbf{c}_P, \mathbf{n}_P \rangle \mid \mathbf{v} \in \mathbf{V}_P^C \right\}$$

$$\mathbf{c}_P = \mathbf{c}_P + \mathbf{n}_P \min_{\mathbf{v} \in \mathbf{V}_P^C} (D(\mathbf{v}) + t^s(\mathbf{v})) \quad (4.5)$$

- $D$  contains the distance of each collision vertex in  $V^C$  to the plane.
- $\mathbf{c}_P$  is the center position of the plane.
- $t^s(v)$  is the softness threshold for vertex  $v$ .

A **RANSAC** based approach is utilized for all regular cases where a sufficient amount of importance weights is reached. Initially, a defined number of vertex-triples are randomly sampled from the relevant set, using the vertex importance values as weights (i.e. vertices with high importance are more likely to be selected). For each selected triple of vertices, their underlying plane is computed. To avoid collision with the body, the plane is then shifted analogue to the procedure in the PCA based approach.

Following the RANSAC method, the fitted surface is evaluated for in- and outliers. For this task, we consider all relevant vertices *inliers* which are positioned within supporting distance of the computed plane. This interval is limited by a defined supporting distance in positive and the body part's softness threshold in negative direction. Rather than the number of inliers, we consider the total sum of vertex weights of all inliers as the quality value for a computed plane.

$$\mathbf{V}_P^S = \left\{ \mathbf{v} \in \mathbf{V}_P^R \mid D(\mathbf{v}) \geq t^s(\mathbf{v}) \wedge D(\mathbf{v}) \leq t^u \right\}$$

$$w_P = \sum_{\mathbf{v} \in \mathbf{V}_P^S} w^r(\mathbf{v}) \quad (4.6)$$

- $\mathbf{V}_P^S$  is the set of relevant vertices that are within the defined supporting distance  $t^u$  of a plane  $P$ .
- $w_P$  refers to the sum of supported vertex weights for a plane.

Additional constraints are required in order to prevent error cases. For this task we introduce a number of penalty factors which are utilized to reduce a plane's quality value under certain conditions.

1. *Reference vector penalty ( $p^r$ )*: The angle between the plane's surface normal and the previously computed reference vector is used to penalize planes that deviate from the body shape's orientation.

$$\delta = \left| \text{atan2} \left( \left\| \mathbf{n}^P \times \mathbf{n}^{\text{ref}} \right\|, \left\langle \mathbf{n}^P, \mathbf{n}^{\text{ref}} \right\rangle \right) \right|$$

$$p^r = 1 - \min \left( \frac{\delta}{\frac{\pi}{4}}, 1 \right) \quad (4.7)$$

$\delta$  is the angle difference between the plane normal and the reference direction.

2. *Angle difference penalty ( $p^a$ )*: This factor is used to prevent large angle differences between neighboring faces of the model. For each plane in the template model, a parent plane is defined in a simple hierarchy, which determines the order in which the template model's planes are computed.

The surface normal of a computed plane is compared with its parent's normal vector ( $\mathbf{n}^{pre}$ ) to penalize wrongly oriented intersection lines. In detail, the algorithm computes the directions of the intersection lines between both the computed plane and its parent as well as between the computed plane and an ideal direction from the parent plane center ( $\mathbf{c}^{pre}$ ) to the computed center position. The angle difference penalty is determined from the angle between these two directions ( $\mathbf{d}_1$  and  $\mathbf{d}_2$ ).

$$\mathbf{d}_2 = \mathbf{n}^P \times \mathbf{n}^{pre}$$

$$\mathbf{d}_1 = (\mathbf{c}^P - \mathbf{c}^{pre}) \times \mathbf{n}^{pre}$$

$$\alpha = \arccos \left( \frac{\langle \mathbf{d}_1, \mathbf{d}_2 \rangle}{\|\mathbf{d}_1\| \|\mathbf{d}_2\|} \right)$$

$$p^a = 1 - \frac{\min(|\alpha|, \frac{\pi}{6})}{\frac{\pi}{6}} \quad (4.8)$$

The penalty factors are applied to the computed quality value and the result is stored for each sample of vertices:

$$w_P^p = w_P(1 - \lambda^r + p^r \lambda^r)(1 - \lambda^a + p^a \lambda^a) \quad (4.9)$$

$\lambda^r$  and  $\lambda^a$  are additional variables scaling the influence of the penalty factors. After a set number of iterations, the fitted model with the best quality value is chosen.

#### 4.2.4 Mesh generation

The completed surface fitting process results in a center position and surface normal direction for each plane of the template model. In the next step, the actual mesh geometry consisting of quadrilateral faces is constructed.

The computed planes are freely positioned according to their corresponding body parts. Following the defined hierarchy, each plane is intersected with its neighbors. For the two planes on each end of the model, its neighbor's intersection line is used. The surface mesh is constructed by placing two vertices on each intersection line, with a fixed distance from the center. The mesh surface is completed by connecting the vertices of neighboring segments and placing a quadrilateral face on each plane. The resulting geometry is both connected and perfectly planar.

In order to support a person's arms, additional faces are added apart from the main hierarchy. For both forearms of the body, a supporting plane is computed via the RANSAC based algorithm (without an angle difference penalty). On this plane, the minimal rectangle supporting all relevant vertices within supporting distance is determined, forming the geometry for the armrests.

At this point it is necessary to evaluate if it is physically possible to place the armrest face and connect it to the main seating surface. For this task, the algorithm checks if there is a free path from the armrest face to the seat face. If any vertices of the body shape block this path, the armrest face is omitted from the model. Otherwise, the armrest face is connected to the seat face of the template model (i.e. the root face of the hierarchy which supports the person's hips) by inserting an additional non-planar connector face. In a slightly modified variant, the headrest face is detached from the original hierarchy to provide more accurate results. In a simple procedure, similar to how the armrests are handled, the headrest face is positioned independently and connected to the main hierarchy via an additional quad face.

#### 4.2.5 Conclusion

The initial approach for the generic template model is capable of providing acceptable support for most basic sitting poses. However, the results highlight a number of general limitations. The overall visual quality of the model leaves room for improvements. While individual faces are planar, a lack of regularity is clearly visible. The model's armrest and headrest faces arguably cause a further negative impact on aesthetics. Furthermore, the template model's geometry is connected and planar, but it does not account for physical

stability and lacks a connection to the ground. A number of generated seating surface meshes are shown in Figure 4.4

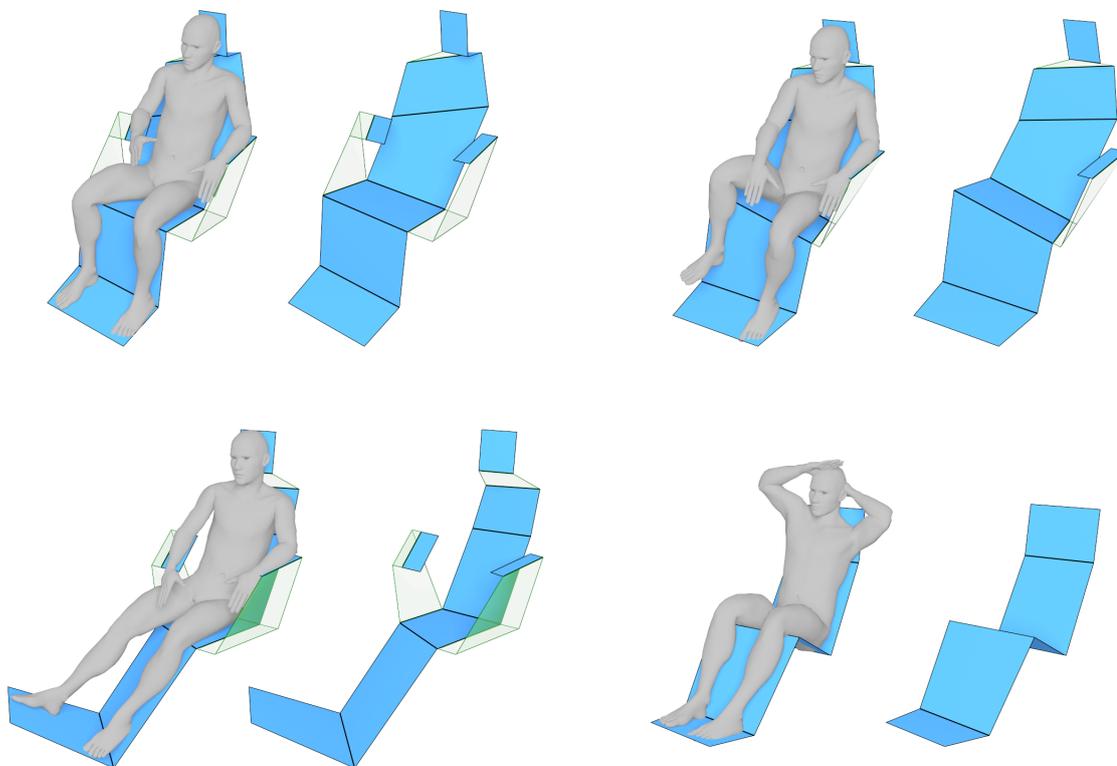


Figure 4.4: Seating surface results generated from the initial template model.

The biggest limitation of the presented model is low accuracy under various conditions. Using a single face for both of a person's legs greatly limits the potential accuracy for sitting poses with asymmetrical leg posture. In most cases, the model can only be optimized to support a single leg.

A related issue is the lack of capabilities to handle special cases as described in Section 3.4. The algorithm is not tuned to handle crossed legs or forward leaning poses. As an additional downside, the algorithm is prone to errors for poses with a bent spine posture. Overall we concluded that the model is not sufficient for our goals. Therefore a more advanced model with a higher level of detail was designed.

### 4.3 Advanced template model

Following the results of the initial template model, another attempt was made to design a more refined model capable of providing more suitable results. We propose an advanced version of the generic template model with the goal of improving the visual and functional quality of generated models as well as generally reducing limitations of the framework.

The basic concept behind the advanced model is to utilize the techniques from the initial model on a more complex mesh structure.

### 4.3.1 Model structure

The advanced approach is based on the *free faces model* used for the initial template model, described in Section 4.2.1. We expand the originally proposed design by using a regular grid of faces for the main seating surface rather than a single strip of quadrilateral faces. 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. In the context of the surface fitting algorithm we refer to the faces along the length (or height) of the body as *rows* and the faces along its width as *columns*.

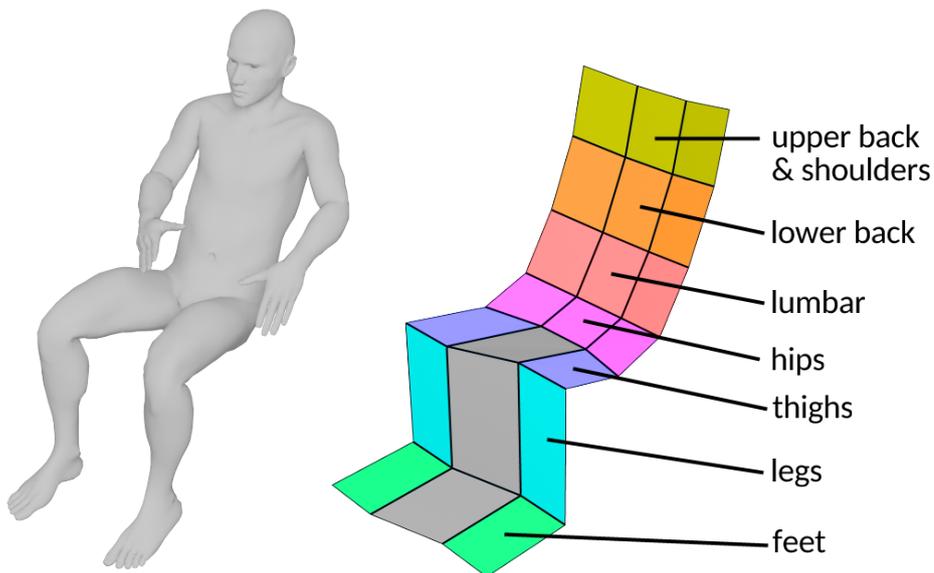


Figure 4.5: Body part mapping in the advanced template model. The rows of the model are mapped to individual body parts. Within a row, the segment in each column is mapped to a subset of the corresponding body vertices. The leg segments are mapped independently to the corresponding body parts.

Using a higher detailed mesh requires a more refined mapping between the individual faces of the model and the body parts. In the advanced template model, the legs are each mapped to an individual column of faces (see Figure 4.5). Mapping the lower body and torso becomes a more complex task as each body part (or group of body parts) has to be mapped to multiple columns of faces in each row of the model. This is performed

by splitting the set of relevant vertices of the respective body parts into three separate sets which are mapped to the respective columns of the model.

The decision on using a two-dimensional grid as seating surface was made based on different factors: Splitting a template model's face into multiple columns greatly increases the possibilities to support individual body parts. This is especially beneficial for the leg area of the seating surface where the grid structure allows both legs to be independently supported. The two-dimensional grid structure of the surface also allows the model to be more accurately fitted to the body shape model in the lower body and torso shape areas. In addition, the faces supporting the back area of a person were increased by an additional row.

A person's arms are supported by additional faces, apart from the main grid surface, that are added in a later stage of the algorithm. In the advanced template model, the headrest face was omitted. This decision was made based on the fact that the importance weights in the head and neck area of the body was generally very low for most sitting poses in the chosen input domain. Therefore it was concluded that supporting the head was insignificant for fulfilling the chosen comfort requirements.

### 4.3.2 Surface fitting and mesh generation

The general process for the fitting of the model and generation of the geometry is similar to the initially proposed variant. For each face in the grid, a plane is fitted to the shape of the respective body parts. As in the initial model, the fitting algorithm utilizes the geometry of a human body shape model transformed into specific sitting pose as well as its computed importance map, indicating which vertices are most important to support to reach optimal comfort.

With better accuracy and expressiveness of the advanced model comes the downside of greater risk for errors. As the base model consists of 21 free floating planes, additional rules are required to enforce a proper position and orientation within the grid. In order to prevent error cases and maintain the general structure of the model, hierarchical constraints are introduced to the surface fitting process. Fitted plane models are restricted depending on their neighboring rows and columns in the model. In order to properly utilize these constraints a clear order must be defined for the surface fitting process.

The general process for fitting of the template model and generation of its geometry is split into multiple passes (see Figure 4.6), each containing a cycle of three central steps: *Surface fitting*, *plane intersection* and *vertex generation*. In the first pass, the planes corresponding to the legs and feet segments as well as the central column of upper body segments are processed in separate, slightly different steps. The second pass repeats the cycle for the outer column segments of the upper body rows of the model. In this section, the individual steps of the mesh generation process are described in detail.

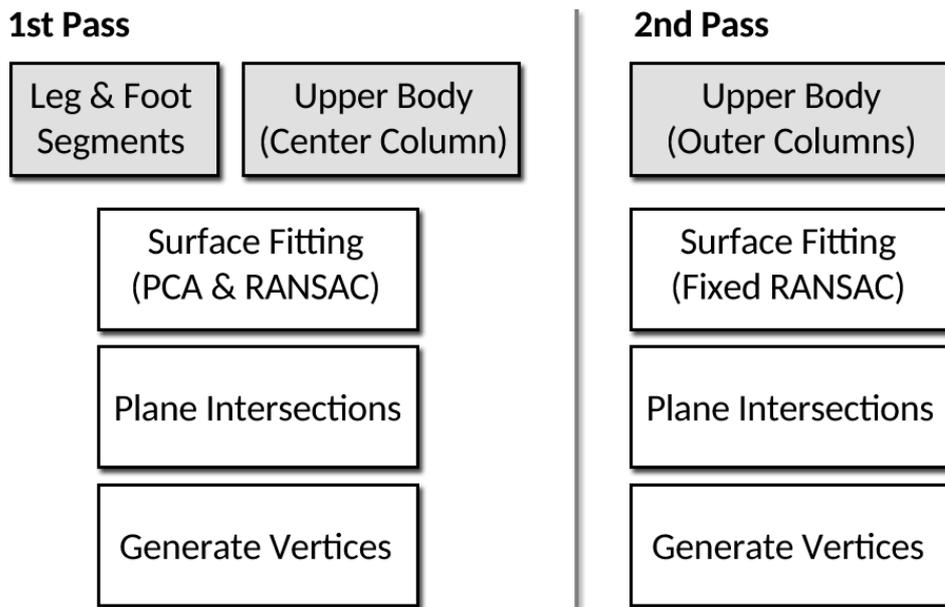


Figure 4.6: Overview of the mesh generation process of the advanced template model. Each pass consists of surface fitting, plane intersection and vertex generation steps.

### Lower body

The fitting algorithm starts by preparing the first three rows of the model corresponding to the person's legs and feet. These planes can be computed independently and require no additional constraints. The planes corresponding to the outer columns of the surface model are fitted to the person's feet, lower legs and thighs respectively. The set of relevant vertices is selected in the previously described manner from the vertices belonging to a single body part. Neighboring body parts are considered for collision detection (see Equations 4.1 and 4.2). The optimal planes are computed, using a three step fitting process on the corresponding sets of vertices.

- (1) A reference normal vector is computed from the average body surface normals (see Equation 4.3).
- (2) Principal component analysis is performed to compute a more accurate reference vector. PCA is performed as described in the surface fitting stage of the initial model (see Section 4.2.3).
- (3) An unconstrained variant of the RANSAC algorithm that was used for the initial model, is utilized to find an optimal surface model supporting the body shape (see Equation 4.6).

### Upper body

The template model's rows responsible for supporting the body shape upwards from the hip are fitted in a more complex optimization step, consisting of two separate passes.

In the first pass, the planes corresponding to the middle column of faces are computed. The second pass is responsible for the planes in the outer columns and is heavily constrained, depending on the middle column faces.

The fitting process is performed for each row, starting at the hips. Initially, the relevant vertices for the entire row are selected according to the body part mapping. Vertices belonging to neighboring rows are selected as an additional set required for collision detection.

Using the average body surface normals as reference vector, an initial plane is computed using **PCA** without further constraints. This fitted plane model serves as a general approximation of the body shape for the corresponding body parts and is used to compute reference direction vectors for the next fitting step:

(1) A *normal reference vector*  $\mathbf{n}^{ref}$  approximately represents an optimal plane normal for the row.

(3) An *upwards reference vector*  $\mathbf{d}^{up}$  is computed from the joint positions of the human skeleton itself and represents the orientation of the body shape along the spine.

(3) A *side reference vector*  $\mathbf{d}^{side}$  is orthogonal to the upwards vector and represents an approximation of an optimal direction of an intersection line with the previous plane.

The computed reference vectors are utilized as constraints in a RANSAC based fitting step to compute an optimal plane for the central column face in the current template model row.

The **RANSAC** based fitting algorithm used in this stage utilizes the same general methods to find an optimal plane. Based on the computed referenced vectors, the following constraints are utilized as penalty factors:

(1) A *reference vector penalty*  $p^r$  (see Equation 4.7) is used to penalize angle differences from the supplied reference vector.

(2) A *horizontal angle difference penalty* is in place to penalize wrong intersection lines between neighboring rows. For this task the plane's intersection direction with its parent plane is compared with the supplied side reference vector. The direction ( $\mathbf{d}^{is}$ ) and a center position ( $\mathbf{c}^{is}$ ) of the intersection line between the plane and its parent is computed via a method by John Krumm [Kru00]. The position  $\mathbf{c}^{is}$  is utilized to check whether the intersection line lies between the current plane's center and its parent plane's center. Otherwise, the plane is rejected.

$$\alpha = \arccos \left( \frac{\langle \mathbf{d}^{side}, \mathbf{d}^{is} \rangle}{\|\mathbf{d}^{side}\| \|\mathbf{d}^{is}\|} \right)$$

$$p^d = 1 - \frac{\min(|\alpha|, m^\alpha)}{m^\alpha} \quad (4.10)$$

$m^\alpha$  is a constant used to define the the maximum allowed angle difference.

The computed penalty factors and reference vectors guide the fitting algorithm into selecting planes which result in ideal intersection angles, reducing the overall error rate and improving the visual quality of the results. The penalty factors are utilized to scale the computed plane value as in previous constrained RANSAC variants.

## Plane intersections

The connected surface mesh geometry is generated by computing intersection lines between neighboring planes. The general process of placing vertices on intersection lines, utilized in the initial model is applicable for a two-dimensional grid as well. However, using a two-dimensional grid as model structure has a number of implications on the generated model:

It is impossible to preserve perfect planarity as both row and column intersections have to be taken into account. Inner vertex positions in the mesh grid are computed from the average positions of intersection lines in both row and column directions. As a result, the computed faces are distorted and planarity is lost. However, utilizing structure preserving constraints in the fitting process, we conclude that the resulting mesh geometry reaches a sufficiently high level of planarity.

In the previous fitting steps, the planes corresponding to the legs as well as the central column of planes supporting the lower and upper body shape have been computed. The remaining outer columns of faces are computed in a separate stage alongside the mesh geometry generation process. Vertex positions in the surface geometry are dependent on plane intersections. We define *column intersection lines* as the intersection lines between neighboring columns within a row (i.e. the intersection lines are vertically aligned) and *row intersection lines* as the intersection lines between adjacent row segments within a column (i.e. horizontally aligned).

**Row intersection lines** are computed during the fitting process by intersecting a fitted segment with its predecessor. This is performed for the inner column of the body rows as well as the outer columns of the leg segments.

**Column intersection lines** determine the width of the column segments in a row. For the leg segments, no actual intersections are computed, as the center column of faces is not fitted and has no supporting function. Instead, suitable intersection lines are estimated from the directions of the corresponding body parts, computed from their joint orientation in the underlying skeleton. These intersection lines on both sides of the limb are estimated by shifting this direction outwards by a distance matching the width of the body part's shape.

In order to compute the mesh grid geometry in the upper rows of the model, the row intersection lines are computed in a similar indirect manner: Rather than fitting the corresponding planes, we first estimate suitable positions for the inner vertices and edges of the row. This is performed by estimating a center position and orientation for the current row's corresponding body parts. The inner vertex positions are placed by shifting the center position outwards on the row's fitted plane by a width computed from a

defined fraction of the body part's total width.

$$\begin{aligned}
D &= \left\{ \left\langle \mathbf{v} - \mathbf{c}^P, \mathbf{d}^{side} \right\rangle \mid \mathbf{v} \in \mathbf{V}_P^R \right\} \\
w &= \max_{d \in D} d - \min_{d \in D} d \\
\mathbf{c}^R &= \mathbf{c}^P + \mathbf{d}^{side} \left( \min_{d \in D} d + \frac{w}{2} \right) \\
\mathbf{c}_1^{ic} &= \mathbf{c}^P + \mathbf{d}^{side} w s^w \\
\mathbf{c}_2^{ic} &= \mathbf{c}^P - \mathbf{d}^{side} w s^w \\
\mathbf{d}_1^{ic} &= \mathbf{d}_2^{ic} = \mathbf{d}^{up}
\end{aligned}$$

- $\mathbf{d}^{side}$  and  $\mathbf{d}^{up}$  are directions computed from the skeleton joint orientation corresponding to the upwards and sideways direction of the body part. Both vectors lie on the current plane defined by position  $\mathbf{c}^P$  and normal  $\mathbf{n}^P$ .
- $D$  contains the distance of each relevant vertex in direction of  $\mathbf{d}^{side}$  from the plane center position.
- $w$  is the total spanning width of the relevant vertices.
- $\mathbf{c}^R$  is the center position of the relevant vertices on the plane.
- $\mathbf{c}_1^{ic}$  and  $\mathbf{c}_2^{ic}$  are the positions of the column intersection lines. The corresponding directions ( $\mathbf{d}_1^{ic}$  and  $\mathbf{d}_2^{ic}$ ) are equal to the upwards direction vector  $\mathbf{d}^{up}$  for the body part.
- $s^w$  is a constant value specifying the width of the inner column segment as fraction of the total width.

These operations are visualized in Figure 4.7:

(1) The process starts with a single row's plane position  $\mathbf{c}^P$  and normal  $\mathbf{n}^P$  and its corresponding relevant body vertices  $\mathbf{V}_P^R$ . The direction  $\mathbf{d}^{body}$  represents the body part's orientation and is computed from the skeleton's joint positions.

(2) The body vertices  $\mathbf{V}_P^R$  and the direction  $\mathbf{d}^{body}$  are projected onto the plane, reducing this problem to 2D.  $\mathbf{d}^{up}$  is the projected and normalized body part orientation.  $\mathbf{d}^{side}$  is the orthogonal direction on the plane.  $w$  is the total spanning width of the projected body vertices, in direction  $\mathbf{d}^{side}$ . The row's position  $\mathbf{c}^R$  is shifted to the center.

(3) The column intersection line positions  $\mathbf{c}_1^{ic}$  and  $\mathbf{c}_2^{ic}$  are determined by shifting the center position by a fixed fraction ( $s^w$ ) of the total width in direction  $\mathbf{d}^{side}$ .

(4) The vertex positions for the inner column segments are computed by intersecting

adjacent intersection lines. Each row intersection line for an inner column segment shares an individual plane with each adjacent column intersection line and is therefore guaranteed to intersect. The intersection points are computed by solving the corresponding line equations:

$$\begin{aligned} l^{ic}(s) &= \mathbf{c}^{ic} + \mathbf{d}^{ic}s \\ l^{ir}(t) &= \mathbf{c}^{ir} + \mathbf{d}^{ir}t \\ \mathbf{c}^{ic} + \mathbf{d}^{ic}s &= \mathbf{c}^{ir} + \mathbf{d}^{ir}t \end{aligned}$$

We compute each inner vertex position  $\mathbf{v}$  as the average of the two intersection points computed for its adjacent row and column intersection lines.

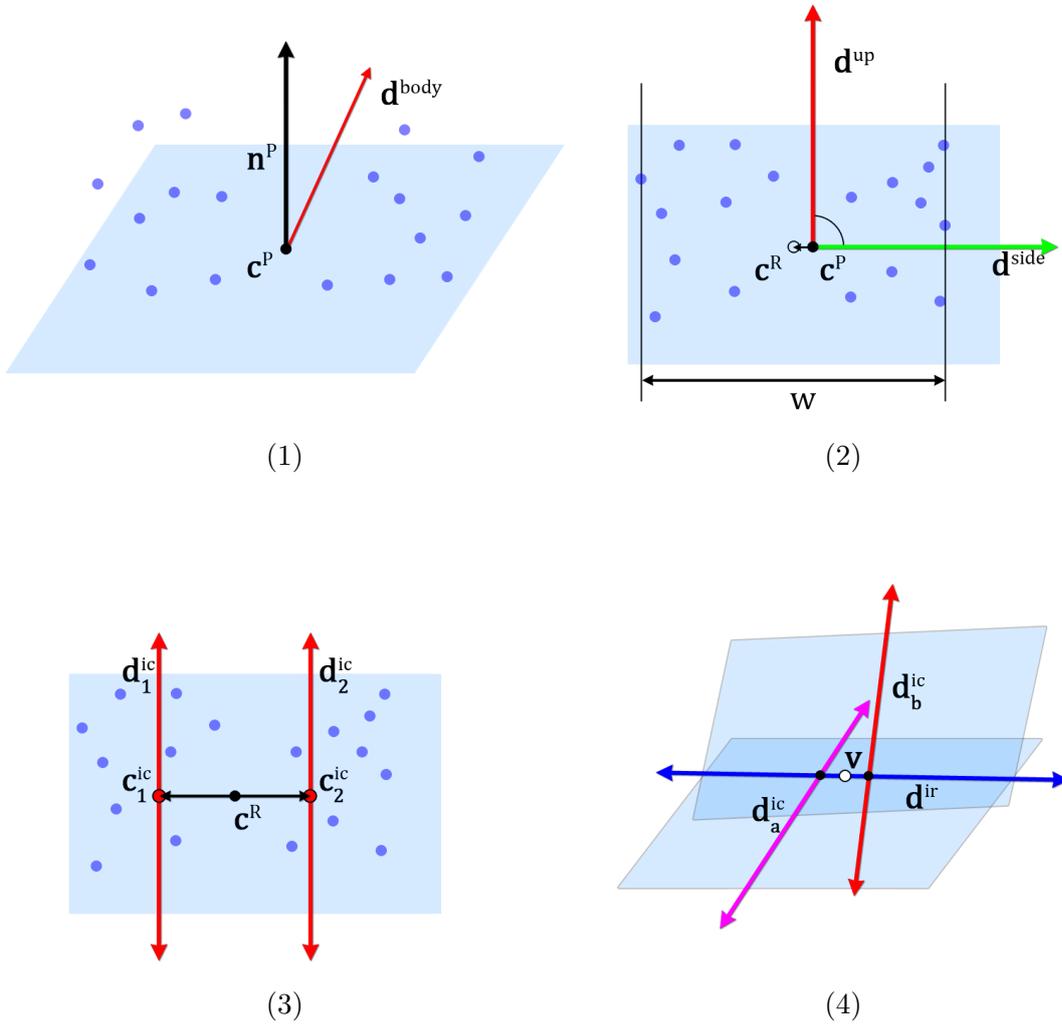


Figure 4.7: Plane intersection and vertex generation process for upper body inner segment vertices.

The remaining outer column segments are determined via a surface fitting algorithm while utilizing the geometry of the inner column segments as hard constraints. For each row from the hips upwards, the planes representing the two outer segments are computed by performing a constrained variant of the RANSAC fitting method.

The **fixed RANSAC** approach is used to find a suitable plane model for an outer segment. As we assume the segment's inner vertices are already in optimal position, they are locked in place and therefore must be part of the fitted model. As a result, the algorithm samples a single vertex rather than groups of three. As the number of possible combinations is significantly smaller in this variant, there is no need for random sampling and it is instead possible to evaluate all candidate planes. In addition to the locked vertices, we utilize the previously described *reference vector* and (*horizontal*) *intersection angle* penalty factors.

As an additional error prevention mechanism, the horizontal intersection line is compared with its preceding intersection line from the previously computed vertically adjacent segment. In this step, the two intersection lines are evaluated up to a defined length. A detected intersection of two lines within the chosen distance, implies that the preceding segment's quadrilateral geometry degenerates into a triangle, in which case the candidate plane is rejected.

$$\begin{aligned}
 \mathbf{p}_1 &= \mathbf{v}_1 + \mathbf{d}^{is}w \\
 \mathbf{p}_0 &= \mathbf{v}_0 + \mathbf{d}_0^{is}w \\
 \mathbf{d}_{01} &= (\mathbf{p}_1 - \mathbf{p}_0) - (\mathbf{n}^{pre} \langle \mathbf{p}_1 - \mathbf{p}_0, \mathbf{n}^{pre} \rangle) \\
 \langle \mathbf{d}_{01}, \mathbf{v}_1 - \mathbf{v}_0 \rangle &\geq 0
 \end{aligned} \tag{4.11}$$

- $\mathbf{d}^{is}$  is the current plane's horizontal intersection line direction with its previous segment.
- $w$  is a constant value for the border width of the model.
- $\mathbf{v}_1$  is the fixed vertex of the current segment that is adjacent to its previous plane.
- $\mathbf{v}_0$  is the predecessor to  $\mathbf{v}_1$ . (The vertex adjacent to  $\mathbf{v}_1$  that is part of the previous plane).
- $\mathbf{d}_0^{is}$  is the horizontal intersection line of the previous segment with its predecessor, located at  $\mathbf{v}_0$ .

The two vertices (corresponding to the two inner vertices of the previous segment) are shifted along the corresponding intersection lines by a set distance. The angle of the vectors between the original points as well as the shifted points are compared and evaluated. The condition described in Equation 4.11 must be fulfilled for a given candidate plane.

The chosen length of this evaluation directly corresponds to the desired width of the outer column segments of the model. Increasing the width further restricts the possible outcomes of this fitting step.

The highly constrained nature of this algorithm eliminates potential error cases, but also greatly restricts the expressiveness of the fitting process. As a result, the resulting outer faces of the model are usually less effective at supporting important areas of the body. If the algorithm cannot find a suitable plane model, the framework iteratively increases the supporting distance threshold and repeats the process.

In order to complete the surface generation process the remaining vertex positions are computed. The surface mesh's outer vertices are positioned by expanding the intersection lines in the outer segments by a fixed width.

The remaining inner vertices are computed by averaging the intersection points between adjacent vertical and horizontal intersection lines. The resulting geometry is a connected 3x7 grid of non-planar quadrilateral faces fitted to the given body shape.

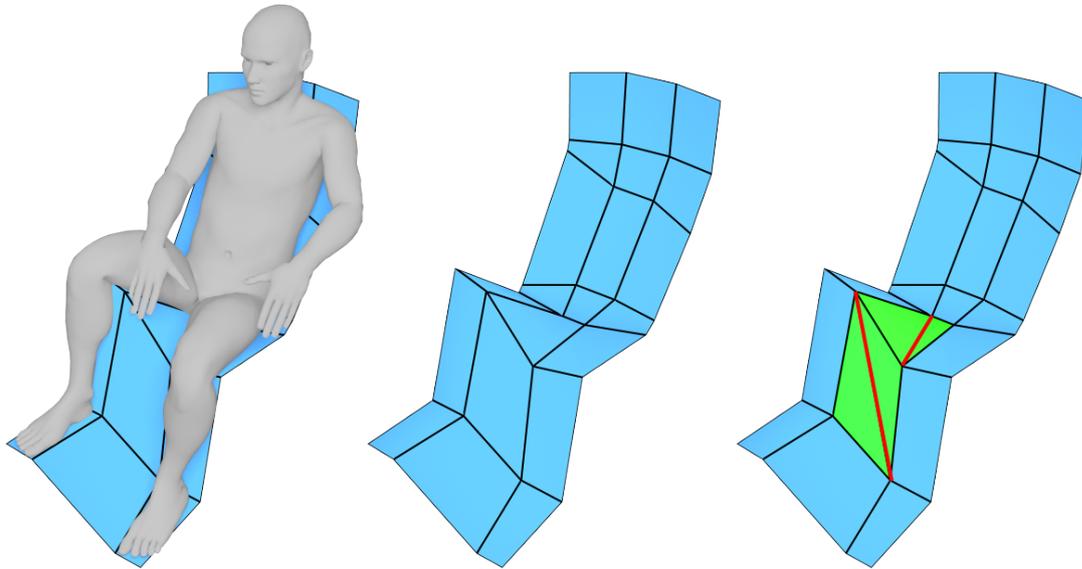


Figure 4.8: Intermediate seating surface results generated from the advanced template model. The main seating surface supporting the legs and body of a person is represented as a 7x3 grid of quadrilateral faces. The resulting surface is non-planar (highlighted in the right image).

### 4.3.3 Special case handling

The proposed algorithm is capable to provide suitable solutions for basic sitting poses. However, certain orientations of body parts in sitting poses can cause errors and require additional measures. The framework must be able to detect these special cases and adjust

the surface fitting process accordingly.

From evaluating intermediate results of this framework we 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.

### **Forward leaning poses**

When a person is leaning forward when sitting, his or her back cannot be actively supported by a chair's backrest. Within the proposed framework, this fact is made apparent by the importance map that is computed for such a pose, as there are very low weights in the lumbar and back area of the body surface. As a result, there is very low confidence in fitted planes for the respective parts of the template model. This greatly raises the risk for errors, as inaccuracies in the fitting step can be propagated further.

In the proposed system we detect and process these special cases in a very simple manner. In the surface fitting process for a row of the template model, the vertex weights of its corresponding body parts are evaluated. If the sum of weights does not exceed a certain threshold, we consider the row obsolete and omit it from the model. As a result all succeeding rows in the hierarchy are removed as well. Effectively this reduces the resulting chair model to a stool for poses where a backrest is not needed.

### **Crossed legs**

Dealing with crossed legs in sitting poses is a more complex task. The problem at hand is that the approach of individually fitting faces to a person's legs fails in situations where one leg is on top of the other. One leg effectively supports the other, leaving no space for a supporting surface. As a result, applying the regular fitting algorithm would result in intersections between the template model's faces and the body shape.

In the proposed framework these error cases are detected after the initial surface fitting process for the leg areas is completed. 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 fit a person's legs individually and therefore initiate the error handling process.

The fitting process is repeated for the affected row with a few adjustments: The relevant vertices in the respective leg parts are combined to a single set for both limbs. Likewise, a single plane is fitted to the combined set of body vertices. Instead of fitting the three column faces of the row separately, a single plane is used for the entire row. Depending on the leg positioning, this process is performed for the rows corresponding to the feet, lower legs or thighs. The proposed measures are sufficient to eliminate errors for poses with crossed legs, with a slight loss of precision and visual quality as a tradeoff.

#### **4.3.4 Refinement stage**

In the previous stages of the algorithm, a connected surface mesh geometry has been generated matching the body shape of a person in a given pose. In the following stages,

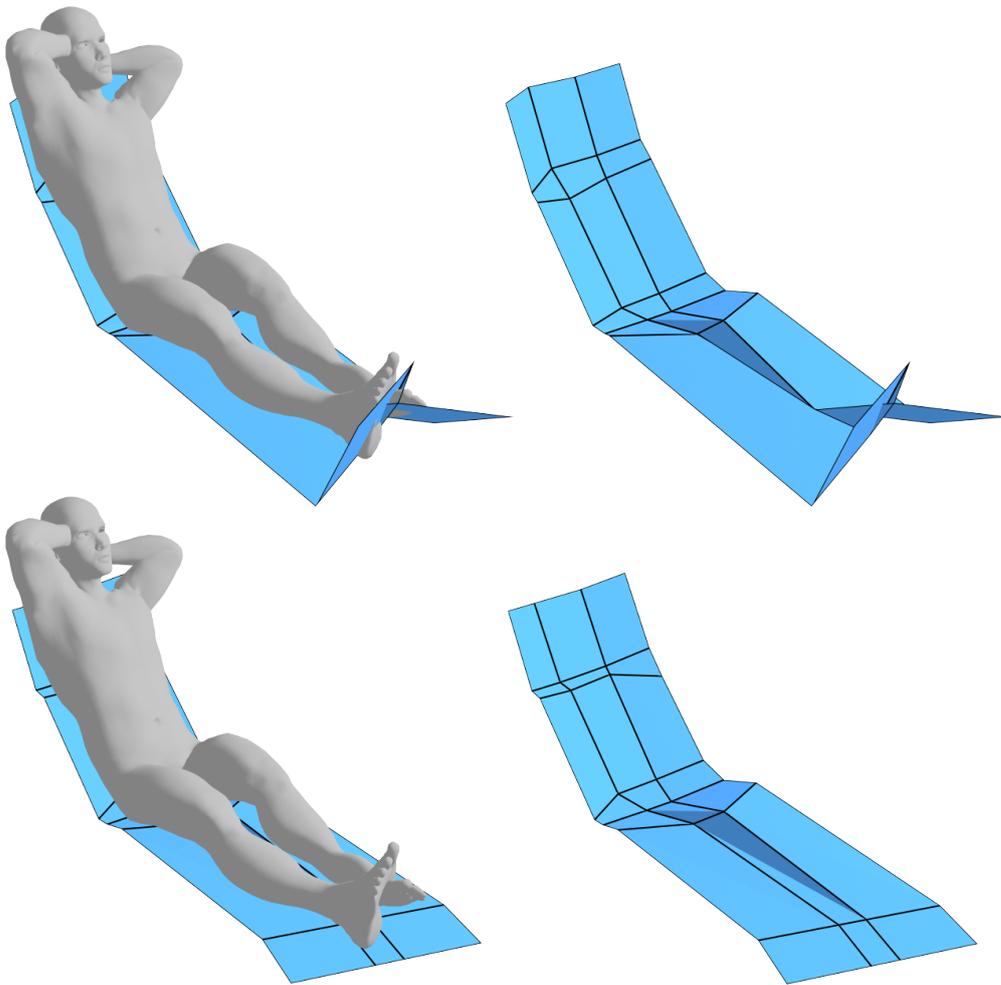


Figure 4.9: Error handling on poses with crossed legs. *Top*: Surface model generated without error handling showing intersections. *Bottom*: Corrected surface mesh.

the surface model's mesh structure is expanded, with the goal of further improving the provided level of comfort as well as visual quality, by creating a smoother and more regular surface mesh. The general approach is to expand the main seating surface by additional faces at the sides of the model, while maintaining a regular grid structure. At this point, two additional tasks are performed.

First, additional faces are fitted to the person's arms, providing additional support. As we add geometry to the model, an additional column of faces is utilized to serve as armrests, in a way that preserves the overall regular grid structure. This includes evaluation whether or not an armrest can be placed as part of the mesh grid without intersecting the main seating surface or the body.

Second, the mesh geometry is expanded by additional faces, in order to connect the mesh to the ground and provide a (hypothetically) stable base. This would generally enable the generated model to be fabricated from solid material as shown in the research by Leimer et al. [LBRM18]. In the proposed framework the following steps are performed to complete these tasks:

(1) 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.

(2) In order to create the geometry we do not utilize intersections between neighboring planes, but instead find the minimal spanning rectangle on the computed plane that contains all relevant vertices that lie within supporting distance of the plane. This rectangular face represents the ideal armrest regarding comfort and visual quality.

(3) The next step is to integrate the armrest faces into the mesh grid structure of the model. This includes evaluating whether or not it is possible to include armrests for the given pose and the corresponding seating surface. Therefore the candidate armrest surface is evaluated for intersections with the body mesh to avoid errors (for instance, when a person's forearm rests on his or her body).

In addition, the armrest rectangle's inner vertex positions needs to span a minimal horizontal distance to the vertices of the main seating surface to allow for an integration into the mesh grid. If all requirements are met, an additional columns of faces is formed, vertically adjacent to both sides of the armrest face. In order to bridge the gap between the armrest column and the main surface, another column of quadrilateral faces is inserted. This process is performed on both sides of the seating surface resulting in a connected surface grid supporting the body and the arms of a person in a given pose.

In addition to the armrest faces and their corresponding rows, the surface 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.

As a result of the surface fitting process, it is possible that the surface geometry contains overhanging faces. As a result, 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 sides of the model. This process rearranges the corresponding vertices so that each outer column face is convex.

Figure 4.10 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, further processing is required.

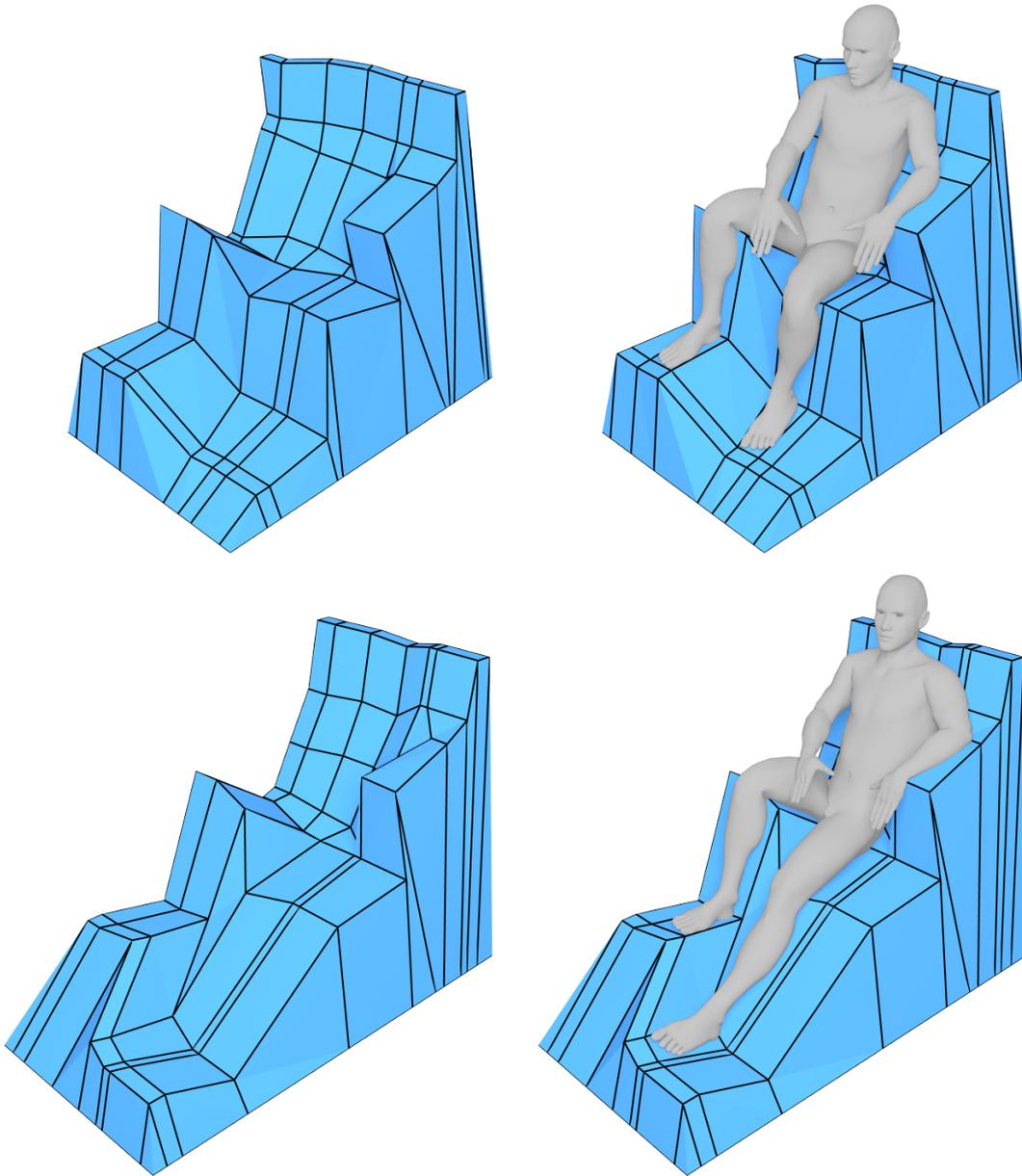


Figure 4.10: Advanced model seating surface results after the refinement stage.



# Optimization

After completing the previous stages of the algorithm, the framework generates seating surface models, fitted to a specific body shape in a given pose. At this point the functional requirements are satisfied to an adequate degree. (The functional quality of the results including comfort measures is evaluated in detail in chapter 7.) In contrast, the visual quality of the results at this stage of the algorithm is lacking in regards to the defined criteria for aesthetics. Therefore, an additional refinement step on the generated seating surface meshes is applied. For this task we perform a non-linear local optimization process.

## 5.1 Goals

In order to clarify the requirements for the optimization process we recall the overall goals for seating surface models:

1. **Comfort:** The surface model should support the important areas of the body shape in order to maximize comfort.
2. **Visual quality:** The surface model should be visually appealing. We assume that a smooth and regular surface, with high planarity is desired.
3. **Functional constraints:** The surface model has to fulfill all additional functional constraints including connectivity and physical plausibility (no intersections, stable base).

The intermediate results from the previous mesh generation steps fulfill the functional constraints and comfort requirements to a high degree. The central goal of the optimization process is to improve the aesthetics of the surface model. For this task we expect improvements in the following aspects of the model:

- **Smoothness:** The finished surface model should have an overall smooth shape. Sharp angles and edges should be avoided.
- **Regularity:** In an optimal case the quadrilateral faces of the model should be rectangular. In addition, the faces within rows and columns of the model should be of equal size and the vertices should be evenly distributed along the model.
- **Planarity:** The advanced model's mesh generation process produces non-planar quad-faces for the seating surface. To maximize visual quality, planar faces are desired.
- **Refined border:** The additional segments of quadrilateral faces that were inserted along the border of the seating surface show a very low degree of visual quality in most cases. Additional refinement is expected in this area.

As visual quality is second to functionality, the optimization process must be designed in a way that preserves the seating surface's level of comfort and its functional constraints. Therefore, any adjustments made to the surface mesh should have a limited effect on the functional quality.

Within the framework we define the functional quality as the sum of body vertex weights that are supported by the model. This was the primary goal of the initial fitting process. For this optimization stage, we therefore aim to ensure that the adjusted model stays in close proximity to the previously fitted model.

## 5.2 Optimization problem overview

The first step for an optimization based approach is to set up the optimization problem and find appropriate methods to solve it. In the following sections, the essential elements of the optimization problem are described in detail, starting with the **variables**.

Next, **qualitative measures** are formulated representing the desired goals for functional and visual quality of the surface model.

The next core element of the optimization problem is the **objective function** that is systematically evaluated to compute the optimal function value within the (constrained) input domain. To formulate the objective function, data and smoothing terms are defined corresponding to the quality measures.

To maintain the structure of the model and fulfill all functional requirements, certain limitations must be imposed on the model's vertex positions. For this task, a number of **constraints** are defined.

## 5.3 Variables

The **variables** used in the optimization problem are the vertex positions of the surface model. The basic idea is to move vertex positions in a way to improve the overall quality

of the model. The expected result of the optimization process is an optimally adjusted set of vertex position representing the finalized seating surface model. Each vertex position consists of its corresponding X, Y and Z components.

## 5.4 Quality measures

In an optimization problem, the objective function's return value represents a quality measure computed for the corresponding set of input values. For this task we aim to find suitable terms to rate a certain configuration of vertex positions according to their visual and functional quality. For the proposed optimization process, we formulate an *energy minimization problem*. This means, rather than computing a configuration's quality, we determine an error value. A configuration where the computed error value is locally lowest is considered the optimal solution of the problem.

In order to compute qualitative measures for the proposed visual and functional aspects of the model, we evaluated a number of metrics to determine which terms are most suitable for the energy function. For the individual terms of the objective function we distinguish between the **data term** responsible for the functional requirements and the **smoothing term** representing the visual quality. For each term a combination of quality measures is used:

### Functional quality measures

The data term's primary objective is to maximize the seating surface's provided comfort. We aim to achieve this by preserving the geometry from the original surface fitting process. Therefore, the functional quality measures used for the optimization process should directly relate to the original model's configuration. For this task, we consider the computed planes from the surface fitting process as well as the vertices of the generated surface model.

**(1) Vertex distance:** We use the distance between a sample configuration's vertex positions to its original positions as error metric. While this metric ensures a high level of comfort, it heavily penalizes all adjustments to the model. As a result, vertex distances have to be weighted independently, which is covered in Section 5.4.1.

**(2) Optimal plane distance:** In addition, we consider the optimal planes that were computed in the surface fitting step: We utilize the distance of each vertex from its corresponding plane as error metric. In contrast to the vertex distance, this error metric allows a vertex to move along its corresponding plane, leaving more room for smoothing operations. As optimal planes were only computed for the central part of the surface mesh, this error metric only affects a part of the model.

### Visual quality measures

To reach overall improvements on visual quality and aesthetics of the model we aim to improve smoothness, planarity and regularity of the surface mesh as well as refine its borders. While we described the factors for visual quality in this framework, there is no numerical ranking between these visual aspects. Therefore, visual quality cannot be seen as an objective quality measure. Choosing visual error metrics is a subjective decision. In order to find the most suitable (subjective) quality measures we evaluated various error metrics with varying parameters. In conclusion, the following visual error metrics were chosen for the smoothing term of the objective function:

**(1) Laplacian smoothing:** To smooth out edges in surface mesh, a Laplacian smoothing based approach has proven effective. The mesh smoothing procedure is performed on each vertex of the mesh by computing a new smoothed position from the average position of its neighbors. As our seating surface model is a regular grid of vertices, this operation can easily be applied to improve the smoothness and regularity of the mesh. The corresponding error metric used in the objective function is the distance of a vertex position to its smoothed position.

**(2) Regular face angles:** To find quality measures related to the planarity and regularity of the surface mesh, we evaluate the model's face angles. One condition for the planarity of a quadrilateral face is that its interior angles sum up to 360 degrees, providing a simple way to rate the planarity of an object. For each quadrilateral face in the mesh grid, the sum of its interior angles is computed. We penalize any differences between a face's sum of angles from an optimal 360 degrees.

**(3) Maximized angles:** From subjective evaluation of visual results it was observed that maximizing the models interior angles leads to an overall improvement of regularity of the shape. While interior angles of 90 degrees are required to form a rectangular face, enforcing perfect rectangles for the mesh grid would flatten out the entire surface. Instead we consider ideal interior angles of 180 degrees for each quadrilateral face, effectively maximizing the interior angles. Combined with other metrics this has proven to be an effective way of increasing regularity without flattening the surface.

#### 5.4.1 Importance weights

Applying the same operations on each vertex of the surface model during the optimization process leads to suboptimal results. Specific error metrics can be suitable for specific parts of the mesh and very unreliable for others. Using custom weights for each of the chosen error metrics can greatly improve the overall quality of the mesh. Therefore, we aim to find suitable weights for the surface model's geometry, effectively mapping each error metric to specific parts of the mesh.

**(1) Vertex importance weights:** For the energy function's data term we aim to define weights, relative to the underlying geometry's importance for optimal comfort.

In other words, faces which support a large area of the human body shape are of high importance for the comfort that the surface model is able to provide. Faces whose primary purpose is to connect supporting faces in the model, are insignificant for the overall comfort. A minimal importance value is assigned to those faces, to preserve the basic structure of the surface model.

The overall comfort provided by the seating surface is almost exclusively depending on the primary faces of the model that were computed in the surface fitting process. Therefore we assign high importance weights to vertices that belong to the primary faces of the model. As the inner vertices of the primary seating surface are adjacent to a larger number of primary faces, their importance values are increased further. These vertex importance weights are utilized for the corresponding error metrics in the data term. As a result, movement for vertices with high importance is heavily penalized. This weighting scheme preserves a high level of comfort while allowing non-supporting vertices to be more freely adjusted in order to improve visual quality.

**(2) Border weights:** In order to provide a suitable weighting for the face angle error metrics, additional weights are assigned to the faces of the mesh grid. Judging from the subjective visual quality of results, it has proven effective to apply higher weights to the outermost border of faces when computing the *regular face angles* error metric.

The four corner faces of the mesh grid have zero weights assigned, as it is impossible for them to be planar without distorting the rest of the model.

For the Laplacian error metric, vertices on the border of the surfaces are weighted significantly lower, to avoid large distortions in the model.

### Error metric factors

In addition to assigning individual weights to the geometry, specific factors are applied to each term of the objective function. We utilize an adjustable global data and smoothing factor, allowing users to adjust the optimization goals between functionality and visual quality.

Furthermore, the magnitude of each individual error metric in the data and smoothing terms is affected by an individual factor that can be adjusted by advanced users. This allows the framework to be fine-tuned for specific applications and requirements.

## 5.5 Constraints

The overall goal of the optimization process is to maximize both comfort and visual quality. These soft constraints are represented by the data and smoothing terms of the objective function. However, a number of functional and visual requirements are hard constraints and therefore cannot be solved by energy minimization. In order to fulfill all hard requirements, additional optimization constraints are utilized.

The resulting seating surface model is required to have a **rectangular base**. All

outermost vertices of the surface mesh grid must therefore lie on the corresponding edges of a rectangle, whereas the surface's corner vertex positions must be equal to the rectangle's corner positions. In addition, each surface mesh vertex on a respective edge of the border rectangle, must lie between its predecessor and successor. In other words, the order of vertices must not change. This requirement is necessary to maintain the structure of the mesh grid as border vertices have very low importance weights (in context of the data term of the objective function).

A solution to this problem can be formulated as a number of linear (in)equalities:

- For each edge of the bordering rectangle, each vertex position must be greater than it's predecessor's in the respective direction of the edge.
- Each border vertex position in orthogonal direction to the edge must be equal to all other vertices on this particular edge.
- All border vertices must have equal height, which must lie between a defined minimal and maximal height. The maximal height corresponds to the lowest point of the remaining surface model.

An additional non-linear constraint is utilized in order to prevent degenerations of the geometry during the optimization process. A **minimal edge length** is defined, restricting the allowed distance between neighboring vertices in the mesh grid. This prevents edges from disappearing and reducing their adjacent quadrilateral faces to triangles.

## 5.6 Energy function

In this section, the objective function and its terms are described in detail. The optimization goal is to find a set of vertex positions which minimizes the proposed energy function.

Each vertex position  $\mathbf{v} \in \mathbf{V}$  consists of separate x, y and z position values. The total number of surface mesh vertices  $n_V = \|\mathbf{V}\|$  can vary between configurations. The energy function is defined as

$$E = (S_L + S_A)\lambda_S + (D_V + D_P)\lambda_D \quad (5.1)$$

where

- $(S_L + S_A)$  is the smoothing term and  $\lambda_S$  is the corresponding global smoothing factor.
- $(D_V + D_P)$  represents the data term with  $\lambda_D$  as global data factor.

### 5.6.1 Smoothing term

The smoothing term ( $S_L + S_A$ ) consists of separate terms for error metrics corresponding to the Laplacian ( $S_L$ ) and angle based ( $S_A$ ) smoothing.

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

$$S_L = \sum_{i=1}^{n^V} \left\| \mathbf{v}_i - \frac{\sum_{j \in N_1(i)} \mathbf{v}_j w_j}{\sum_{j \in N_1(i)} w_j} \right\|^2 \lambda_S^l \quad (5.2)$$

- $N_1(i)$  is the 1-ring neighborhood of  $\mathbf{v}_i$  in the mesh grid.
- $w_j$  is a corresponding importance weight for the neighboring vertex  $\mathbf{v}_j$ .
- $\lambda_S^l$  is the Laplacian error metric factor.

The **angle based smoothing** term consists of two separate error metrics as described in Section 5.4. The angle based error metrics are accumulated over the faces of the surface model.

$$S_A = \sum_{j=1}^{n^F} \left( \left( \sum_{i \in F_j} \alpha_i - 2\pi \right)^2 w_j^1 \lambda_S^{A1} + \left( \sum_{i \in F_j} (\alpha_i - \pi)^2 \right) w_j^2 \lambda_S^{A2} \right) \quad (5.3)$$

- $n^F$  is the total number of faces in the surface mesh.
- Each face  $F_j$  is a set of 4 vertices.
- $\alpha_i$  is the  $i$ th interior angle of the face.
- $\lambda_S^{A1}$  and  $\lambda_S^{A2}$  refer to the angle based error metric factors.
- $w_j^1$  and  $w_j^2$  are importance weights assigned to face  $j$  for the corresponding error metrics.

The first part of the above equation represents the *regular face angles error*. For each (quadrilateral) face in the mesh grid, the sum of its four interior angles and its squared difference to  $2\pi$  is computed. In addition, a per-face importance factor as well as a global scaling factor are applied.

The second term computes the *maximized angles error metric*. For each face, the sum of squared differences between its angles and  $\pi$  is computed, multiplied with a importance factor as well as a global scaling factor.

### 5.6.2 Data term

The objective function’s data term is composed of a vertex distance ( $D_V$ ) and plane distance ( $D_P$ ) term.

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

$$D_V = \lambda_D^V \sum_{i=1}^{n^v} \|\mathbf{v}_i - \tilde{\mathbf{v}}_i\|^2 w_i \quad (5.4)$$

- $\mathbf{v}_i$  is the position of the  $i$ th vertex in the current configuration.
- $\tilde{\mathbf{v}}_i$  is the original position of the same vertex.
- $w_i$  is a corresponding per-vertex importance weight.
- $\lambda_D^V$  is the vertex distance term’s global scaling factor.

The squared vertex distances are scaled by their importance for comfort (as described in Section 5.4.1). In addition, a global scaling factor is applied.

The **plane distance term** 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. The same vertex importance weights are applied to each plane distance value. The plane distance error metric is the sum of plane distances weighted by a global scaling factor.

$$D_P = \sum_{j=1}^{n^F} \left( \sum_{i \in F_j} \langle \mathbf{v}_i - \mathbf{c}_j^P, \mathbf{n}_j^P \rangle^2 w_i \right) \lambda_D^P \quad (5.5)$$

- $\mathbf{v}_i$  is the position of the  $i$ th vertex of the  $j$ th face in the current configuration.
- $\mathbf{c}_j^P$  and  $\mathbf{n}_j^P$  are the center position and surface normal of the supporting plane that was computed for the  $j$ th face.
- $w_i$  is the per-vertex importance weight assigned to the corresponding vertex of the  $j$ th face.
- $\lambda_D^P$  is the plane distance term’s global scaling factor.

## 5.7 Gradient

In the previous sections, the objective function, variables and constraints have been defined, providing a sufficient description of the optimization problem, allowing a suitable non-linear solver to find a minimum.

In order to improve the efficiency of the solving process, we evaluate the gradient of the energy function. Unless provided, numerical solvers estimate gradients via finite differences. Supplying the derivatives of the objective function can therefore increase the speed and accuracy of the energy minimization process.

For this task, we need to provide the computations corresponding to the derivatives for each variable (i.e. the x, y and z coordinates of the surface mesh vertices) of the objective function. For each configuration the gradient is evaluated.

### 5.7.1 Energy function derivatives

In order to evaluate the gradient, the derivatives for the energy function need to be defined. For this task, each term of the function can be derived separately.

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

$$S_L = \sum_i^{n^v} \sum_{k=x,y,z} \left( v_{ik} - \frac{\sum_{j \in N_1(i)} v_{jk} w_j}{\sum_{j \in N_1(i)} w_j} \right)^2 \lambda_S^l \quad (5.6)$$

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.

$$\frac{\partial S_L}{\partial x} = 2\lambda_S^l \left( x - \frac{\sum_{j \in N_1(i)} x_j w_j}{\sum_{j \in N_1(i)} w_j} \right) \quad (5.7)$$

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:

$$\frac{\partial S_L}{\partial x_{n1}} = \frac{-2\lambda_S^l w_{n1} \left( x - \frac{\sum_{j \in N_1(i)} x_j w_j}{\sum_{j \in N_1(i)} w_j} \right)}{\sum_{j \in N_1(i)} w_j} \quad (5.8)$$

### 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):

$$\frac{\partial \alpha_{abc}}{\partial a_x} = -\frac{\frac{\mathbf{e}_{bc,x}}{m_{bc}m_{ab}} - \frac{\mathbf{e}_{ab,x}S_{abc}}{m_{bc}m_{ab}^3}}{\sqrt{1 - \langle \mathbf{t}_{ab}, \mathbf{t}_{bc} \rangle}} \quad (5.9)$$

$$\frac{\partial \alpha_{dab}}{\partial a_x} = -\frac{\frac{\mathbf{e}_{da} + \mathbf{e}_{ab}}{m_{ab}m_{bc}} - \frac{\mathbf{e}_{da,x}S_{dab}}{m_{ab}^3m_{da}} + \frac{\mathbf{e}_{ab,x}S_{dab}}{m_{ab}m_{da}^3}}{\sqrt{1 - \langle \mathbf{t}_{da}, \mathbf{t}_{ab} \rangle}} \quad (5.10)$$

$$\frac{\partial \alpha_{cda}}{\partial a_x} = -\frac{\frac{-\mathbf{e}_{cd,x}}{m_{cd}m_{da}} - \frac{\mathbf{e}_{da,x}S_{cda}}{m_{cd}m_{da}^3}}{\sqrt{1 - \langle \mathbf{t}_{cd}, \mathbf{t}_{da} \rangle}} \quad (5.11)$$

$$S_{abc} = \sum_{k=x,y,z} (b_k - a_k)(b_k - c_k)$$

- $a, b, c$  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

$$D_V = \lambda_D^V \sum_{i=1}^{n^V} \left( (v_{ix} - \tilde{v}_{ix})^2 + (v_{iy} - \tilde{v}_{iy})^2 + (v_{iz} - \tilde{v}_{iz})^2 \right) w_i \quad (5.12)$$

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

$$\frac{\partial D_V}{\partial v_{ix}} = 2\lambda_D^V w_i (v_{ix} - \tilde{v}_{ix}) \quad (5.13)$$

#### 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:

$$\frac{\partial D_P}{\partial v_{ix}} = 2\lambda_D^P w_j n_x^{P_j} \langle v_i - c^{P_j}, n^{P_j} \rangle \quad (5.14)$$

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  $n^V * 3$ , where each value corresponds to the sum of the error metric gradient values for an individual variable.

## 5.8 Optimization results

Minimizing the proposed energy function results in a set of optimized vertex positions for the seating surface mesh. The optimization process is able to increase the visual quality of the results by a large margin, while preserving a high level of functionality. A visual comparison to unoptimized surface models is shown in Figure 5.1.

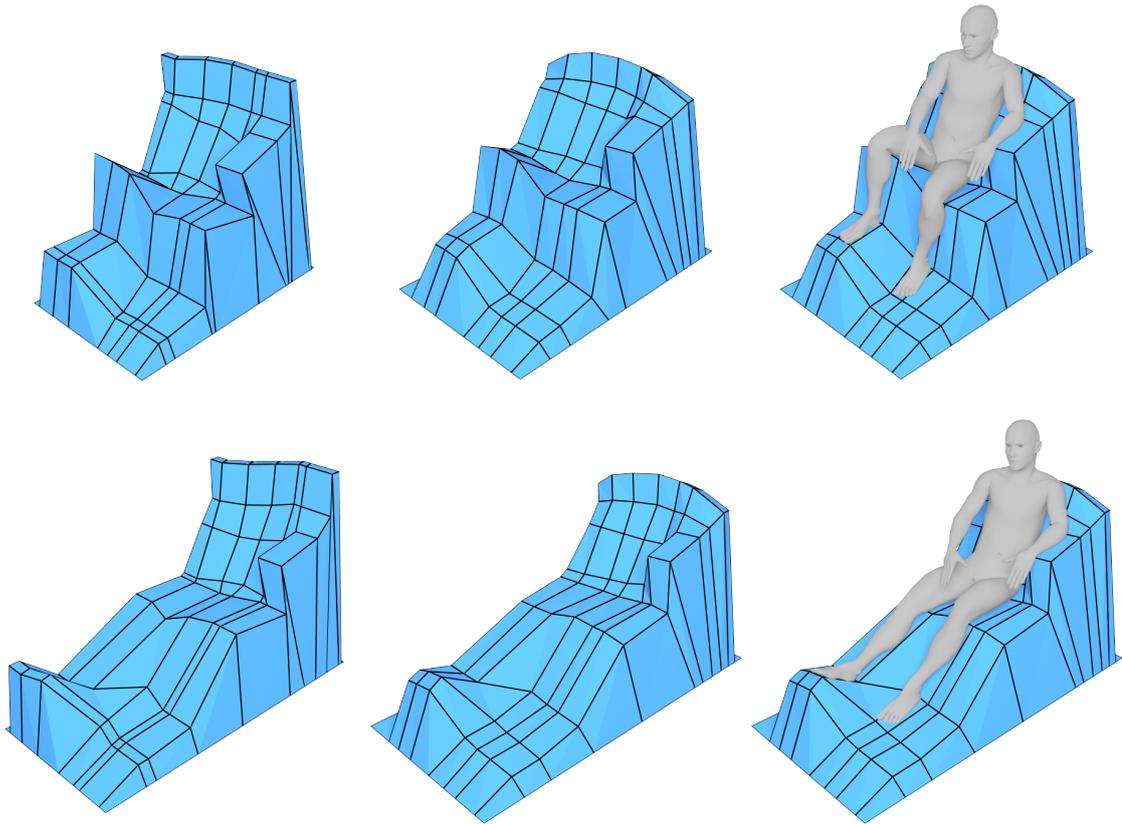


Figure 5.1: Finalized seating surface results after the optimization process. *Left*: Seating surface before optimization. *Center/Right*: Results after optimization.

# Implementation

In the previous chapters, the ideas and concepts behind the main contributions of this thesis were presented. We explained the methods and techniques utilized in the algorithms of the proposed framework.

In this chapter, the implementation of the algorithms is presented. We describe the structure of the proposed system and its components, as well as its environment and the used technologies.

## 6.1 Overview

A system overview of the proposed framework was given in Figure 3.1. In the context of the implementation of the system, we expand and alter this notation to provide a complete overview of the developed framework and technologies that were utilized (Figure 6.1).

- The largest part of the framework, including all core algorithms that were presented, is implemented in **MATLAB**.  
The provided algorithms by Leimer et al. that are utilized within the framework are implemented in MATLAB and are directly usable in our system.
- The **Rhinoceros 3D** CAD application was utilized for two reasons.
  - (1) It is used to run the Grasshopper visual programming environment, which is utilized as user interface of the framework.
  - (2) The input body shape 3D model and the generated seating surface results are directly usable in Rhino to render or to place within a scene.
- A **Grasshopper** component is implemented in **C#** and executed within the Rhino application. This plugin serves as GUI of the framework, handling the required

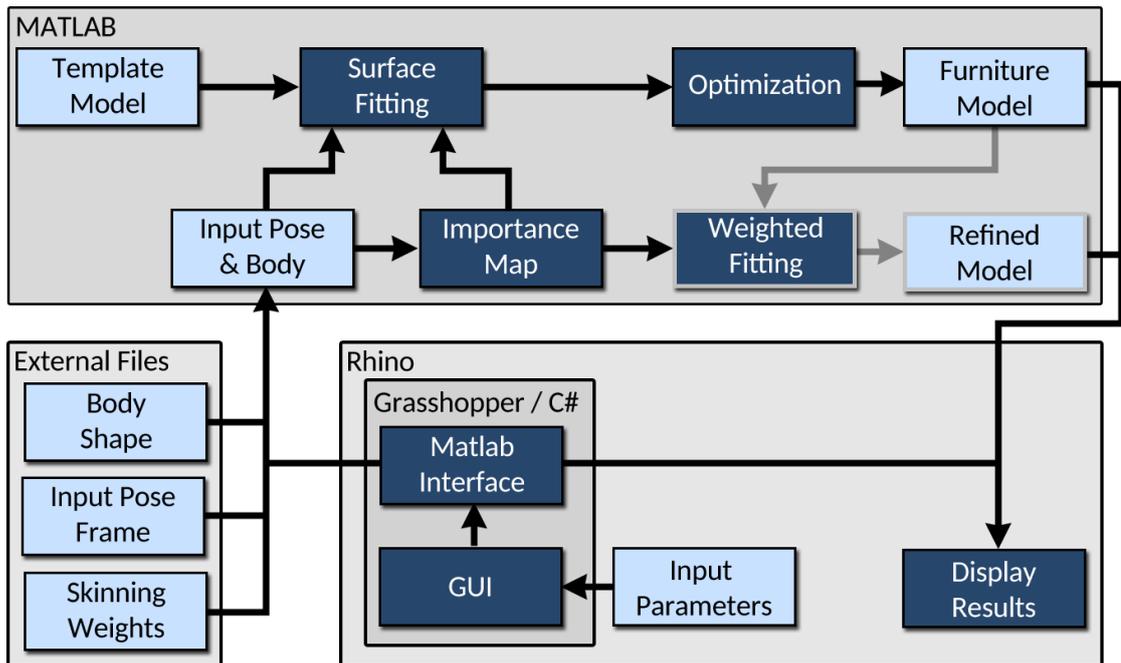


Figure 6.1: An extended overview of the proposed system: The input parameters specified in the Grasshopper GUI specify which body and pose file to load. The C# Grasshopper plugin serves as the MATLAB interface and controls the data exchange. The seating surface results are displayed in Rhino 3D.

input and output parameters. Furthermore it serves as interface to the MATLAB environment, transmitting the required input data and executing the corresponding MATLAB functions.

## 6.2 MATLAB

In this section the MATLAB implementation of the template model, including the surface fitting and optimization steps is explained in detail. As the functionality of the algorithms has been described in detail in earlier chapters, the focus lies on implementation details, such as the used data structures and functions.

In this section, we first provide an overview of the methods used in the MATLAB implementation of the proposed algorithms. Furthermore, certain data structures as well as the implementation of specific methods is described in detail.

### 6.2.1 Algorithm overview

The MATLAB implementation consists of separate functions responsible for the individual tasks. The first stage of the algorithm is to read the input files, generate and fit the generic template model and perform optimization. This process is started in the method

`generateBaseModel`, which takes a pose frame number and a prefix specifying the body shape model as input (See Algorithm 6.1).

---

**Algorithm 6.1:** Input data loading and preprocessing (`generateBaseModel()`)

---

```

1 readFrame(); // Load body shape model and pose animation data.
  Compute importance map for pose and body.
2 getBodyPartSoftness(); // Defines body part specific softness
  distance threshold values.
3 getBodyPartIndices(); // Defines mapping between template model
  segments and body parts.
4 getBodyPartDirections() // Computes orientation of body parts
  from the body skeleton.
5 checkRequiredRows(); // Reduces the number of rows for forward
  leaning poses.
```

---

### Surface fitting

The surface fitting step starts with the first 3 rows of the model containing the faces supporting the legs and feet of the body shape (See Algorithm 6.2).

---

**Algorithm 6.2:** Surface fitting for legs and feet (`generateBaseModel()`)

---

```

1 for rows 1 to 3 do
2   | select relevant and collision vertices;
3   | compute reference normal vector (PCA);
4   | perform RANSAC fitting;
5   | intersect with previous segment; // row intersection lines
6 end
```

---

The next step revolves around fitting the upper body rows of the model (See Algorithm 6.3).

The leg and feet segments are evaluated to detect special cases where the legs are crossed. If the distance between the left and right segments of a row is too small, the legs are treated as single body part and supported by a combined surface (See Algorithm 6.4).

The vertices for the inner columns of upper body segments are computed and locked in place, before a fixed RANSAC algorithm is performed to fit the outer segments of the corresponding row (See Algorithm 6.5).

---

**Algorithm 6.3:** surface fitting for (inner) upper body segments. (generateBaseModel())

---

```
1 for rows 4 to n do
2   select relevant and collision vertices;
3   compute reference normal vector (PCA);
4   compute reference side and upwards direction vectors;
5   perform (constrained) RANSAC fitting;
6   intersect with previous segment; // horizontal intersection lines
7   estimate column intersection lines; // from body part directions
8 end
```

---

---

**Algorithm 6.4:** Crossed legs special case detection. (generateBaseModel())

---

```
1 for rows 2 to 1 do
2   compute distance between left and right leg (or feet) segments;
3   if distance < 0.15 then
4     select relevant and collision vertices for both legs;
5     compute reference normal vector (PCA);
6     compute reference side and upwards direction vectors;
7     perform (constrained) RANSAC fitting;
8     intersect with adjacent rows;
9     adjust vertex positions;
10  end
11 end
```

---

---

**Algorithm 6.5:** Mesh generation for upper body segments. (generateBaseModel())

---

```
1 for rows 4 to n do
2   select intersection lines for inner segment;
3   compute inner segment vertex positions; // from line intersections
4   for outer column segments do
5     compute reference normal vector (PCA);
6     compute reference side and upwards direction vectors;
7     perform fixed RANSAC fitting;
8     intersect with previous segment; // row intersection lines
9   end
10 end
```

---

In the last step of the initial mesh generation, the outer vertices of the model are placed according to the intersection lines and a fixed border width. (See Algorithm 6.6).

---

**Algorithm 6.6:** Outer vertex generation (`generateBaseModel()`)

---

```

1 for rows 1 to n do
2   select horizontal intersection lines for outer segment;
3   expand by border width constant; // 0.13
4   check for triangle intersections;
5   place outer vertices;
6 end

```

---

The resulting seating surface is then refined by adding additional border segments. In this step, the faces for the armrests of the model are fitted and integrated into the mesh grid (See Algorithm 6.7).

---

**Algorithm 6.7:** Refinement & post processing (`generateBaseModel()`)

---

```

1 Align surface mesh and body shape vertices to axis;
2 for each armrest segment do
3   // surface fitting for left and right forearms.
4   select relevant and collision vertices;
5   compute reference normal vector (PCA);
6   perform (unconstrained) RANSAC fitting;
7   getArmrestFace(); // compute minimal spanning rectangle for
   supported arm vertices.
8 end
9 ExtrudeBorders(); // Expand the surface mesh grid by an
   additional row (or column) of faces in each direction.
   For each valid armrest an additional column is
   inserted.
10 AddGroundBorderFaces(); // Add an additional border of segments
   connecting the mesh to the ground plane.
11 RearrangeBorderVerts(); // Perform linear optimization to
   rearrange the outer vertices to avoid irregular shapes.

```

---

**Optimization**

The second major step of the algorithm is the optimization process which is performed in the method `optimizeControlMesh()`. At this stage, the non-linear optimization is prepared and initiated (See Algorithm 6.8).

The objective function's implementations completes two major tasks: (1) Given a con-

---

**Algorithm 6.8:** Optimization process (optimizeControlMesh())

---

```
1 precompute vertex and face weights;
2 prepare vertex and face indices; // Matrix based operations in the
  objective function require corresponding indices for
  specific vertices and faces.
3 define linear constraints. // Prepare matrices for linear equations
4 prepare variables and initial values. // Initial values = current
  vertex positions
5 define non-linear objective function; // See pfun()
6 define non-linear constraint function; // See pcon()
7 define optimization parameters; start non-linear optimization (fmincon);
```

---

figuration of vertex positions, the error metric values corresponding to the data and smoothing terms of the energy function are computed. (2) The corresponding gradient values are accumulated for each variable. (See Algorithms 6.9 and 6.10).

As last part of the optimization algorithm, a non-linear constraint function is defined in order to enforce the minimum edge length constraint (See Algorithm 6.11).

### 6.2.2 Input data

The first task of the main MATLAB function is to load the required data. The provided input parameters specify which body shape model and animation frame to use and the corresponding files are loaded from the disk. At this stage the skinning weights are utilized to transform the vertex data according to the pose. From the provided code by Leimer et al., the corresponding importance map is computed.

This leaves us with the following data, available for the algorithm:

- Positions, normals and importance weights for each vertex of the body shape. This information is stored in separate  $n * 3$  (positions, normals) or  $n * 1$  (importance weights) matrices.
- A skinning weights matrix mapping each vertex to its body parts ( $21 * n$ )

In addition, the static information regarding body part mapping and softness thresholds are defined. For each row in the template model (7 rows are used in the proposed version) its corresponding body parts are defined.

### 6.2.3 Template model data structure

The main part of the template model is represented as a  $3 * 7$  grid of individual segments. The model is based on a fixed hierarchy of 7 vertical segments, referred to as *rows*. Each

**Algorithm 6.9:** Objective function (pfun()) Part 1 of 2

---

```

// Laplacian distance smoothing error metric.
1 compute weights for Laplacian error metric; // Reduced weights for
  border vertices.
2 compute smoothed vertex positions; // Weighted average from
  neighboring vertex positions.
3 compute Laplacian distance error metric; // squared distance between
  original and smoothed position.
4 accumulate gradient values for Laplacian error metric;
  // Angle based smoothing error metrics.
5 compute edge vectors for mesh grid; // Subtract neighboring vertices.
6 compute normalised edge direction vectors;
7 compute face interior angles from edge directions;
8 sum up angles for each face;
9 compute regular face angles error metric; // squared difference of
  angle sum from 2*pi
10 compute maximized angles error metric; // squared difference of
  angle from pi
11 compute weights for angle based smoothing; // Reduced weights for
  border faces.
12 for i = 1 to 4 do
    // compute values for angles at position i in a face.
13     compute angle based gradient values; // For left, center and
        right vertex position of an angle.
14 end
15 accumulate angle based gradient values for all variables;

```

---

row consists of 3 horizontal segments (i.e. *columns*) and has a specific role and name (e.g. "seat"). Body parts are either assigned to the entire row (e.g. backrest) or a single segment (e.g. legs).

For each row (or row segment), the direction of its corresponding body part is computed from the skeleton itself, using the positions and orientations of its joints.

The planes computed in the surface fitting process are stored as positions and normal directions in two separate  $3 \times 7 \times 3$  matrices.

Row and column intersection lines between adjacent planes are stored as pairs of positions and direction vectors in a grid arrangement in separate matrices.

For intermediate results of the generated surface model, the vertex positions are stored in a two-dimensional grid. As the surface model is expanded by additional segments, this corresponding vertex position matrix is enlarged.

As the surface mesh is a regular grid, its edges and (quadrilateral) faces as well as their corresponding vertex indices are implicitly defined.

---

**Algorithm 6.10:** Objective function (pfun()) Part 2 of 2

---

```
// Vertex distance error metric.  
1 compute squared distance between vertex positions and original positions.  
2 compute vertex distance weights. // From support values.  
3 compute vertex distance gradient values. // Plane distance error  
  metric.  
4 compute distances between vertices and original fitted planes. compute plane  
  distance weights. // From support values.  
5 compute plane distance gradient values. // Objective function result.  
6 define factors for individual error metrics;  
7 compute weighted sum of error metric values; // Apply factors and  
  weights.  
  // Objective function gradient.  
8 accumulate gradient values for each variable; // Apply factors and  
  weights.
```

---

---

**Algorithm 6.11:** Non-linear constraint function (pcon())

---

```
1 prepare horizontal vertex position indices;  
2 compute horizontal edges;  
3 prepare vertical vertex position indices;  
4 compute horizontal edges;  
5 compute edge distances;  
6 compute constraint values; // Minimum edge length - edge lengths
```

---

### 6.2.4 Surface fitting

The surface fitting process in this framework utilizes principal component analysis as well as a RANSAC based fitting algorithm.

#### PCA

To perform weighted principal component analysis, the MATLAB function `pca` is utilized. This method uses the singular value decomposition (SVD) algorithm to return the principal component coefficients for a supplied matrix.

In the proposed algorithm, we supply an  $n \times 3$  matrix containing the relevant vertices for a given segment in the template model as well as a vector of size  $n$  consisting of the corresponding importance weights. The vertex importance weights are used as observation weights for the `pca` method.

The returned coefficients for the principal components define vectors forming the basis and normal vector of the plane.

## RANSAC

The RANSAC based fitting algorithm is a custom implementation based on the random sample consensus principle. Throughout the framework, different variants of this algorithm are used. The constrained variant, that is used for general surface fitting is described in Algorithm 6.12.

### 6.2.5 Optimization

The optimization process in this framework is implemented using the MATLAB function `fmincon` which provides a nonlinear programming solver. The function solves an optimization problem specified by finding the minimum of  $f(x)$  such that  $c(x) \leq 0 \wedge Aeq * x = beq \wedge x \leq ub$

- $f(x)$  is the energy function (`fcon`) computing the sum of error metrics for a given configuration (see Algorithm 6.9).
- $c(x)$  is the nonlinear constraint function. In the proposed framework this function (`pcon`) is responsible for the minimum edge length constraint (see Algorithm 6.11).
- $Aeq$  and  $beq$  specify a system of linear equalities that are used to keep the border vertex positions in a rectangular shape.
- $ub$  is the upper bound of variable values that is utilized to keep the border vertex positions in place at the bottom of the shape.

The `fmincon` function is run in multiple iterations using the *interior point* algorithm. The number of 50000 iterations was chosen as a tradeoff between computation time and quality. We include the gradient evaluation in the objective function, in order to speed up the optimization process. Experiments have shown 25 to 35% faster processing time for the optimization step when utilizing the gradient evaluation.

### 6.2.6 Results

The optimization process returns a set of vertex positions corresponding to the optimal configuration. As the mesh is a regular grid and the number of vertices does not change, it is a trivial task to rearrange the geometry structure for use in different applications.

In order to utilize the weighted fitting method by Leimer et al. the surface model is prepared in mesh structure consisting of (1) a list of vertex positions, (2) a list of quadrilateral faces containing the indices of the corresponding vertices and (3) a list of vertex colors that are used to provide additional information for the weighted fitting algorithm (Specific color information locks a vertex in place for further optimization steps. This is used to keep the border vertices in place on the ground).

---

**Algorithm 6.12:** Constrained RANSAC surface fitting algorithm (optimizePlane\_ransac())

---

```
1 define penalty factor constants; // Maximum angles and distances,
  Scalars
2 get  $k * 3$  weighted random samples from  $n$  indices; //  $n$  = number of
  relevant vertices,  $k$  = number of iterations, Vertex
  importance weights are used.
3 initiate  $PW_{max}$  as infinite; // Current maximum sum of weights for
  a plane.
4 initiate  $pP$  and  $pN$  as zero; // Position and normal of the best
  plane.
5 select next 3 sample vertices;
6 compute plane from vertices;
7 check plane normal direction; // flip normal if distance from
  reference vector exceeds a threshold.
8 plane center = average vertex position projected onto the plane;
9 shift plane to avoid collision; // consider distance of collision
  vertices to the plane.
  // Penalty Factors:
10 compute reference vector penalty factor;
11 if Vertical parent/neighbor exists then
12   |  $dir_{ref}$  = supplied reference intersection direction; // Side reference
  | vector from body orientation.
13   |  $dir_{IS}$  = intersection line between plane and (vertical) parent;
14   | validate plane intersection position; // Check if the plane
  | intersection position lies between the current plane
  | and its neighbor.
15   | if invalid position then
16   | | reject plane;
17   | end
18   | compute angle between  $dir_{IS}$  and  $dir_{ref}$ ;
19   | compute horizontal angle difference penalty factor;
20 end
  // Evaluate current plane
21  $SU$  = supporting distance; // Vertices within this distance are
  considered supported.
22 compute distances of relevant vertices to current plane;
23  $SV$  = vertices where the distance  $\leq SU$ ; // Supported vertices.
24 sum importance weights from supported vertices.
25  $PW$  = supported weights * penalty factors.
26 if  $PW \geq PW_{max}$  then
27   |  $PW_{max} = PW$ ;
28   | set  $pP$  and  $pN$  to current plane position and normal;
29 end
30 return  $pP$  and  $pN$ ;
```

---

The `weightedFitting` method is called using the model structure, the body shape information as well as parameters for iterations, smoothing and subdivision steps.

## 6.3 Rhino 3D

In this section, the components of the framework that were implemented in Rhino 3D and its Grasshopper environment are described. This includes the user interface and the 3D model output.

### 6.3.1 User Interface

The user interface for this framework utilizes the Grasshopper visual programming interface that is run within the Rhino 3D application. The GUI is based on a custom Grasshopper component that is placed within a visual program, providing interfaces for the required input and output parameters. A screenshot of the component within the Grasshopper environment is shown in Figure 6.2.

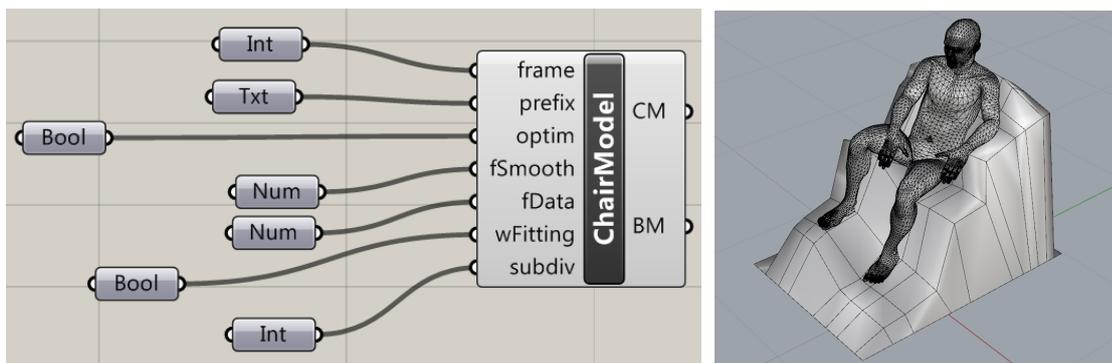


Figure 6.2: *Left:* The Grasshopper component serving as the user interface. (Input parameters on the left, output on the right.) *Right:* Rhino 3D scene of the generated mesh.

The Grasshopper component shows a number of input and output parameters that can be connected to other nodes within the environment. The nodes on the left hand side are used to specify the input parameters of the framework. The following parameters are available:

- *frame*: The number of the pose animation frame from the provided files.
- *prefix*: A filename prefix specifying which body shape to use.
- *optim*: A boolean value specifying whether to use optimization or not.
- *fSmooth*: The smoothing term factor used in the optimization process.

- *fData*: The data term factor used in the optimization process.
- *wFitting*: A boolean value specifying whether to use weighted fitting or not.
- *subdiv*: The number of subdivision steps used for the weighted fitting algorithm.

The component's output parameters are 3D meshes of the body shape and the generated seating surface. These results can be rendered in Rhino 3D (See Figure 6.2) or further utilized in different Grasshopper components.

### 6.3.2 Grasshopper plugin

The presented Grasshopper component was implemented as a plugin in C# using MS Visual Studio. This C# program contains the necessary definitions for the input and output parameters of the component and provides the MATLAB interface, required to exchange data and call the required functions.

Each time a parameter is changed, the plugin's `SolveInstance` function is executed, which performs several tasks:

- (1) The program fetches the input parameter values from its Grasshopper data interface. At this stage, it is checked whether the parameters have changed in order to prevent redundant computations.
- (2) A connection to the MATLAB automation sever is established. The input parameters are sent to the MATLAB workspace. The MATLAB functions for mesh generation, optimization and weighted fitting are executed. The results remain in the MATLAB workspace.
- (3) The resulting geometry data is retrieved from the MATLAB workspace. The vertex and face information from MATLAB is converted to a mesh data structure that is compatible with Rhino 3D. The resulting meshes for the body shape and the seating surface are then provided as output of the Grasshopper component.

# Results

In this chapter, we present the results created with the proposed framework. This includes both a visual display of generated furniture models as well as a detailed evaluation of results in regards to visual and functional quality measures as well as performance. In addition, we utilize created seating surfaces as control meshes in the algorithm by Leimer et al. and evaluate and compare its results.

As the proposed framework produces 3D models of seating furniture, the most suitable way to present its results is to provide visual examples. In addition, we compute a pair of quality measure values for each seating surface result, corresponding to the functional and visual goals. In order to display the range of the framework's ability, we provide a variety of visual examples using varying poses, parameter values and body types.

## 7.1 Overview

In the following sections, we first define the **quality measures** and error metrics used to evaluate the framework's results.

Next, we provide a selection of **visual results** highlighting the algorithm's performance for general results, special cases and body shape variations. In addition, we observe how created surface models perform as control mesh in the weighted fitting algorithm and compare its results with generic shapes.

The last part of this chapter is a larger scale **evaluation** of different sitting poses using a variety of options and parameters as well as different body shapes. We evaluate the algorithm's performance in regards to the presented quality measures as well as the run time of individual steps of the algorithm.

### 7.1.1 Evaluation criteria

In previous chapters of this thesis we used the term quality measures in context of surface fitting and optimization. We described both functional goals, referring to the surface’s ability to support a body in a given pose, as well as visual goals, describing the aesthetics of the furniture model. We revisit the computations utilized in the surface fitting and optimization steps in order to define measures usable to evaluate the functional and visual quality of the framework’s results.

#### Total support value

The total support value is based on the notation for functional quality used in the surface fitting process. This quality value is computed as the total sum of weights of all vertices that are within supporting distance of the seating surface.

For each quadrilateral face of the seating surface mesh, the body vertices that are within supporting distance of its plane (in orthogonal direction to the plane) and within the bounds of the quadrilateral face (parallel to the plane) are considered supported. The total support value (TS) is the sum of all supported weights. Body vertices that would be supported by multiple faces of the model are counted only once.

#### Visual error value

As visual quality measure, we define a visual error value (VE) which specifies a model’s deviation from an (visually) ideal shape in regards to the visual error metrics defined in the optimization stage of the framework.

For this task, we utilize the computations described in Equations 5.2 and 5.3 to compute the Laplacian and angle based error metrics for a given surface model using the same vertex and face weights as in the optimization process.

The visual error value is therefore defined as  $VE = S_L + S_A$  which is the smoothing term of the energy function (excluding its corresponding global scale factor).

## 7.2 Visual results

In this section, visual seating surface results along with corresponding quality measures are presented. We display a number of test result sets for a range of different test scenarios.

### 7.2.1 General results

Figures 7.1 and 7.2 show a number of visual examples of generated results for different sitting poses. Each pose was fitted with two sets of parameters. The images on the left hand side show our preferred default setup ( $Smoothing = 1$ ,  $Data = 5$ ), whereas the right hand side shows results with a bigger focus on visual quality for the respective poses

---

(*Smoothing* = 1, *Data* = 2). The values (TS, VE) shown with the images correspond to the previously described quality measures *total support value* and *visual error value*.

### 7.2.2 Special cases

A selection of special case results is displayed in Figure 7.3. These poses have either forward leaning posture or crossed legs and are not suitable for the default implementation of the algorithm. Therefore a number of extra steps are utilized (as described in Section 4.3.3) to create satisfying results.

### 7.2.3 Body shape variations

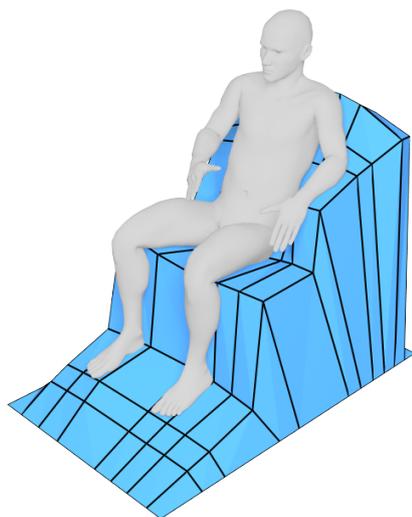
The framework was designed to be able to support a variety of human body types. Figure 7.4 shows a number of visual examples for generated results with alternative body shapes. The used female body shape model has a larger number of vertices and therefore a larger number of total importance weights, which affects the computed quality metrics. The resulting total support values are therefore not directly comparable with those computed for different body shapes.

### 7.2.4 Weighted fitting

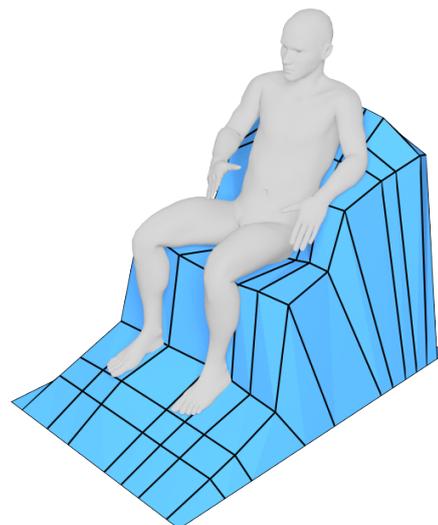
One of the goals for this framework was to produce suitable shapes suitable as control meshes for the weighted fitting algorithm by Leimer et al. In Figure 7.5 a few examples of generated surfaces after application of the weighted fitting algorithm are presented. No visual error metric has been computed for these results as its computations are no longer applicable for the subdivided surface.

To show the benefits of using a generated control mesh, we compare the results of the weighted fitting algorithm using generic rectangular control meshes as well as our generated surfaces (See Figure 7.6). The weighted fitting algorithm is configurable to create either smoother or more precise results. The optimal parameters vary between input shapes. For the presented results, we have chosen parameters that fit all input poses. For the generic results, the smoothness parameter is increased as the algorithm is otherwise prone to creating self-intersecting surfaces.

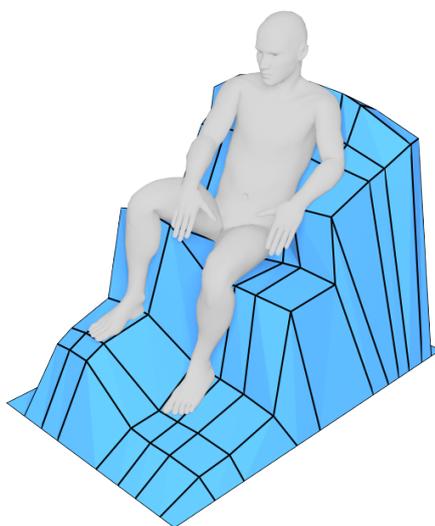
Both sets of results show similar *total support values* and therefore similar functional quality. While visual quality is subjective, the generic results clearly show less regularity in their overall shape. In addition, the results generated from the generic rectangles self-intersect in some areas of the mesh and are therefore not valid for fabrication. Therefore the rectangular patches are not reliable input shapes for this algorithm. To create valid results at a higher rate, larger smoothness parameter values are required, which negatively affects the functional quality of the results.



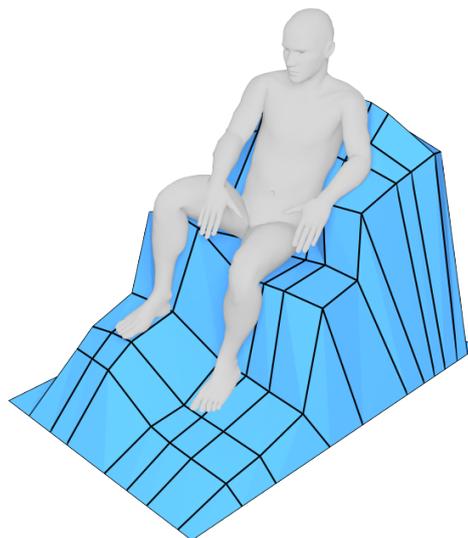
Frame 4 ( $S = 1, D = 5$ ):  
TS = 205.84, VE = 927.29



Frame 4 ( $S = 1, D = 2$ ):  
TS = 190.13, VE = 917.05

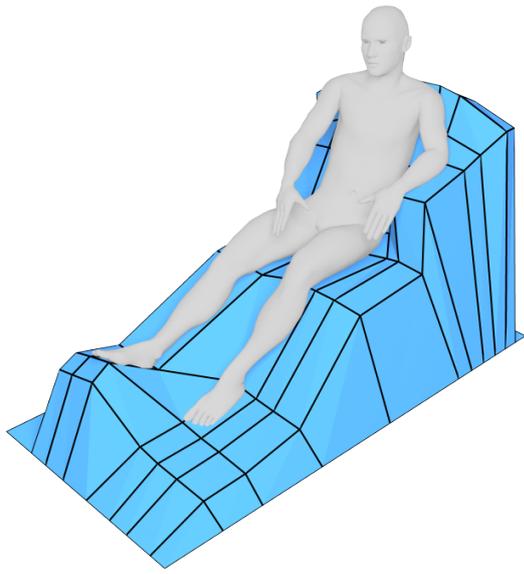


Frame 5 ( $S = 1, D = 5$ ):  
TS = 197.03, VE = 937.00

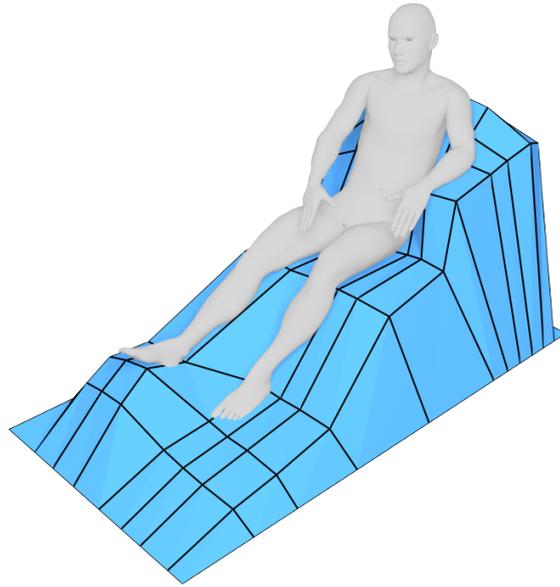


Frame 5 ( $S = 1, D = 2$ ):  
TS = 196.66, VE = 924.76

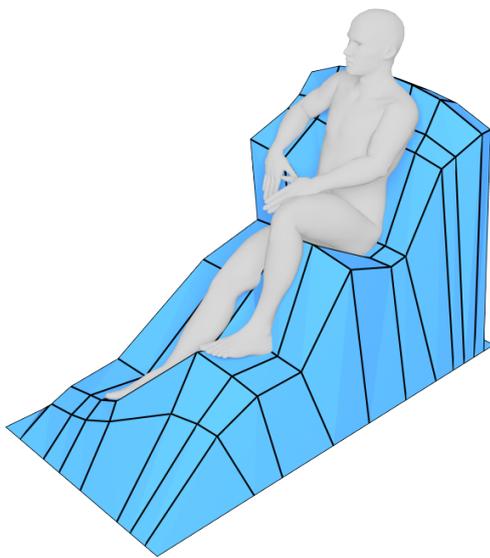
Figure 7.1: General results using different poses and varying parameter values for the smoothing ( $S$ ) and Data ( $D$ ) factors of the optimization.



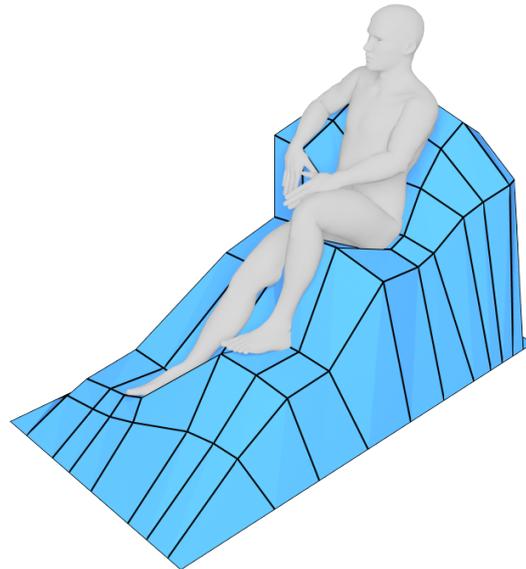
Frame 6 ( $S = 1, D = 5$ ):  
TS = 196.62, VE = 930.45



Frame 6 ( $S = 1, D = 2$ ):  
TS = 191.58, VE = 915.80

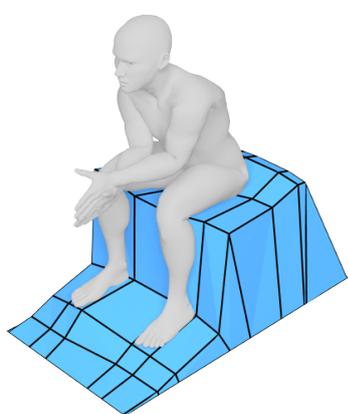


Frame 7 ( $S = 1, D = 5$ ):  
TS = 192.27, VE = 937.60

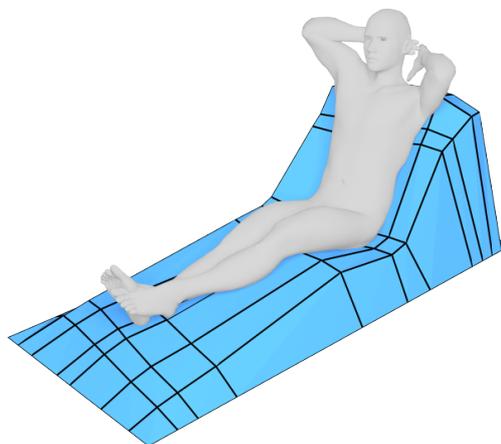


Frame 7 ( $S = 1, D = 2$ ):  
TS = 181.93, VE = 925.08

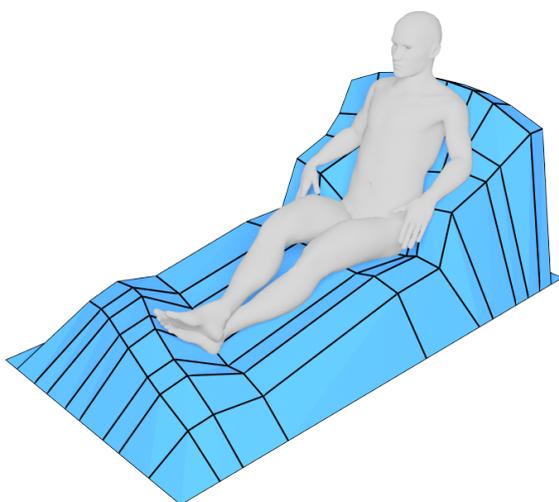
Figure 7.2: General results using different poses and varying parameter values for the smoothing ( $S$ ) and Data ( $D$ ) factors of the optimization.



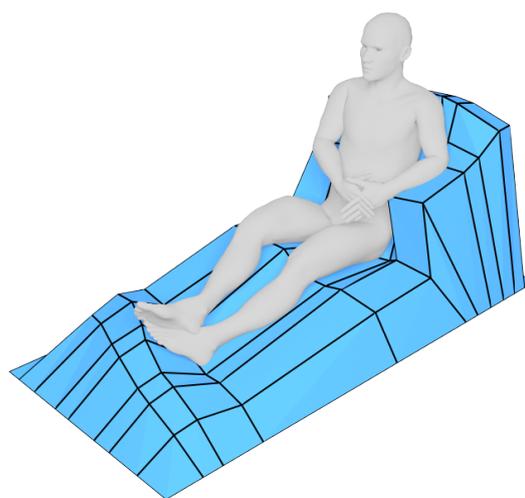
Frame 1 ( $S = 1, D = 5$ ):  
TS = 166.33, VE = 577.88



Frame 10 ( $S = 1, D = 5$ ):  
TS = 190.01, VE = 793.40

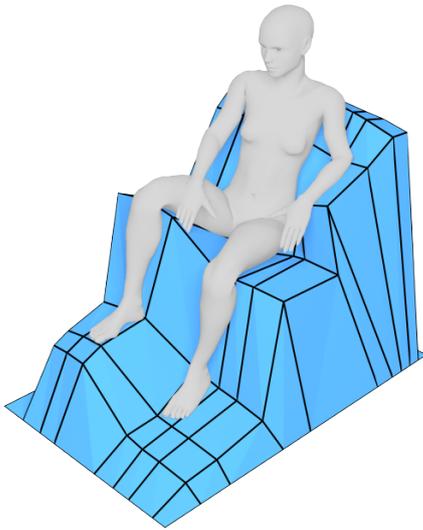


Frame 11 ( $S = 1, D = 5$ ):  
TS = 191.31, VE = 1021.89

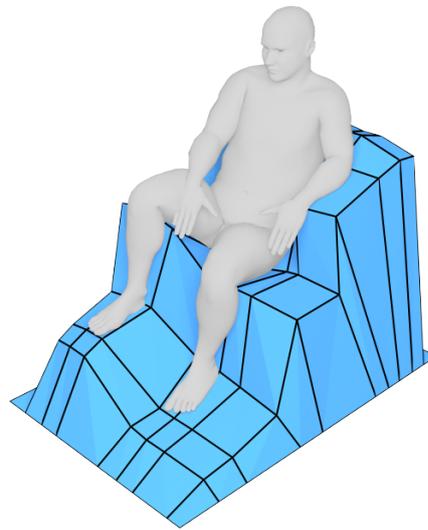


Frame 12 ( $S = 1, D = 5$ ):  
TS = 190.36, VE = 915.14

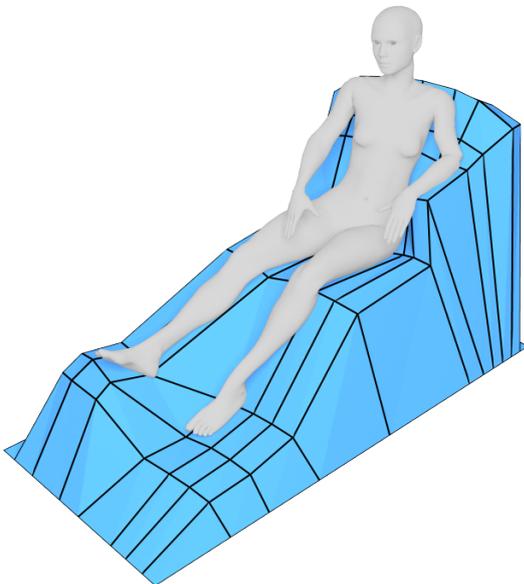
Figure 7.3: Special case results for poses with a forward leaning posture or crossed legs.



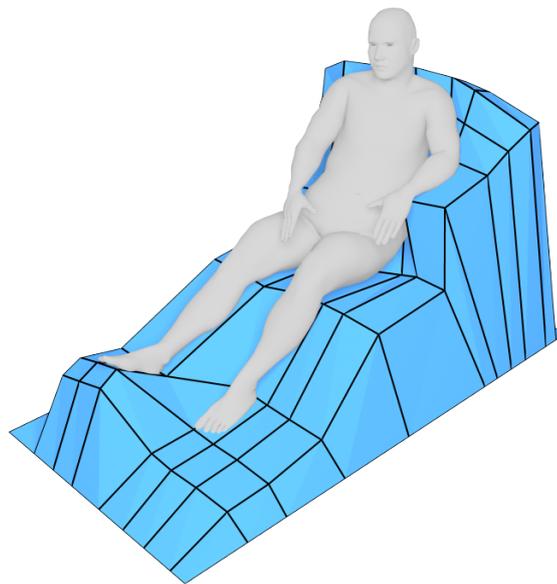
Frame 5 ( $S = 1$ ,  $D = 5$ ):  
TS = 236.56, VE = 852.85



Frame 5 ( $S = 1$ ,  $D = 5$ ):  
TS = 196.03, VE = 846.77

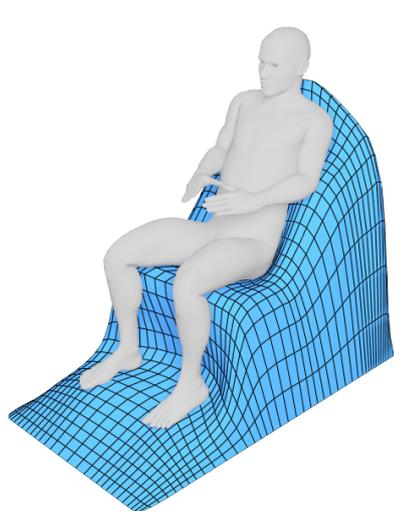


Frame 6 ( $S = 1$ ,  $D = 5$ ):  
TS = 242.06, VE = 933.59

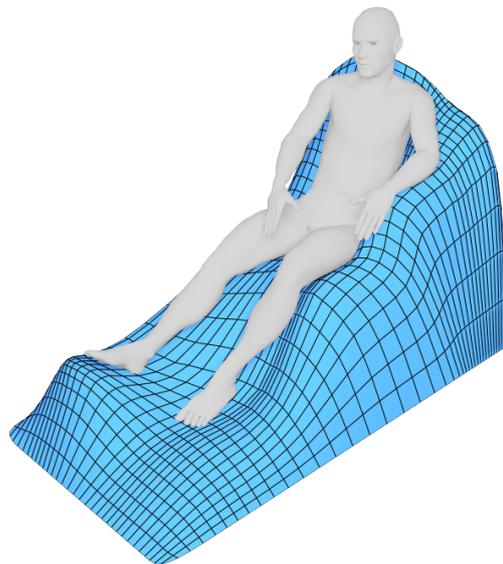


Frame 6 ( $S = 1$ ,  $D = 5$ ):  
TS = 217.35, VE = 930.85

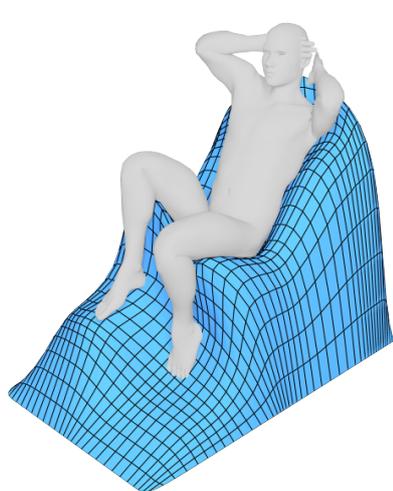
Figure 7.4: Seating surface results with female and overweight male body shapes.



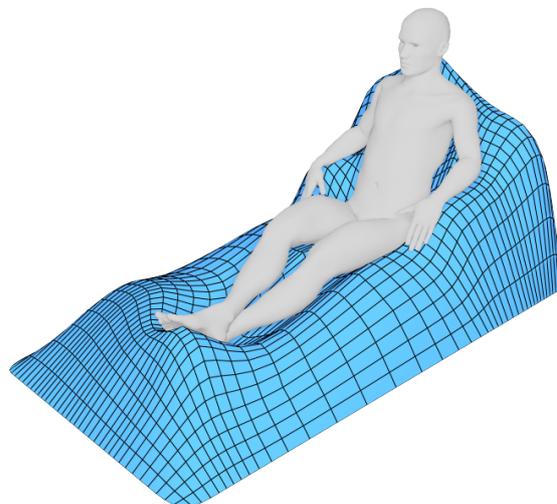
Frame 3 ( $S = 1, D = 5$ ):  
TS = 193.51



Frame 6 ( $S = 1, D = 5$ ):  
TS = 208.23

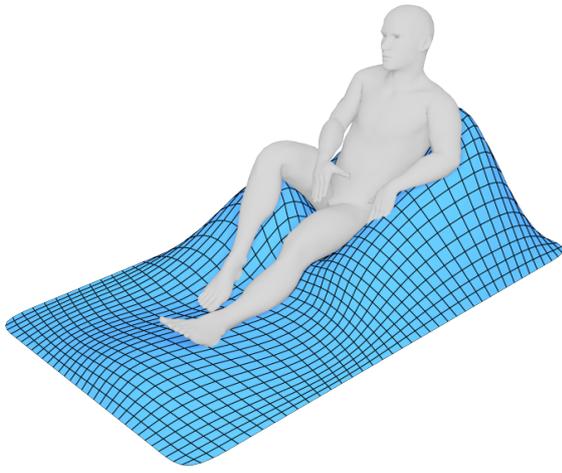


Frame 9 ( $S = 1, D = 5$ ):  
TS = 172.59



Frame 11 ( $S = 1, D = 5$ ):  
TS = 200.57

Figure 7.5: Seating surface results after utilizing the weighted fitting algorithm.



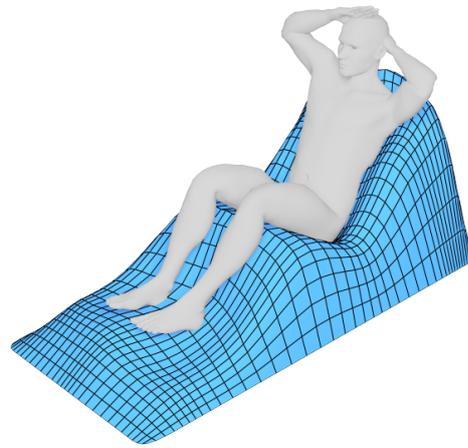
Frame 13:  
TS = 207.986



Frame 13 (S = 1, D = 5):  
TS = 205.57

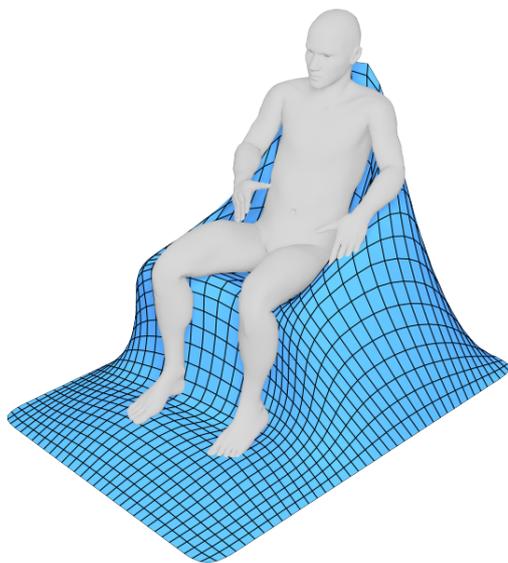


Frame 8:  
TS = 197.966

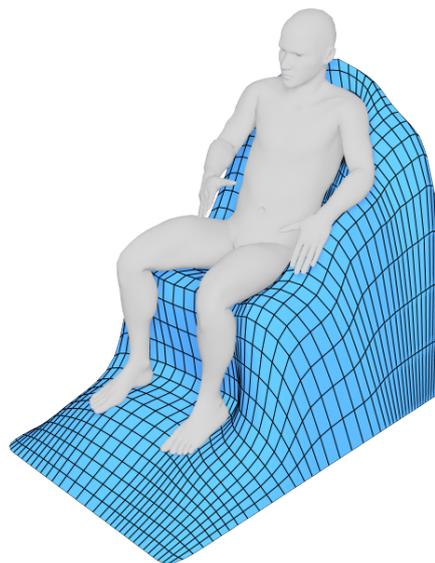


Frame 8 (S = 1, D = 5):  
TS = 191.51

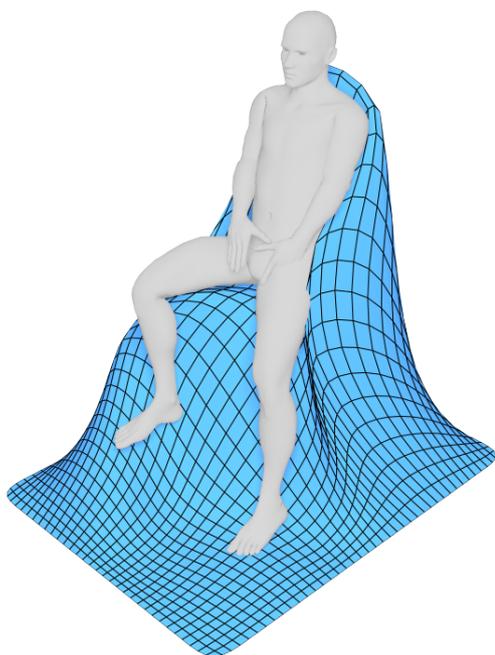
Figure 7.6: Weighted fitting results when using generic rectangular shapes (left) and our generated surfaces (right) as control mesh.



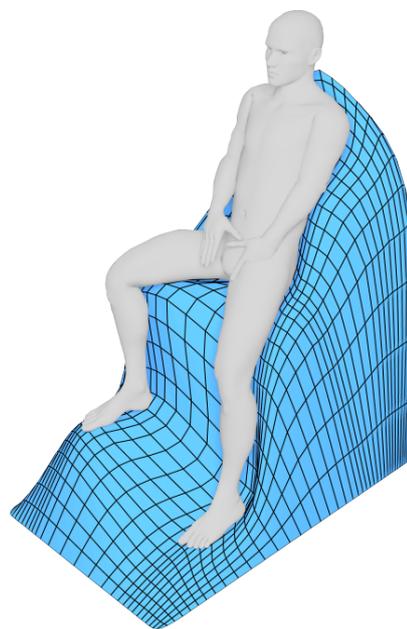
Frame 4:  
TS = 192.263



Frame 4 (S = 1, D = 5):  
TS = 203.56



Frame 34:  
TS = 168.996



Frame 34 (S = 1, D = 5):  
TS = 176.54

Figure 7.7: Weighted fitting results when using generic rectangular shapes (left) and our generated surfaces (right) as control mesh.

### 7.2.5 Unsupported poses

While the special case detection algorithm is able to deal with a number of difficult poses, the proposed framework cannot correctly process all kinds of sitting poses. Figure 7.8 shows generated results from unsupported poses where the surface intersects with the body shape on multiple occasions. While leaning poses are generally not supported by the framework, the algorithm was able to find a proper solution for such a pose.

### 7.2.6 Comparison to initial model

Figure 7.9 shows a visual comparison between the results generated from the initially proposed model as well as the advanced model. In an ideal case, the functional quality of the initial model is on par to the advanced model for simple sitting poses. While the simple model's surface is less detailed and accurate, the initial model is able to provide armrest support in situations where the advanced model can not, because of its additional requirements.

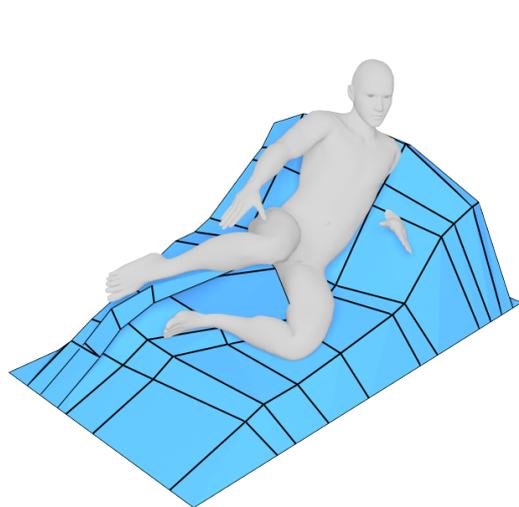
The biggest qualitative difference however, is the variety of poses that can be supported by the model without visual or functional errors. The generated results for frame 6 shown in Figure 7.9 are symbolic for an erroneous seating surface result. Overall the advanced model is much more robust and capable of properly supporting a much greater variety of sitting poses.

## 7.3 Evaluation

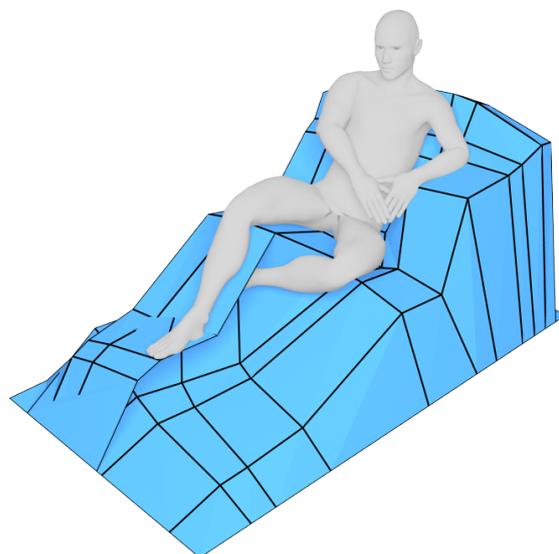
The following tables (7.1 & 7.2) show detailed results for different poses, body shapes and parameters sets. Each table row corresponds to an animation frame, where the quality measures as well as the algorithm run time is recorded in different stages of the algorithm.

- **F** is the animation frame number
- **TS** & **VE** refer to the *total support* and *visual error* values.
- **t(s)** is the runtime in seconds for the corresponding algorithm step.
- **fS** & **fD** refer to the scaling factors chosen for the data and smoothing terms in the optimization step.

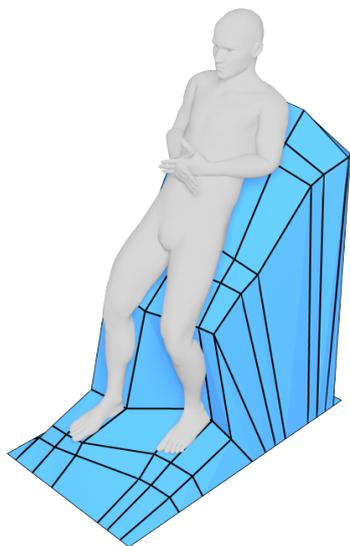
The evaluation was performed on an Intel i7-4770k CPU (3.5 GHz).



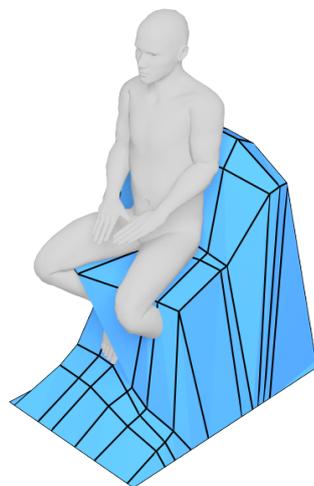
Frame 22 ( $S = 1, D = 5$ )



Frame 29 ( $S = 1, D = 5$ )



Frame 30 ( $S = 1, D = 5$ ):  
TS = 182.69, VE = 824.90



Frame 35 ( $S = 1, D = 5$ )

Figure 7.8: (Failed) seating surface results generated from unsupported poses.

| <b>body shape: male</b> |                     |         |       | <b>total vertex count: 11783</b> |     |        |         |       |                  |      |       |
|-------------------------|---------------------|---------|-------|----------------------------------|-----|--------|---------|-------|------------------|------|-------|
|                         | before optimization |         |       | after optimization               |     |        |         |       | weighted fitting |      | total |
| F                       | TS                  | VE      | t(s)  | fS                               | fD  | TS     | VE      | t(s)  | TS               | t(s) | t(s)  |
| 1                       | 162.45              | 599.25  | 6.21  | 1.0                              | 5.0 | 163.77 | 577.64  | 12.37 | 159.61           | 0.38 | 18.96 |
| 2                       | 195.17              | 858.44  | 8.97  | 1.0                              | 5.0 | 186.33 | 816.24  | 12.67 | 191.49           | 1.59 | 23.24 |
| 3                       | 191.87              | 839.90  | 9.44  | 1.0                              | 5.0 | 188.97 | 808.50  | 12.49 | 194.13           | 1.47 | 23.40 |
| 4                       | 200.43              | 966.12  | 9.20  | 1.0                              | 5.0 | 196.62 | 927.69  | 12.37 | 203.43           | 2.72 | 24.30 |
| 5                       | 197.06              | 985.60  | 9.20  | 1.0                              | 5.0 | 198.51 | 937.02  | 12.77 | 209.23           | 2.81 | 24.77 |
| 6                       | 199.10              | 977.13  | 9.53  | 1.0                              | 5.0 | 191.18 | 929.63  | 12.42 | 206.50           | 2.57 | 24.52 |
| 7                       | 194.07              | 986.12  | 9.34  | 1.0                              | 5.0 | 192.32 | 938.30  | 12.22 | 202.78           | 2.81 | 24.36 |
| 8                       | 199.51              | 816.50  | 9.63  | 1.0                              | 5.0 | 187.37 | 790.02  | 12.46 | 190.54           | 2.12 | 24.21 |
| 9                       | 173.19              | 867.20  | 9.72  | 1.0                              | 5.0 | 162.25 | 815.87  | 11.70 | 172.36           | 3.33 | 24.75 |
| 10                      | 193.74              | 828.83  | 10.48 | 1.0                              | 5.0 | 187.41 | 792.83  | 12.67 | 203.31           | 2.27 | 25.42 |
| 11                      | 193.03              | 1076.58 | 11.07 | 1.0                              | 5.0 | 193.57 | 1021.64 | 13.72 | 199.13           | 6.78 | 31.56 |
| 12                      | 193.51              | 961.63  | 10.02 | 1.0                              | 5.0 | 188.83 | 912.85  | 12.87 | 200.88           | 1.24 | 24.13 |
| 13                      | 201.00              | 992.06  | 9.28  | 1.0                              | 5.0 | 183.95 | 929.52  | 12.05 | 207.05           | 2.30 | 23.63 |
| 24                      | 192.19              | 834.62  | 9.97  | 1.0                              | 5.0 | 203.54 | 794.35  | 12.11 | 204.70           | 1.83 | 23.91 |
| 30                      | 190.10              | 863.07  | 9.19  | 1.0                              | 5.0 | 182.97 | 824.49  | 12.18 | 190.46           | 2.73 | 24.10 |
| 31                      | 164.07              | 805.08  | 8.35  | 1.0                              | 5.0 | 160.60 | 752.90  | 11.66 | 161.73           | 2.06 | 22.07 |
| 33                      | 162.47              | 799.31  | 8.37  | 1.0                              | 5.0 | 149.78 | 756.55  | 11.70 | 160.36           | 1.45 | 21.52 |
| 34                      | 176.47              | 860.34  | 9.41  | 1.0                              | 5.0 | 172.85 | 827.23  | 12.90 | 177.92           | 3.11 | 25.42 |
| 1                       | 156.63              | 599.64  | 6.45  | 1.0                              | 2.0 | 164.04 | 571.87  | 11.73 | 147.63           | 0.33 | 18.52 |
| 2                       | 190.06              | 858.37  | 8.91  | 1.0                              | 2.0 | 174.58 | 804.44  | 12.19 | 186.78           | 1.78 | 22.88 |
| 3                       | 192.83              | 836.77  | 9.40  | 1.0                              | 2.0 | 180.63 | 798.29  | 12.28 | 186.18           | 1.45 | 23.14 |
| 4                       | 204.31              | 964.75  | 9.04  | 1.0                              | 2.0 | 189.07 | 918.14  | 12.38 | 205.63           | 2.36 | 23.78 |
| 5                       | 197.17              | 984.48  | 9.15  | 1.0                              | 2.0 | 199.75 | 924.45  | 11.91 | 208.46           | 1.98 | 23.04 |
| 6                       | 201.28              | 977.25  | 9.48  | 1.0                              | 2.0 | 190.30 | 915.90  | 12.68 | 206.12           | 1.75 | 23.91 |
| 7                       | 201.11              | 983.59  | 9.29  | 1.0                              | 2.0 | 185.21 | 924.20  | 11.67 | 203.27           | 2.21 | 23.17 |
| 8                       | 201.32              | 818.93  | 9.65  | 1.0                              | 2.0 | 174.33 | 783.32  | 11.71 | 186.23           | 1.82 | 23.17 |
| 9                       | 174.98              | 865.76  | 9.68  | 1.0                              | 2.0 | 149.73 | 801.96  | 11.66 | 165.21           | 2.38 | 23.72 |
| 10                      | 196.48              | 828.17  | 10.47 | 1.0                              | 2.0 | 185.46 | 783.40  | 11.91 | 200.97           | 2.06 | 24.43 |
| 11                      | 186.81              | 1080.64 | 10.87 | 1.0                              | 2.0 | 193.00 | 1012.14 | 13.73 | 201.96           | 6.48 | 31.09 |
| 12                      | 194.61              | 964.14  | 10.02 | 1.0                              | 2.0 | 193.35 | 904.57  | 13.03 | 198.77           | 1.36 | 24.42 |
| 13                      | 200.16              | 991.94  | 9.31  | 1.0                              | 2.0 | 176.08 | 913.55  | 11.82 | 206.88           | 2.62 | 23.75 |
| 24                      | 191.36              | 1024.95 | 9.91  | 1.0                              | 2.0 | 183.56 | 915.98  | 12.49 | 206.90           | 3.03 | 25.43 |
| 30                      | 194.57              | 865.23  | 9.25  | 1.0                              | 2.0 | 174.89 | 810.93  | 12.07 | 176.37           | 1.47 | 22.79 |
| 31                      | 165.39              | 807.27  | 8.44  | 1.0                              | 2.0 | 145.81 | 739.89  | 11.59 | 174.83           | 2.08 | 22.11 |
| 33                      | 169.43              | 800.03  | 8.40  | 1.0                              | 2.0 | 136.40 | 742.36  | 11.56 | 144.77           | 1.72 | 21.67 |
| 34                      | 181.71              | 860.71  | 9.40  | 1.0                              | 2.0 | 162.65 | 812.92  | 12.44 | 176.69           | 1.38 | 23.22 |

Table 7.1: Table of results using the default male body shape model.

7. RESULTS

| body shape: female |                     |         |       | total vertex count: 11997 |     |        |         |       |                  |      |       |
|--------------------|---------------------|---------|-------|---------------------------|-----|--------|---------|-------|------------------|------|-------|
|                    | before optimization |         |       | after optimization        |     |        |         |       | weighted fitting |      | total |
| F                  | TS                  | VE      | t(s)  | fS                        | fD  | TS     | VE      | t(s)  | TS               | t(s) | t(s)  |
| 1                  | 184.50              | 600.43  | 7.07  | 1.0                       | 5.0 | 179.28 | 578.64  | 13.36 | 177.78           | 0.56 | 20.98 |
| 2                  | 212.12              | 765.31  | 8.60  | 1.0                       | 5.0 | 195.67 | 737.23  | 12.54 | 210.98           | 2.17 | 23.30 |
| 3                  | 221.90              | 837.30  | 9.62  | 1.0                       | 5.0 | 211.35 | 807.50  | 12.23 | 227.26           | 1.96 | 23.80 |
| 4                  | 239.43              | 973.72  | 9.82  | 1.0                       | 5.0 | 236.34 | 930.08  | 11.76 | 241.39           | 2.38 | 23.96 |
| 5                  | 225.99              | 893.63  | 8.87  | 1.0                       | 5.0 | 233.04 | 851.64  | 11.78 | 238.21           | 1.89 | 22.54 |
| 6                  | 245.33              | 993.20  | 9.73  | 1.0                       | 5.0 | 243.30 | 932.23  | 11.61 | 248.92           | 3.38 | 24.72 |
| 8                  | 244.69              | 821.84  | 10.56 | 1.0                       | 5.0 | 231.00 | 792.60  | 12.32 | 240.05           | 2.68 | 25.57 |
| 9                  | 215.78              | 880.91  | 10.04 | 1.0                       | 5.0 | 205.17 | 823.57  | 11.91 | 219.53           | 1.54 | 23.49 |
| 10                 | 240.40              | 858.73  | 10.46 | 1.0                       | 5.0 | 246.57 | 802.82  | 12.34 | 251.32           | 1.60 | 24.40 |
| 11                 | 228.60              | 1167.32 | 11.10 | 1.0                       | 5.0 | 243.02 | 1036.75 | 13.33 | 262.01           | 6.68 | 31.11 |
| 12                 | 234.18              | 867.37  | 9.60  | 1.0                       | 5.0 | 224.78 | 825.95  | 12.01 | 247.91           | 2.40 | 24.02 |
| 13                 | 237.28              | 911.48  | 8.89  | 1.0                       | 5.0 | 206.00 | 846.66  | 11.36 | 237.18           | 3.20 | 23.45 |
| 24                 | 238.96              | 897.91  | 9.53  | 1.0                       | 5.0 | 235.06 | 834.60  | 11.51 | 247.73           | 3.02 | 24.05 |
| 31                 | 189.02              | 632.83  | 6.44  | 1.0                       | 5.0 | 182.24 | 593.70  | 11.20 | 181.24           | 2.59 | 20.23 |
| 33                 | 197.50              | 800.53  | 8.33  | 1.0                       | 5.0 | 177.96 | 759.93  | 11.27 | 199.75           | 1.63 | 21.24 |
| 34                 | 203.93              | 866.24  | 9.83  | 1.0                       | 5.0 | 190.26 | 830.25  | 11.47 | 209.64           | 1.89 | 23.19 |
| 1                  | 185.16              | 600.41  | 6.16  | 1.0                       | 2.0 | 168.09 | 572.01  | 11.31 | 164.86           | 0.47 | 17.95 |
| 2                  | 205.79              | 765.63  | 9.12  | 1.0                       | 2.0 | 169.80 | 726.75  | 12.41 | 208.96           | 1.75 | 23.28 |
| 3                  | 225.44              | 837.45  | 10.35 | 1.0                       | 2.0 | 194.72 | 798.76  | 12.76 | 224.28           | 2.47 | 25.57 |
| 4                  | 236.59              | 979.13  | 10.00 | 1.0                       | 2.0 | 223.51 | 920.16  | 12.14 | 240.30           | 2.27 | 24.41 |
| 5                  | 227.52              | 896.41  | 9.05  | 1.0                       | 2.0 | 220.41 | 841.71  | 12.06 | 239.13           | 3.15 | 24.26 |
| 6                  | 251.78              | 992.78  | 10.05 | 1.0                       | 2.0 | 234.82 | 917.86  | 12.20 | 249.85           | 2.20 | 24.46 |
| 7                  | 230.70              | 989.89  | 10.16 | 1.0                       | 2.0 | 203.17 | 922.55  | 11.95 | 244.62           | 2.63 | 24.75 |
| 8                  | 249.14              | 820.86  | 11.27 | 1.0                       | 2.0 | 219.58 | 787.27  | 11.69 | 245.29           | 2.46 | 25.42 |
| 9                  | 218.24              | 883.43  | 10.18 | 1.0                       | 2.0 | 197.78 | 808.88  | 11.44 | 217.26           | 1.78 | 23.41 |
| 10                 | 251.55              | 830.12  | 10.52 | 1.0                       | 2.0 | 241.96 | 784.84  | 12.44 | 247.44           | 1.44 | 24.40 |
| 11                 | 238.94              | 1091.65 | 11.19 | 1.0                       | 2.0 | 235.25 | 1011.14 | 12.71 | 259.34           | 5.75 | 29.65 |
| 12                 | 233.34              | 866.49  | 9.71  | 1.0                       | 2.0 | 210.41 | 815.02  | 12.39 | 246.80           | 4.90 | 27.00 |
| 13                 | 236.84              | 912.71  | 8.93  | 1.0                       | 2.0 | 182.28 | 831.06  | 11.54 | 230.33           | 2.64 | 23.10 |
| 24                 | 219.69              | 758.32  | 9.53  | 1.0                       | 2.0 | 187.60 | 704.54  | 11.41 | 239.62           | 1.91 | 22.85 |
| 30                 | 246.48              | 869.33  | 9.76  | 1.0                       | 2.0 | 218.60 | 814.57  | 11.90 | 210.01           | 4.04 | 25.70 |
| 31                 | 189.12              | 633.30  | 6.52  | 1.0                       | 2.0 | 171.08 | 584.72  | 11.33 | 180.22           | 1.26 | 19.11 |
| 33                 | 199.47              | 804.76  | 8.38  | 1.0                       | 2.0 | 180.96 | 745.88  | 11.22 | 189.84           | 1.40 | 21.00 |
| 34                 | 202.60              | 866.08  | 10.38 | 1.0                       | 2.0 | 172.64 | 815.86  | 12.68 | 200.12           | 1.46 | 24.51 |

Table 7.2: Table of results using the default female body shape model.

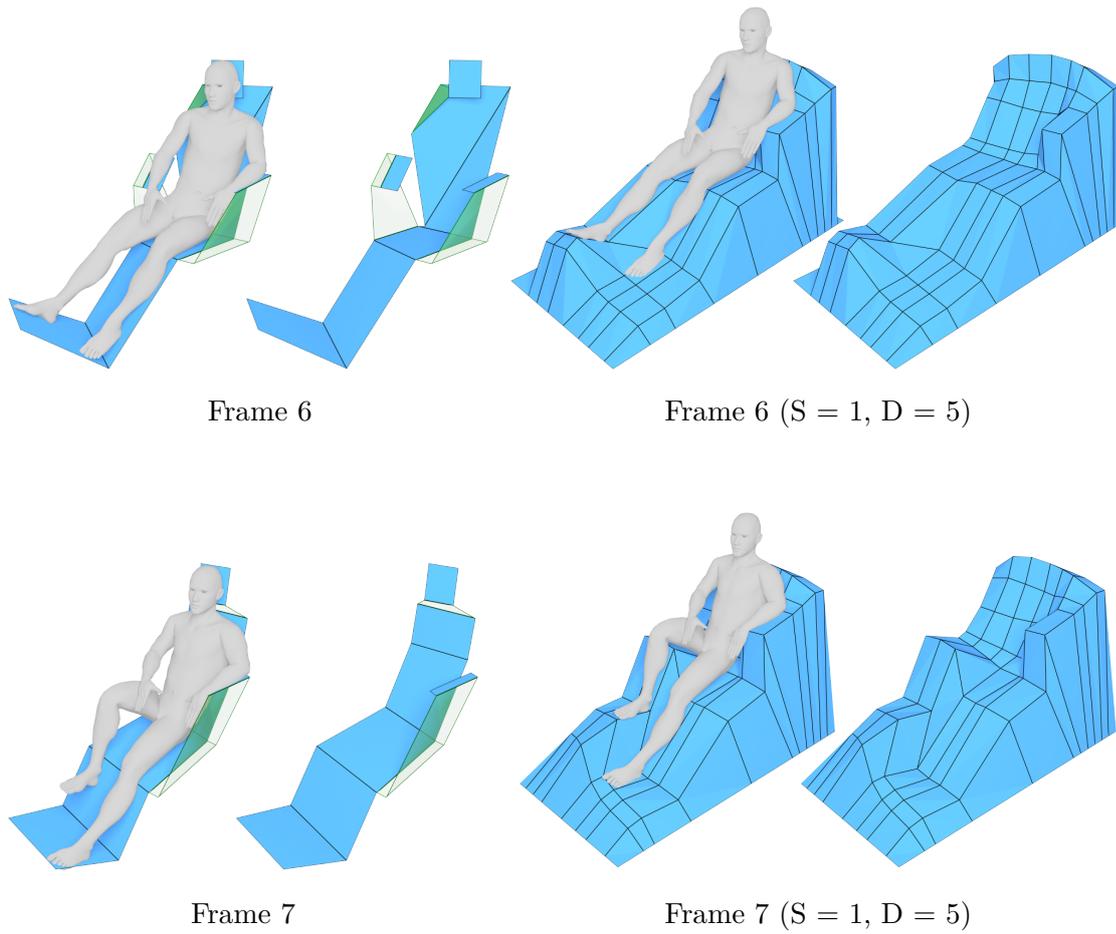


Figure 7.9: Visual comparison of results generated from the initial model and the advanced variant.





# Conclusion

In this thesis we proposed a computational approach for personalized furniture design. In this final chapter, we summarize our contributions and discuss some of the limitations and shortcomings of the proposed framework. Furthermore, possible future improvements to the framework are proposed.

## 8.1 Contributions

The main contribution of this work is a software framework for the automated creation of seating furniture models. The proposed algorithm is capable of generating personalized furniture models for a variety of sitting poses fully automatically. Given a specific human body shape and sitting pose, the framework generates a seating surface model with the primary goal of maximizing sitting comfort while aiming for a high level of aesthetics. As comfort is seen as a subjective feeling of a specific person, we utilize objective comfort measures based on pressure distribution to approximate a person's sitting comfort.

The proposed algorithm operates on multiple stages to generate a suitable seating surface:

- In an initial surface fitting step, a generic template model consisting of a loose arrangement of planes is fitted to the individual body parts of a person's body in a given sitting pose. Based on the positions and orientations of the planes, a connected surface mesh is generated in a regular grid structure.
- The second stage of the algorithm is a non-linear optimization process which further refines the surface mesh in order to maximize its visual quality while maintaining its ability to support the person's body.

- As final and optional stage of the framework, we utilize the weighted fitting algorithm by Leimer et al. to create a more detailed, higher resolution mesh, capable of providing a closer fit to the body shape. For this task, the previously created surface model is used as control mesh in the weighted fitting algorithm.
- Accordingly our proposed framework serves the additional purpose of providing detailed control meshes for the weighted fitting algorithm, thus eliminating the need of manual design effort.

## 8.2 Limitations and future work

While the developed framework fulfills our goals to a satisfying degree, we acknowledge a number of limitations and weaknesses. In this section, certain problem areas of the framework are reviewed and possible improvements are proposed.

### 8.2.1 Input poses

The goal for this framework is to create suitable seating surfaces for a variety of sitting poses. While the algorithm covers various difficult special cases, there is a number of common sitting poses that are currently not supported.

This includes poses where a person is sitting in a sideways orientation or one or both feet are tucked under the body. Furthermore the framework does not correctly interpret sitting poses where a person's legs are dangling in the air, rather than being connected to a ground surface. 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.

In addition to difficult sitting poses, the algorithm could be extended in the future to support other classes of poses such as leaning or lying. We assume that the general principle of supporting body parts with individual planes is theoretically applicable for a multitude of classes of poses.

### 8.2.2 Multiple poses

The current framework is designed to create a surface that optimally supports a specific body shape in a single pose. In future work, the framework could be expanded to create multi-purpose surfaces. One possible method is to process multiple body shapes and poses in order to create a surface that provides satisfying support for all input variants. A different possible approach would be to create seating surface that is capable of supporting multiple persons at once. The general idea is to create a larger surface containing segments supporting each person's individual body parts.

### 8.2.3 Quality measures

The design and optimization goals to maximize the the visual and functional quality of a seating surface are based on a number of approximations. Ultimately, both comfort and the aesthetics of a seating surface are subjective features that cannot be maximized in a general context.

While the chosen model for functional quality is based on the insights of numerous studies and research on objective comfort measures it makes use of number of approximations and simplifications. For future improvements of the framework, a more precise physical model could potentially increase the functional quality of the created seating surfaces.

The visual quality measures, utilized in the optimization stage of the algorithm, were chosen based on subjective visual evaluation of the resulting surface models. While the concepts of planarity and regularity are utilized in architecture for the creation of free form surfaces, our approach at visual refinement is lacking in concrete scientific evidence. Therefore, future work suggests to perform user studies on the visual quality of seating surface meshes to gain feedback from general users as well as professional artists and furniture design experts.

### 8.2.4 Geometry refinement

The proposed framework generates a connected seating surface mesh, consisting of quadrilateral faces. While the algorithm attempts to keep the surface mesh as planar as possible, general planarity is not guaranteed. Most created models contain a number of faces with low planarity, mostly at the border of the surface. When the finished model is triangulated for rendering or fabrication, large non-planar faces can negatively affect the aesthetics of the model. In addition, the functional quality of non-planar supporting quadrilateral faces is dependent on its triangulation.

For an additional refinement stage, a custom triangulation algorithm could be developed, which chooses a mesh triangulation that maximizes functional and visual quality of the seating surface.

### 8.2.5 Conclusion

Overall, the proposed framework fulfills the originally formulated requirements and improves the efficiency of the furniture design process for inexperienced users as well as professionals. However, fully automatic computational design still does not entirely replace human creativity or the need for subjective evaluation.



# List of Figures

|      |   |    |
|------|---|----|
| 2.1  | 3D furniture model converted into individual parts. [LOMI11] . . . . .                          | 9  |
| 2.2  | SketchChair [SLMI11] promotional image showing individual parts and an assembled chair. . . . . | 9  |
| 2.3  | Chair built from interlocking parts. [SP13] . . . . .   | 10 |
| 2.4  | Table assembled from 3D printed structural nodes and timber elements. [HMBB17]                  | 10 |
| 2.5  | Model of comfort and discomfort by de Looze et al. [DLKEVD03] . . . . .                         | 12 |
| 2.6  | Overview of the pose-inspired shape synthesis algorithm by Fu et al. [FCSF17]                   | 16 |
| 2.7  | Optimal furniture arrangement and sitting position in ActiveErgo. [WWT <sup>+</sup> 18]         | 18 |
| 2.8  | User input for furniture design via gestures and voice commands. [LCMS16]                       | 18 |
| 2.9  | Intermediate steps in the Sit&Relax [LBRM18] framework . . . . .                                | 19 |
| 2.10 | Multi-purpose surface created for sitting and leaning poses. [LBRM18] . . . . .                 | 20 |
| 2.11 | Laplacian smoothing step on a single vertex . . . . .   | 23 |
| 2.12 | Laplacian smoothing on a triangle mesh . . . . .  | 23 |
| 2.13 | Catmull-Clarke Subdivision Surface applied to a cube. [CPH15] . . . . .                         | 25 |
| 3.1  | An overview of the proposed system . . . . .  | 30 |
| 3.2  | The human skeleton model used in the proposed framework . . . . .                               | 32 |
| 3.3  | Rendered images of human body models in various sitting poses. . . . .                          | 33 |
| 4.1  | Visual comparison of the hierarchical model variants . . . . .                                  | 41 |
| 4.2  | Overview of the free faces model . . . . .  | 43 |
| 4.3  | Visual example for the geometry of the initial template model . . . . .                         | 45 |
| 4.4  | Seating surface results generated from the initial template model. . . . .                      | 50 |
| 4.5  | Body part mapping in the advanced template model . . . . .                                      | 51 |
| 4.6  | Overview of the mesh generation process of the advanced template model                          | 53 |
| 4.7  | Plane intersection and vertex generation process for upper body inner segment vertices. . . . . | 57 |
| 4.8  | Intermediate seating surface results generated from the advanced template model . . . . .       | 59 |
| 4.9  | Error handling on poses with crossed legs . . . . .   | 61 |
| 4.10 | Advanced model seating surface results after the refinement stage. . . . .                      | 63 |
| 5.1  | Finalized seating surface results after the optimization process . . . . .                      | 76 |

|     |   |     |
|-----|---|-----|
| 6.1 | An extended overview of the proposed system . . . . .   | 78  |
| 6.2 | Grasshopper component and Rhino 3D Scene. . . . .   | 87  |
| 7.1 | General results with varying parameters. . . . .  | 92  |
| 7.2 | General results with varying parameters. . . . .  | 93  |
| 7.3 | Special case results for poses with a forward leaning posture or crossed legs. . . . .          | 94  |
| 7.4 | Seating surface results with female and overweight male body shapes. . . . .                    | 95  |
| 7.5 | Seating surface results after utilizing the weighted fitting algorithm. . . . .                 | 96  |
| 7.6 | Comparison of weighted fitting results. . . . .   | 97  |
| 7.7 | Comparison of weighted fitting results. . . . .   | 98  |
| 7.8 | (Failed) seating surface results generated from unsupported poses. . . . .                      | 100 |
| 7.9 | Visual comparison of results generated from the initial model and the advanced variant. . . . . | 103 |

# List of Tables

|     |   |     |
|-----|---|-----|
| 7.1 | Table of results using the default male body shape model. . . . .   | 101 |
| 7.2 | Table of results using the default female body shape model. . . . . | 102 |



# List of Algorithms

|      |   |    |
|------|---|----|
| 6.1  | Input data loading and preprocessing ( <code>generateBaseModel()</code> ) . . . . .   | 79 |
| 6.2  | Surface fitting for legs and feet ( <code>generateBaseModel()</code> ) . . . . .      | 79 |
| 6.3  | surface fitting for (inner) upper body segments. ( <code>generateBaseModel()</code> ) | 80 |
| 6.4  | Crossed legs special case detection. ( <code>generateBaseModel()</code> ) . . . . .   | 80 |
| 6.5  | Mesh generation for upper body segments. ( <code>generateBaseModel()</code> ) . . .   | 80 |
| 6.6  | Outer vertex generation ( <code>generateBaseModel()</code> ) . . . . .                | 81 |
| 6.7  | Refinement & post processing ( <code>generateBaseModel()</code> ) . . . . .           | 81 |
| 6.8  | Optimization process ( <code>optimizeControlMesh()</code> ) . . . . .                 | 82 |
| 6.9  | Objective function ( <code>pfun()</code> ) Part 1 of 2 . . . . .                      | 83 |
| 6.10 | Objective function ( <code>pfun()</code> ) Part 2 of 2 . . . . .                      | 84 |
| 6.11 | Non-linear constraint function ( <code>pcon()</code> ) . . . . .                      | 84 |
| 6.12 | Constrained RANSAC surface fitting algorithm ( <code>optimizePlane_ransac()</code> )  | 86 |



# Bibliography

- [ACSYD05] Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun. Variational tetrahedral meshing. In *ACM SIGGRAPH 2005 Courses*, page 10. ACM, 2005.
- [Bas18] Manuel Bastioni. Manuel bastioni laboratory: open source 3d software, 3d human models, 3d innovation. "<http://www.manuelbastioni.com>", 2018.
- [BHRS91] Ram R. Bishu, M. Susan Hallbeck, Michael W. Riley, and Terry L. Stentz. Seating comfort and its relationship to spinal profile: A pilot study. *International Journal of Industrial Ergonomics*, 8(1):89–101, 1991.
- [Bra69] Philip E. Branton. Behaviour, body mechanics and discomfort. *Ergonomics*, 12(2):316–327, 1969. PMID: 5810917.
- [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.
- [CC78] Edwin Catmull and James Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-aided design*, 10(6):350–355, 1978.
- [Che04] Long Chen. Mesh smoothing schemes based on optimal delaunay triangulations. In *IMR*, pages 109–120. Citeseer, 2004.
- [CPH15] Yueyue Cheng, Henri Palleis, and Wolfgang Höhl. Simple 3d low poly modeling tool with intuitive gui. 07 2015.
- [DLKEVD03] Michiel P. De Looze, Lottie F. M. Kuijt-Evers, and Jaap Van Dieen. Sitting comfort and discomfort and the relationships with objective measures. *Ergonomics*, 46(10):985–997, 2003.
- [EC87] Jörgen A. E. Eklund and E. Nigel Corlett. Evaluation of spinal loads and chair design in seated work tasks. *clinical Biomechanics*, 2(1):27–33, 1987.

- [EÜZ09] Hale Erten, Alper Üngör, and Chunchun Zhao. Mesh smoothing algorithms for complex geometric domains. In *Proceedings of the 18th International Meshing Roundtable*, pages 175–193. Springer, 2009.
- [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.
- [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, 2017.
- [Fre97] Lori A. Freitag. On combining laplacian and optimization-based mesh smoothing techniques. Technical report, Argonne National Lab., IL (United States), 1997.
- [FS07] Jeng-Neng Fan and Daniel Schodek. Personalized furniture within the condition of mass production. In *The 9th international conference on ubiquitous computing (Ubicomp '07)*. Innsbruck, Austria, 2007.
- [GGVG11] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1529–1536. IEEE, 2011.
- [GZZ<sup>+</sup>17] Nenad Grujovic, Fatima Zivic, Miroslav Zivkovic, Milan Sljivic, Andreja Radovanovic, Luka Bukvic, Milos Mladenovic, and Aleksandar Sindjelic. Custom design of furniture elements by fused filament fabrication. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(1):88–95, 2017.
- [Her58] Hans T. E. Hertzberg. Seat comfort. *Annotated bibliography of applied physical anthropology in human engineering*, WADC Technical Report, pages 56–30, 1958.
- [HMBB17] M Haeusler, MANUEL Muehlbauer, SASCHA Bohnenberger, and JANE Burry. Furniture design using custom-optimised structural nodes. In *22nd International Conference of the Association for Computer-Aided Architectural Design Research in Asia*, pages 841–851. The Association for Computer-Aided Architectural Design Research in Asia, 2017.
- [HZ97] Martin G. Helander and Lijian Zhang. Field studies of comfort and discomfort in sitting. *Ergonomics*, 40(9):895–915, 1997.
- [IBMT82] G. I. Bardsley and P. M. Taylor. The development of an assessment chair. *Prosthetics and orthotics international*, 6:75–8, 09 1982.

- [JL94] Zhao Jianghong and Tang Long. An evaluation of comfort of a bus seat. *Applied Ergonomics*, 25(6):386–392, 1994.
- [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.
- [KHLM17] Bongjin Koo, Jean Hergel, Sylvain Lefebvre, and Niloy J Mitra. Towards zero-waste furniture design. *IEEE transactions on visualization and computer graphics*, 23(12):2627–2640, 2017.
- [KL14] Changgu Kang and Sung-Hee Lee. Environment-adaptive contact poses for virtual characters. In *Computer Graphics Forum*, volume 33, pages 1–10. Wiley Online Library, 2014.
- [Kru00] John Krumm. Intersection of two planes. "<https://www.microsoft.com/en-us/research/publication/intersection-of-two-planes/>", May 2000.
- [KTOK82] Ken Kamijo, Harutoshi Tsujimura, Hideo Obara, and Masaaki Katsumata. Evaluation of seating comfort. *SAE transactions*, pages 2615–2620, 1982.
- [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.
- [LFT93] Janilla Lee, Paul Ferraiuolo, and J Temming. Measuring seat comfort. *Automotive Engineering*, pages 25–30, 07 1993.
- [Lo85] S. H. Lo. A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, 21(8):1403–1426, 1985.
- [LOMI11] Manfred Lau, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. Converting 3d furniture models to fabricatable parts and connectors. In *ACM Transactions on Graphics (TOG)*, volume 30, page 85. ACM, 2011.
- [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.

- [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.
- [MH94] Dominique P. Michel and Martin G. Helander. Effects of two types of chairs on stature change and comfort for individuals with healthy and herniated discs. *Ergonomics*, 37(7):1231–1244, 1994.
- [MM<sup>+</sup>01] Maddock Meredith, S. Maddock, et al. Motion capture file formats explained. *Department of Computer Science, University of Sheffield*, 211:241–244, 2001.
- [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.
- [Pau97] Rajendra D. Paul. Nurturing and pampering paradigm for office ergonomics. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 41, pages 519–523. SAGE Publications Sage CA: Los Angeles, CA, 1997.
- [Ric80] LG Richards. On the psychology of passenger comfort. *human factors in transport research edited by Dj Osborne, Ja Levis*, 2, 1980.
- [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.
- [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.
- [SC99] A. J. Salewytch and J. P. Callaghan. Can quantified lumbar spine postures and trunk muscle activation levels predict discomfort during prolonged sitting. In *Proceedings of the 31st Annual Conference of the Human Factors Association of Canada*, pages 316–321, 1999.
- [SCS69] B. Shackel, K. D. Chidsey, and Pat Shipley. The assessment of chair comfort. *Ergonomics*, 12(2):269–306, 1969.
- [SKŠ<sup>+</sup>10] Miroslav Svub, Přemysl Krsek, Michal Španěl, Vít Štancl, Radek Bartoň, and Jiří Vadura. Feature preserving mesh smoothing algorithm based on local normal covariance. 2010.

- [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*, pages 73–80. ACM, 2011.
- [SP13] Yuliy Schwartzburg and Mark Pauly. Fabrication-aware design with intersecting planar pieces. In *Computer Graphics Forum*, volume 32, pages 317–326. Wiley Online Library, 2013.
- [Sta98] Jos Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Siggraph*, volume 98, pages 395–404. Citeseer, 1998.
- [Tau95] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.
- [Tau00] Gabriel Taubin. Geometric signal processing on polygonal meshes. *Proc. of EUROGRAPHICS 2000*, 2000.
- [UIM12] Nobuyuki Umetani, Takeo Igarashi, and Niloy J Mitra. Guided exploration of physically valid shapes for furniture design. *ACM Trans. Graph.*, 31(4):86–1, 2012.
- [VDVD<sup>+</sup>11] Vincent Verhaert, Hans Druyts, Dorien Van Deun, Daniel Berckmans, Johan Verbraecken, Marie Vandekerckhove, Bart Haex, and Jos Vander Sloten. The use of a generic human model to personalize bed design. In *Proceedings of 1st international symposium on digital human modeling*, volume 2202, 2011.
- [VMM99] Jörg Vollmer, Robert Mencl, and Heinrich Mueller. Improved laplacian smoothing of noisy surface meshes. In *Computer graphics forum*, volume 18, pages 131–138. Wiley Online Library, 1999.
- [WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [WWT<sup>+</sup>18] Yu-Chian Wu, Te-Yen Wu, Paul Taele, Bryan Wang, Jun-You Liu, Pingsung 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.
- [XN06] Hongtao Xu and Timothy S. Newman. An angle-based optimization approach for 2d finite element mesh smoothing. *Finite Elements in Analysis and Design*, 42(13):1150–1164, 2006.

- [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.
- [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*, 38(3):377–389, 1996.
- [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, 2016.
- [ZS00] Tian Zhou and Kenji Shimada. An angle-based approach to two-dimensional mesh smoothing. *IMR*, 2000:373–384, 2000.
- [ZSW10] Mirko Zadavec, Alexander Schiftner, and Johannes Wallner. Designing quad-dominant meshes with planar faces. *Computer Graphics Forum*, 29(5):1671–1679, 2010.