

# GarmentCodeData: A Dataset of 3D Made-to-Measure Garments With Sewing Patterns

Maria Korosteleva<sup>1</sup>, Timur Levent Kesdogan<sup>1</sup>, Fabian Kemper<sup>2</sup>, Stephan Wenninger<sup>2</sup>, Jasmin Koller<sup>1</sup>, Yuhang Zhang<sup>1</sup>, Mario Botsch<sup>2</sup>, and Olga Sorkine-Hornung<sup>1</sup>

<sup>1</sup> ETH Zurich, Rämistrasse 101, 8092 Zurich, Switzerland

{maria.korosteleva, olga.sorkine}@inf.ethz.ch

{tkesdogan, kollerja, yuhzhang}@student.ethz.ch

<sup>2</sup> TU Dortmund University, Otto-Hahn-Strasse 16, 44227 Dortmund, Germany  
{fabian.kemper, stephan.wenninger, mario.botsch}@tu-dortmund.de

**Abstract.** Recent research interest in learning-based processing of garments, from virtual fitting to generation and reconstruction, stumbles on a scarcity of high-quality public data in the domain. We contribute to resolving this need by presenting the first large-scale synthetic dataset of 3D made-to-measure garments with sewing patterns, as well as its generation pipeline. GarmentCodeData contains 115,000 data points that cover a variety of designs in many common garment categories: tops, shirts, dresses, jumpsuits, skirts, pants, etc., fitted to a variety of body shapes sampled from a custom statistical body model based on CAESAR [28], as well as a standard reference body shape, applying three different textile materials. To enable the creation of datasets of such complexity, we introduce a set of algorithms for automatically taking tailor’s measures on sampled body shapes, sampling strategies for sewing pattern design, and propose an automatic, open-source 3D garment draping pipeline based on a fast XPBD simulator [22], while contributing several solutions for collision resolution and drape correctness to enable scalability.

**Keywords:** Garments data · Dataset generation · Cloth reconstruction

## 1 Introduction

Reconstructing CAD representation of garments – sewing patterns – is an increasingly relevant problem. The sewing pattern is a key garment representation that not only connects design with production, but is also essential to the fitting process, be it physical or virtual, allowing detailed garment editing and application of various appearance models and fabrics. The advancements of generative models, virtual fitting rooms and reconstruction of photorealistic full-body avatars all require scalable solutions for obtaining garment rest shapes from raw inputs – images and 3D models. Learning-based solutions have started to appear [13,18], but the progress is hindered by the lack of open, large-scale datasets of 3D garments labeled with sewing patterns.



**Fig. 1:** GarmentCodeData samples covering a variety of designs and body shapes.

Gathering datasets of the required scale and diversity from real-world data faces major challenges w.r.t. copyright, paywalls and general unavailability of garment-pattern pairs. Synthetic datasets [5, 12, 18] provide an alternative that does not face these issues, while offering high-quality data and predictable coverage of the domain. However, generating a reasonably representative set is still a difficult problem. In this work we address the problem of design and body shape variation in synthetic garment generation, creating the first large-scale open dataset of made-to-measure garments with our generation pipeline.

Our design sampling process is based on the parametric modeling approach of GarmentCode [14], with the addition of sampling probabilities to the design variation parameters and the editing of the design space itself to reduce the chances of randomly creating invalid samples (e.g., skirts twice as long as the height of a person). Through consultations with a professional fashion designer, we have also improved the quality of the sewing patterns originally implemented in GarmentCode and expanded the set to include new elements, e.g. hoods. As for body shape variation, we utilize a statistical body model and introduce a set of algorithms for automatically evaluating body measurements from body shape samples. The dataset is obtained by random sampling of body shapes and design parameters to generate sewing patterns, which are then draped on the bodies by our automatic draping pipeline, based on the extended position-based dynamics (XPBD) simulator [22, 23]. To achieve the required quality of drapes across body shapes and designs, we introduce several solutions to combat the issues that arise, including cloth-cloth and body-cloth constraints that help to resolve collisions in the initial sewing pattern placement. For each generation step, we implement quality control conditions to exclude bad samples from the data.

The final dataset consists of 115,000 design samples, each fitted to a random body shape and a canonical neutral body model (gender-neutral statistical average), labeled with a ground truth sewing pattern and a 3D draped mesh, alongside segmentation labels, a UV map corresponding to initial panel shapes, body measurements, design parameters and basic renders for visualization. De-

signs span a wide variety of lower-body (skirts and pants of different styles), upper-body (tops, shirts, blouses), and full-body (dresses, jumpsuits) garments.

Our work presents the following contributions:

- Controllable parametric design space of quality sewing patterns, suitable for garment design sampling.
- Fit-aware automatic body measurement method.
- Open-source pipeline for generating 3D garment models from sewing patterns at scale.
- The first open, large-scale dataset of made-to-measure garments. It covers a variety of complex designs and body shapes, and provides 3D models and corresponding sewing patterns with each data point.

The links to the dataset and available code are shared on the project page.<sup>3</sup>

## 2 Related work

**3D garment datasets.** Existing datasets of 3D garments or clothed humans can be put into three categories: those based on 3D scans of physical garments [2, 10, 15, 21, 30, 36, 38, 40], synthetic [5, 11, 12, 18, 24, 27, 32, 34, 35], and datasets with garments manually created by professional designers [6, 41]. Datasets of real garments capture their realistic behavior, but are expensive to collect, often contain noise and face copyright issues when sharing, e.g. [2], which contains samples from commercial repositories. The design variation they provide is also limited, often focusing on common clothing pieces – t-shirts, jeans, hoodies – not capturing the diversity of designs available in the wild. To this day, Deep Fashion3D [40] is the most diverse dataset of captured garments, but it contains only 563 items. These datasets are *unusable* for sewing pattern reconstruction since they do not provide the sewing patterns corresponding to the 3D garments.

The garments created in virtual CAD by professional designers in [6, 41] represent clothing of high complexity and allow seamless integration of these garments into generation pipelines for synthetic images. Unfortunately, this approach is hard to scale due to the labor resources for each piece. This is also a reason why a diverse set of body shapes is hard to achieve in this approach – due to the need to manually fit a garment to a desired body shape. Garment sewing patterns should be available for these datasets in theory, but de facto they are shared in proprietary file formats that cannot be processed outside of the corresponding CAD systems.

In this work, we opt for a synthetic dataset. This is a scalable solution with fully controlled output data distribution and high-quality data samples, well-aligned across the dataset, and with exact ground truth labels. Several synthetic garment datasets exist, but most do not supply sewing pattern labels at all [5, 11, 27, 39], or cover only a small selection of garment styles [24, 32, 34, 35]. Korosteleva and Lee [12] and Liu et al. [18] are the closest works to our new dataset that provide sewing patterns across a wide selection of garment designs, with the latter [18] using the former [12] as the backbone. Korosteleva and

---

<sup>3</sup> <https://igl.ethz.ch/projects/GarmentCodeData/>

Lee [12] laid significant groundwork for sewing pattern-based generation, but their dataset lacks in complexity of generated garments and does not provide variation of body shapes. Their generation pipeline is based on a commercial CPU-based physics simulator, limiting its adoption. Inspired by this work, we aim to address its issues. We introduce made-to-measure design samples fitted to a variety of body shapes, increase the complexity of garment designs, add small variations to poses for robustness, and implement the generation pipeline using an open-source GPU-based simulator [22], allowing for easier adaptation and scalability. We provide a significantly larger dataset consisting of 115,000 designs compared to 23,500 in Korosteleva and Lee [12].

**Automatic body measurements.** To accurately generate clothing for a specific body shape, the fitting of the sewing patterns should be informed by anthropometric measurements on the body. Various methods for body shape measurement have been proposed: Pujades et al. [26] use a VR tracking system and ask the user to place a VR controller on various locations on their body, thereby only requiring minimal data input. Given a 3D scan of a person, body measurements can be directly inferred from the corresponding 3D point cloud [20]. Registering a statistical template model to 3D scans allows to define landmarks and piecewise-linear curves (polylines) on the template topology, which can be used to guide the measurement algorithms [4, 17, 25]. From these template-based methods, models for regressing anthropometric measurements directly from (silhouette) images can be trained [29]. Typically, axis-aligned planes and Euclidean as well as geodesic distances between landmarks are used to provide a set of measurements. Our body shape measurements are taken in a similar fashion, but we additionally optimize the predefined polylines to adjust them to the given body.

## 3 Data Generation Pipeline

### 3.1 Overview

The data generation process consists of three steps: body shape sampling, sewing pattern sampling, and draping. We sample 5,000 body shapes from a custom statistical body model, and then automatically evaluate the body measurements for each sample with a collection of algorithms designed for different types of body measurements (Sec. 3.3). Apart from body shape variations, we include small variations in the arm angles to increase the robustness of the models trained on our dataset to slight variations in A-poses in the inputs. Next, sewing pattern designs are sampled (Sec. 3.2). We carefully adjust the design space to ensure a diverse yet reasonable design distribution in the final dataset. Every design sample is fitted to the canonical neutral body model (statistical average across genders) and to one of the randomly sampled body shapes based on the evaluated measurements, which provides a ground truth design reference in canonical shape for every sample. The generated sewing patterns are then draped on top of their corresponding body models using our custom draping pipeline, implemented using Warp [22], a GPU-based XPBD simulator.

### 3.2 Design space preparation and sampling

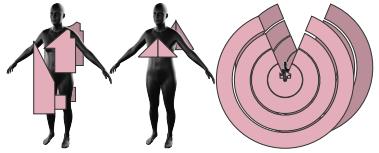
GarmentCode [14] provides a rich design space of garments. However, a naive uniform sampling of its design parameters results in numerous non-representative design samples, e.g., skirts or sleeve elements longer than the body dimensions, or unsuitable garment appearance, see Fig. 2. We update the design space to ensure the sampling of interesting but reasonable garments.

**Aligning the design parameters.** First, we align the design variations with body measurements and with each other, e.g., lower body elements' lengths are kept within the leg length, and cuff elements' lengths are kept smaller than the length of corresponding sleeve or pant elements.

**Sampling probability.** Secondly, we notice that usually, most of the design elements in daily wear follow a common style convention, with only some elements varying in style to add uniqueness to a particular piece. We implement this idea by introducing sampling probabilities to the default value of each design parameter  $P_i$ . We flip a coin with a probability  $p_i$  to draw the default value of  $P_i$ . If the default value is not selected, we draw a non-default value from the corresponding parameter range uniformly. The default values and their probabilities are defined manually to ensure reasonable and interesting designs and more common features to appear more often in the data samples. By adjusting the default value probabilities for the top-level garment structure, we set up roughly a third of the samples to be upper-body garments, another third – lower-body garments, and the rest to be full-body garments (dresses and jumpsuits). We define a strong preference for symmetric garments by setting  $p_i = 0.8$  for the parameter that controls left-right asymmetry in necklines and sleeves.

**Updating the template patterns.** Finally, with the help of a professional fashion designer, we improve the default sewing patterns of the fitted garment elements (bodice, pencil skirt, pants) provided with GarmentCode in terms of the fit quality and pattern-making conventions alignment. We also add a hood element to represent hoodies in our dataset.

Another important modeling consideration is the initial placement of the pattern panels (representing pieces of fabric) around the body, which heavily influences simulation quality (see Sec. 4.2). We start by placing the highest-located garment elements relative to body locations, e.g., the top of an upper-body garment is placed at the neck level, tops of waistbands or lower-body garments are placed at the waist level when upper-body elements are not present, sleeves are rotated to align with arm pose angle. The remaining pieces are placed based on stitching information in relation to already placed components in order to reduce the chances of inter-panel collisions, e.g., lower-body garment elements are placed below a waistband, if present, and cuff elements are placed below corresponding sleeve or pant components.



**Fig. 2:** Samples without guidance.

We also expand the functionality of GarmentCode by introducing the ability to add labels to panels or individual panel edges, which facilitates the simulation process by providing references for attachment constraints and body part segmentation, as described below.

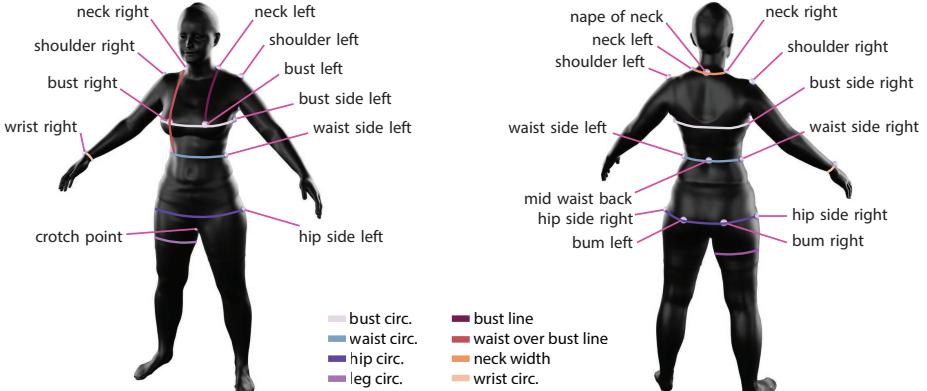
**Quality control.** Incompatibility of some parameter combinations is a common pitfall of complex parametric modeling systems, ours included. After a design sample is drawn, we employ additional quality control procedures to remove inadequate samples that could not be prevented with the measures above. We exclude garment samples that contain self-intersecting panels, full-body garments whose total length exceeds the floor length (distance from the base of the neck to the bottom of the feet) for a given body shape, empty patterns, garments that only consist of a belt, and lower garments with a high chance of sliding off. The latter include skirts with the upper circumference wider than the body trunk, and heavy (wide or long) flare skirts with a very thin or absent belt. The “heaviness” is defined heuristically based on our experiments of simulating flare skirts with different design parameters.

### 3.3 Automatic body measurements

**Body shape sampling.** We generate a statistical human shape model by registering a template mesh to the European subset of the CAESAR database [28], consisting of 1700 3D scans annotated with anthropometric measurements. To register our template to the database, we follow the common template fitting approach [1, 7, 19], yielding 1700 surface meshes with the same topology and pose. We apply principal component analysis (PCA) to the registered meshes to obtain a low-dimensional shape space, which we sample to generate 5000 body shapes (see details in the supplemental material).

**Body measurements.** We automatically compute the body measurements required to generate garment patterns for the sampled body shapes. These serve as input data to fit the updated GarmentCode [14] patterns to the bodies. One of the main rules in garment fit is balance: the main feature lines should look straight, e.g., the side stitch is vertical, and the waistline is parallel to the ground, making the wearer’s posture look straight and upright. To support this goal we take the body measurements directly along the balance lines, ensuring correctly fitted sewing patterns as a result. Hence, most measurements are taken parallel to the ground or vertical direction. To this end, we select predefined landmarks (see Fig. 3) and curves on the template model to guide the measurement algorithms. We take a total of 25 measurements on the body shapes, grouped into different classes of measurement types.

To measure the *bust circumference*, *hip circumference*, *waist circumference* and *leg circumference*, we use an edge loop defined on the template mesh as an initial guess. We then compute a measurement plane that is parallel to the ground plane and minimizes the sum of squared distances to the vertices of this edge loop (see Fig. 3). The convex hull of the intersection between the mesh and the measurement plane (a planar polyline) provides the required circumference.



**Fig. 3:** Overview of the body landmarks and measurements.

This mimics a tailor placing a tape measure around the body. It is possible for the arms to cross the measurement plane, so if multiple closed polylines are detected, the largest circumference is selected.

Due to some amount of tangential drift in the template fitting process, the computed measurement plane may be slightly misplaced. To remedy this, we shift the plane in its up-direction in the range  $[-2 \text{ cm}, 2 \text{ cm}]$  in 5 mm steps and measure the resulting convex hull length. The final measurement value is defined by the local maximum of these values for the *bust circumference* and *hip circumference* and the local minimum for the *waist circumference*. Subsequently, we recalculate the landmark positions for the *waist side* (left and right), *mid waist back*, *bust* (left and right), *bust side* (left and right), *hip side* (left and right) and *bum* (left and right), so that these landmarks are the closest point on the intersection polyline. For the *leg circumference* measurement, the measurement plane is shifted down until both legs are separated in this plane. We measure both leg circumferences and record the minimum value of both legs.

For the *back width*, the *hip back width* and the *waist back width* measurements, only the back-facing part of the circumference curve is relevant. We use the corresponding left and right landmarks on the body to define the coronal plane and measure the back-facing part of the convex hull described above.

We use similar principles to compute the remaining measurements, as detailed in the supplemental material.

### 3.4 Draping

We choose Warp [22], specifically its XPBD implementation, as our backbone simulator, since it is one of very few open-source and GPU-based simulators, providing the speed and scalability necessary for data generation. However, Warp includes only basic functionality, not specifically targeted for garment draping, so we have to solve multiple challenges to successfully simulate randomized gar-

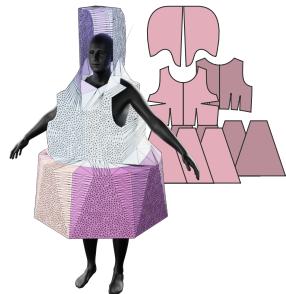
ment designs at scale.

**Draping setup.** Sampled sewing patterns are stored in a JSON-based format [12] that describes a sewing pattern as a structure. To prepare it for simulation, we generate the corresponding mesh using a box mesh paradigm: we compute a single connected mesh from the pattern panels by connecting vertices of edge pairs participating in a stitch (Fig. 4). The details of this process are provided in the supplemental material.

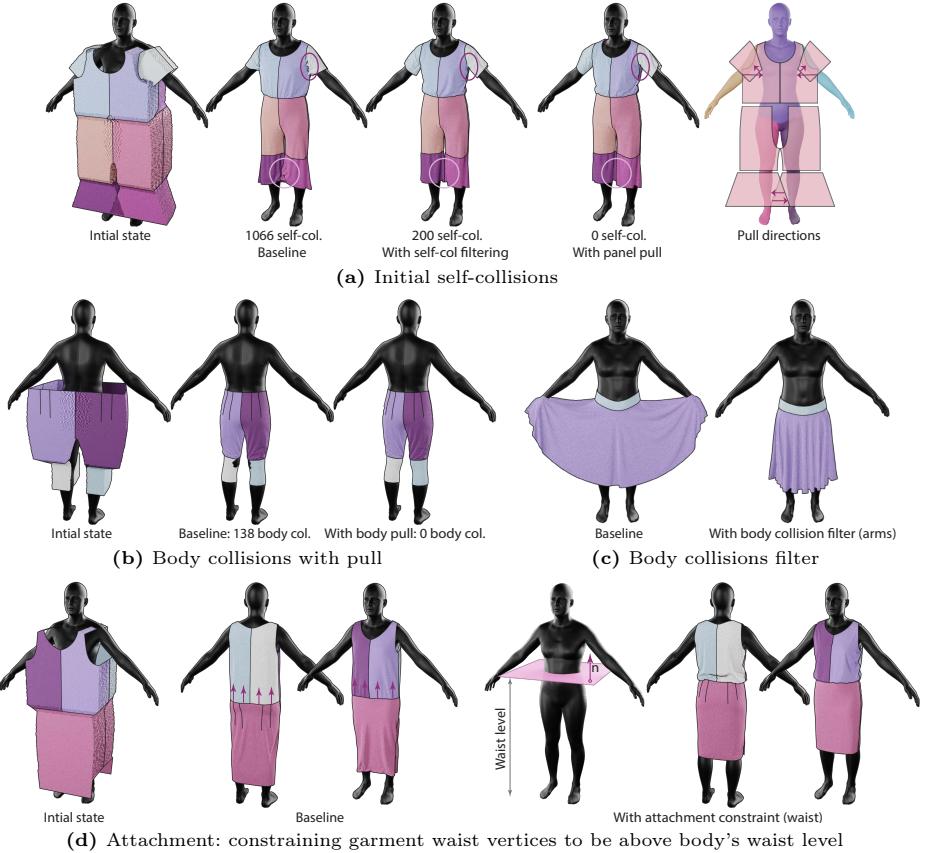
We set up the simulation of the box mesh by supplying the original mesh edge lengths as a reference, collapsing the edges elongated due to stitch vertices merging and effectively stitching the panels together. We also implement the point-triangle and edge-edge self-collision prevention algorithms for XPBD [9, 16], which are not available in Warp. Similarly to Korosteleva and Lee [12], we allow the first few frames of the simulation to be gravity-free for the benefit of the stitching process. We also use similar conditions on the static equilibrium. If a small enough percentage of mesh vertices (1.5%) change their location between the frames by a magnitude greater than a specified threshold (0.04 cm), the mesh is considered static. With the addition of material properties selection, this constitutes the simplest draping setup. However, due to the diversity of the pattern designs and the heuristic initial placement of the pattern panels, many examples would be draped incorrectly (Sec. 3.4). Below we describe our approach to more robust, automatic draping.

**Resolving initial collisions.** One of the major challenges in draping is the collision-free initial state requirement. Collision handling methods are typically designed to prevent new collisions from occurring, but cannot resolve collisions that already exist in the initial state. In our scenario, the designs are varied automatically, and the panel placement is only approximately assigned, so both self- and body-collisions are inevitable. Surprisingly little work addresses this issue when applied to cloth simulation. Existing approaches either tackle special cases irrelevant to us [3, 8] or use optimization-based solutions [33, 37] that complicate the system and significantly slow the draping. We propose a selection of methods that efficiently handle different types of initial collisions using XPBD-friendly techniques.

We note that the standard self-collision resolution methods aim to preserve the existing relative state of the garment mesh, counteracting attempts to resolve self-collisions by other means. We propose to perform global collision detection by finding the mesh edges that intersect non-adjacent faces using ray-intersect checks and then excluding these mesh edges and their adjacent vertices from the self-collision prevention mechanism, effectively allowing these elements to pass through the garment mesh freely. This approach already allows for successful collision-free draping of box meshes akin to spring-based pattern stitching, but some artifacts remain, as demonstrated in Fig. 5a.



**Fig. 4:** Box mesh



**Fig. 5:** Resolving simulation issues. *Pull* refers to collision resolution constraints.

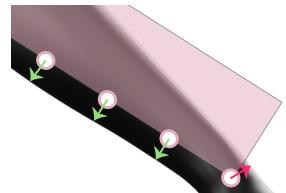
For the initial self-collisions, the hardest problem is determining the direction of pulling on the mesh vertices to resolve the collisions. We notice that most of the overlaps between the panels of the same pattern occur in the areas between body parts, where little space is available: the trunk and the arms, or between the legs. Our idea is to introduce an additional constraint that pulls apart the intersecting geometry towards the body parts they semantically correspond to, thereby eventually resolving the collisions (Fig. 5a). This constraint is applied to vertices of edges that intersect the garment mesh faces belonging to a different body part compared to the edge in question.

To determine panel-body part correspondence, we utilize the fact that we have access to the pattern modeling process from GarmentCode [14] and that all the body meshes share the same topology. First, we define a simple segmentation mask on the body indicating each limb (left arm, right arm, left leg, right leg) and trunk, as shown in Fig. 5a (right). In addition, we define combinations of limbs (both arms, both legs) for convenience. We then match every panel and its

corresponding mesh vertices to one of the indicated body parts by calculating the body part closest to most of the vertices of that panel or using the panel label if provided. Note that only the pant panels are matched with the right and left legs separately, while skirts are matched to both legs as a single unit.

Mesh-body segmentation labels allow us to define collision filters: it is unlikely that the lower body garments collide with the arms, so to prevent wide skirts from being stuck in the arm area, as in the example in Fig. 5c, we ignore the collisions between arms on the body mesh and the parts of the garments mesh associated with the legs, allowing skirts to fall through the arms freely. We use the same mechanism to discard collisions with internal geometry pieces of the body (mouth and eyeballs).

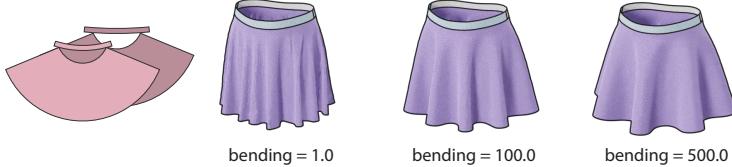
Finally, we apply a constraint that pulls the vertices of the garment mesh found inside the body mesh towards the closest point at the surface of the body, see Fig. 6. Granted, this simple approach may match the garment point to the body surface incorrectly resulting in worsening the body-garment collisions (Fig. 9g), but we found it helpful on average in our design drapes, see Sec. 4.3. We assume that thanks to reasonable placement initialization, most of the garment vertices erroneously placed inside the body are not too far from the correct side of the body.



**Fig. 6:** Body-cloth constraint. Red arrow indicates an erroneous match.

**Attachment constraints.** Errors in panel placement not only cause collision issues, but sometimes lead to incorrect placement of garment elements on the body in the final simulation. We employ attachment constraints on the lower-body garments (pants, skirts) to force their placement on the waist area, to pull strapless tops up towards the underarm level, and to pull neckline vertices closer to the neck, which effectively places the shoulder area of the garment at the correct location. The attachment constraints are released after a set amount of frames to allow position adjustment towards the natural location (e.g., low-rise bottoms should be placed lower than the waist).

To simplify attachment constraint specification, we propose what we call a *half-space attachment constraint*, where instead of specifying a desired location for each vertex participating in the constraint, we specify the target half-space, defined by a plane and its normal direction. With this formulation, the waist constraint can be expressed as simply having the vertical coordinate of the affected vertices to be higher than the waistline of the body shape, which is agnostic to the mesh topology of the body model, see Fig. 5d. We use the known body measurements information to define the half-spaces for the attachment constraints: waist level for lower garments, underarm level for strapless tops, both with the vertical normal direction, and half of the neck width to the left and to the right of the head for collar vertices, with the normal pointing towards the center of the body along the horizontal axis.



**Fig. 7:** Simulation materials used in GarmentCodeData.

**Material properties.** We fix the stiffness across all samples to have minimal stretching. To add variation in materials, we use different levels of fabric softness by varying the bending properties (Fig. 7). The values are chosen manually to cover typical behavior of fabrics in daily wear. Materials are assigned to samples randomly, each covering approximately a third of the data points.

**Quality control.** Due to the significant diversity in design samples, some may not drape correctly and hence need to be excluded from the dataset to avoid confusing trained models with unrealistic data points. We limit the simulation of each sample by time (5 minutes) and number of frames (2400). If the static equilibrium is not achieved, e.g., a lower garment has slid off and keeps falling, the example is marked as a fail. We also filter out examples with a high number of cloth-body and self-collisions, counted as the number of garment mesh edges intersecting the body mesh or the garment mesh itself, respectively. The success rate of simulation, accounting for failures of mesh generation and quality control, is about 72%. We analyze the failure cases in Sec. 4.2.

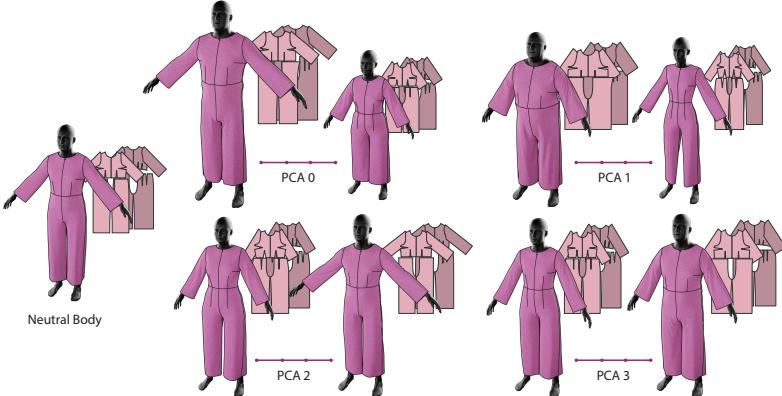
**Generation speed.** We create the dataset in batches of 5,000 design samples each. The batches are independent of each other and can be processed in parallel. Sewing patterns that fail the quality checks are discarded, and the generation continues until the desired number of correct samples is reached. It takes about 7 hours to obtain a set of 5,000 designs, fitted for both random body shapes and the neutral body, making it 10,000 sewing patterns in total that passed the quality checks when running on one core of AMD EPYC 9654 CPU.

Each batch is then processed in the draping step, running a simulation on the neutral body and random body shapes in parallel, 5,000 drapes for each. The body mesh consists of 47.5k triangles and each garment contains about 30k triangles on average. When running on NVIDIA GeForce RTX 3090, each sample takes 30 seconds to simulate on average, amounting to about 44 hours for 5,000 drapes, with about 2 hours overhead for renders and other processing, resulting in approximately 3,200 final designs that pass the quality check per batch.

## 4 Analysis

### 4.1 Body measurements quality

In order to evaluate the accuracy of our automatic measurements, we evaluate those measurements that have a counterpart in the CAESAR data set. These



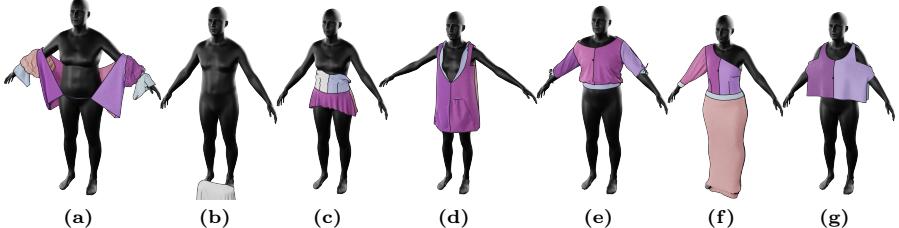
**Fig. 8:** Design transfer across extreme PCA values based on automatic measurements.

measurements have been manually taken by tailors and are assumed to be the ground truth in our evaluation. The measurements are on average overestimated by 2.81 cm for the *bust circumference*, 1.99 cm for the *hip circumference* and 3.49 cm for the *leg circumference*. High errors arise due to possible tangential drift (bust measurement) and when the scanned individuals are wearing wide, non-skin-tight pants (leg circumference). Overestimations in our measurements may also arise due to the missing compression of soft tissue when measuring a static mesh. The *shoulder width* is underestimated on average by 4.57 cm our method, as we measure on top of the shoulder ball joint, where the garment shoulder line is typically located. The ground truth measurement is taken on the left and right sides of the deltoid muscles. The overestimation of the circumference measurements does not have a negative impact on the fitting of the garments in our experiments.

**Garment fitting across body shapes.** To showcase the success of our automatic measurement-taking algorithms, we retarget a fitted jumpsuit pattern to a selection of bodies corresponding to the most extreme PCA values present in the body shape samples (Fig. 8). The fitted jumpsuit sewing pattern covers all of the body curves on the body trunk and hence is a suitable design to evaluate the quality of tight fit. In all the examples used, the garments follow the body shape smoothly without excessive wrinkling or balance line distortions, indicating correct fit. The quality of retargeting is almost surprisingly good for heuristic-based body measurements and parametric sewing patterns.

## 4.2 Simulation failure cases

Fig. 9 showcases typical failures of the simulation process. Figs. 9a to 9c show sliding errors, occurring when the garment is bulky and heavy, while the sewing pattern design does not allow for proper grip on the given body shape. For example, the low-rise circle skirt in Fig. 9b falls down due to low body curvature



**Fig. 9:** Simulation failure cases.

**Table 1:** Evaluation of collision resolution constraints (*pull*) for drapes on neutral and random body shapes.  $k^{th}$  stands for  $k^{th}$  percentile.

	#Body collisions			#Self-collisions		
	Median	80 <sup>th</sup>	95 <sup>th</sup>	Median	80 <sup>th</sup>	95 <sup>th</sup>
neutral w/o pull	0	0	82.8	10	162	747
neutral with pull	0	0	52.2	0	104	423
random w/o pull	0	24	141	6	200	535
random with pull	0	10	71.2	0	45	346

and material stretch. Such designs are likely to be impractical in real life context, so we consider these failures as legitimate. Fig. 9d demonstrates heavy self-intersections of a skirt element with a long upper-body piece, failing to preserve a non-colliding state during the application of attachment constraints. While this example reveals weaknesses in collision resolution, one could argue that this design is unrealistic to begin with, since such a top would normally be disconnected from the bottom and flow over it, or the skirt would not be fitted to sit on the waist, but rather hang at the bottom in a dropped waistline style. Figs. 9e and 9f are dressing failures, where our method failed to resolve body-cloth misalignment, leaving cuffs sticking out (Fig. 9e) or skirt caught in the feet (Fig. 9f). Fig. 9g is a similar case and a likely result of an erroneously applied body-cloth collision constraint, when the wide shirt trunk has initial intersections with the arms after draping, and the internal parts are incorrectly pulled towards the top of the arm. These failure cases demonstrate limitations of hand-crafted design spaces, as well as a need for better body-cloth collision resolution due to initial pattern placement.

### 4.3 Simulation pipeline ablation study

As Sec. 3.4 demonstrates, attachment constraints and body collision filters show qualitative improvements for cases they are designed for, hence the need for their use is self-evident. However, the proposed constraint for body-cloth collision resolution has known failure cases when vertices are matched to the wrong side of the body. Hence, we set up an experiment where we run a simulation on the same dataset of 100 samples with and without additional collision constraints,

reporting the statistics for body-cloth and, for completeness, cloth-cloth collisions across the simulated samples (including failure cases). Tab. 1 clearly shows the decrease in the number of collisions when constraints are enabled, showing the usefulness of the constraints on average.

## 5 Discussion and limitations

We presented a generation pipeline for sampling 3D made-to-measure garments with ground-truth sewing patterns, as well as GarmentCodeData – a dataset of 115,000 3D garments of different designs fitted to a variety of body shapes. Although we aimed to create a large-scale dataset suitable for a sewing pattern reconstruction task, we envision other potential applications. The proposed probability-based design space control allows for its easy adjustment and creation of specialized design sets, for example, specific garment types (workwear, uniforms, etc.). The availability of various designs fits to a neutral body shape in the dataset opens up applications for design retargeting. Using the pipeline for neural cloth simulation research could also be an interesting direction to explore.

**Limitations.** Our work improves on some of the limitations of existing works, but there is still ample room for creating truly comprehensive datasets. First, we inherit some limitations of the GarmentCode [14] patterns, addressing which would require additional effort put into the draping pipeline: lack of fabric layers, sharp pleats, hardware (e.g., zippers, buttons or hooks), the ability to represent full outfits. Our take on fabric material distribution is rather minimalistic, and further effort is needed to model materials according to real-world fabric properties. As discussed in Sec. 4.2, some of the simulation errors we encounter are due to issues with initial body-pattern alignment and methods for body-cloth collision resolution, which require further research. As for body measurements, our approach may result in inaccurate measurements due to errors in surface fit, especially on extreme body shapes. Such errors may lead to the topology-based initial guess being far off, and the correct horizontal plane may not be found within the given optimization bounds.

**Ethical considerations.** While CAESAR [28] is one of the best collections of 3D human body shape data, which is of good quality and responsibly collected, it mainly covers the body shapes of healthy European and North American adults, hence the statistical body model is biased towards these body shapes and would have trouble representing geographic variations in body sizes, people with disabilities, teens and children, transferring this bias into the 3D garment samples. Due to the price point on the full dataset, we only use the European subset of CAESAR, which further worsens the problem. The unavailability of open, unbiased 3D human datasets is a critical issue that propagates biases into many downstream applications.

Our work required a sizable amount of computational power to generate the data. However, the server infrastructure we use is carbon neutral, and is running fully on renewable energy [31], which minimizes the climate impact of this work.

**Acknowledgments.** This work was supported by the European Research Council (ERC) under the European Union’s Horizon 2020 Research and Innovation Programme (ERC Consolidator Grant, agreement No. 101003104, MYCLOTH). Stephan Wenninger has been funded by “Stiftung Innovation in der Hochschullehre” through the project “Hybrid Learning Center” (FBM2020-EA-690-01130). We thank Ami Beuret for his help in refactoring GarmentCode codebase. We are grateful to Jana Schuricht for her professional consultations on patternmaking and to members of IGL and GGG for their continuous support and cheer throughout this project.

## References

1. Achenbach, J., Waltemate, T., Latoschik, M.E., Botsch, M.: Fast generation of realistic virtual humans. In: Proc. of ACM Symposium on Virtual Reality Software and Technology. pp. 12:1–12:10 (2017)
2. Antić, D., Tiwari, G., Ozcomlekci, B., Marin, R., Pons-Moll, G.: CloSe: A 3D Clothing Segmentation Dataset and Model. In: International Conference on 3D Vision (3DV) (1 2024), <http://arxiv.org/abs/2401.12051>, <https://virtualhumans.mpi-inf.mpg.de/close3dv24/>
3. Baraff, D., Witkin, A., Kass, M.: Untangling Cloth. ACM Trans. Graph. **22**(3), 862–870 (2003)
4. Bartol, K., Bojanic, D., Petković, T., Peharec, S., Pribanić, T.: Linear Regression vs. Deep Learning: A Simple Yet Effective Baseline for Human Body Measurement. Sensors **22**(5) (3 2022). <https://doi.org/10.3390/s22051885>
5. Bertiche, H., Madadi, M., Escalera, S.: CLOTH3D: Clothed 3D Humans. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) European Conference on Computer Vision. pp. 344–359. Springer International Publishing, Virtual (12 2020). [https://doi.org/10.1007/978-3-030-58565-5\\_21](https://doi.org/10.1007/978-3-030-58565-5_21), <https://arxiv.org/abs/1912.02792>, [http://link.springer.com/10.1007/978-3-030-58565-5\\_21](http://link.springer.com/10.1007/978-3-030-58565-5_21)
6. Black, M.J., Patel, P., Tesch, J., Yang, J.: BEDLAM: A Synthetic Dataset of Bodies Exhibiting Detailed Lifelike Animated Motion. In: Proceedings IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8726–8737 (2023), <https://bedlam.is.tue.mpg.de/>.
7. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Dynamic 2D/3D registration. In: Eurographics Tutorials. pp. 1–17 (2014)
8. Buffet, T., Rohmer, D., Barthe, L., Boissieux, L., Cani, M.P.: Implicit untangling: A robust solution for modeling layered clothing. ACM Transactions on Graphics **38**(4) (7 2019). <https://doi.org/10.1145/3306346.3323010>
9. Cincotti, C.: Cloth self-collisions (2022), <https://carmencincotti.com/2022-11-21/cloth-self-collisions>
10. Ho, H.I., Xue, L., Song, J., Hilliges, O.: Learning Locally Editable Virtual Humans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023), <https://custom-humans.github.io/>
11. Jiang, B., Zhang, J., Hong, Y., Luo, J., Liu, L., Bao, H.: BCNet: Learning Body and Cloth Shape from a Single Image. In: Computer Vision – ECCV 2020. pp. 18–35. Springer International Publishing, Cham (8 2020). [https://doi.org/10.1007/978-3-030-58565-5\\_2](https://doi.org/10.1007/978-3-030-58565-5_2), <https://github.com/jby1993/BCNet>

12. Korosteleva, M., Lee, S.H.: Generating Datasets of 3D Garments with Sewing Patterns. In: Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks. pp. 1–10 (6 2021). <https://doi.org/10.5281/ZENODO.5267549>, <https://github.com/maria-korosteleva/Garment-Pattern-Generator>
13. Korosteleva, M., Lee, S.H.: NeuralTailor: Reconstructing Sewing Pattern Structures from 3D Point Clouds of Garments. ACM Trans. on Graph. **1**(1) (2022), <http://arxiv.org/abs/2201.13063>
14. Korosteleva, M., Sorkine-Hornung, O.: GarmentCode: Programming Parametric Sewing Patterns. ACM Transactions on Graphics **42**(6) (12 2023). <https://doi.org/10.1145/3618351>
15. Lal Bhatnagar, B., Tiwari, G., Theobalt, C., Pons-Moll, G.: Multi-Garment Net: Learning to Dress 3D People from Images. In: The IEEE International Conference on Computer Vision (ICCV). pp. 5420–5430. IEEE, Seoul, Republic of Korea (2019)
16. Lewin, C.: Cloth self collision with predictive contacts (2018), <https://api.semanticscholar.org/CorpusID:202704112>
17. Li, X., Li, G., Li, T., Lv, J., Mitrouchev, P.: Remodeling of mannequins based on automatic binding of mesh to anthropometric parameters. Visual Computer (2022). <https://doi.org/10.1007/s00371-022-02738-1>
18. Liu, L., Xu, X., Lin, Z., Liang, J., Yan, S.: Towards Garment Sewing Pattern Reconstruction from a Single Image. ACM Transactions on Graphics (11 2023). <https://doi.org/10.1145/3618319>, <http://arxiv.org/abs/2311.04218>, <http://dx.doi.org/10.1145/3618319>, <https://sewformer.github.io/>, <https://github.com/sail-sg/sewformer>
19. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Transactions on Graphics **34**(6), 248:1–248:16 (2015)
20. Lu, J.M., Wang, M.J.J.: Automated anthropometric data collection using 3D whole body scanners. Expert Systems with Applications **35**(1-2), 407–414 (7 2008). <https://doi.org/10.1016/j.eswa.2007.07.008>
21. Ma, Q., Yang, J., Ranjan, A., Pujades, S., Pons-Moll, G., Tang, S., Black, M.J.: Learning to Dress 3D People in Generative Clothing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6469–6478. IEEE, Virtual (2020), <https://cape.is.tue.mpg.de>.
22. Macklin, M.: Warp: A high-performance python framework for gpu simulation and graphics. <https://github.com/nvidia/warp> (March 2022), nVIDIA GPU Technology Conference (GTC)
23. Macklin, M., Müller, M., Chentanez, N.: XPBD: Position-based simulation of compliant constrained dynamics. In: Proceedings - Motion in Games 2016: 9th International Conference on Motion in Games, MIG 2016. pp. 49–54. Association for Computing Machinery, Inc (10 2016). <https://doi.org/10.1145/2994258.2994272>
24. Narain, R., Samii, A., O'Brien, J.F.: Adaptive anisotropic remeshing for cloth simulation. ACM Transactions on Graphics **31**(6) (11 2012). <https://doi.org/10.1145/2366145.2366171>
25. Nourbakhsh Kaashki, N., Hu, P., Munteanu, A.: Anet: A Deep Neural Network for Automatic 3D Anthropometric Measurement Extraction. IEEE Transactions on Multimedia (2021). <https://doi.org/10.1109/TMM.2021.3132487>
26. Pujades, S., Mohler, B., Thaler, A., Tesch, J., Mahmood, N., Hesse, N., Bülthoff, H.H., Black, M.J.: The Virtual Caliper: Rapid Creation of Metrically Accurate

- Avatars from 3D Measurements. *IEEE Transactions on Visualization and Computer Graphics* **25**(5), 1887–1897 (5 2019). <https://doi.org/10.1109/TVCG.2019.2898748>
- 27. Pumarola, A., Sanchez, J., Choi, G.P.T., Sanfeliu, A., Moreno-Noguer, F.: 3DPeople: Modeling the Geometry of Dressed Humans. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2242–2251. IEEE, Seoul, Republic of Korea (2019), <https://arxiv.org/pdf/1904.04571.pdf>
  - 28. Robinette, K.M., Blackwell, S., Daanen, H., Boehmer, M., Fleming, S.: Civilian american and european surface anthropometry resource (caesar), final report. volume 1: Summary. Tech. rep., Sytronics Inc (2002)
  - 29. Ruiz, N., Bellver, M., Bolkart, T., Arora, A., Lin, M.C., Romero, J., Bala, R.: Human body measurement estimation with adversarial augmentation. In: Proc. of the International Conference on 3D Vision (3DV). pp. 219–230 (2022). <https://doi.org/10.1109/3DV57658.2022.00034>
  - 30. Tiwari, G., Bhatnagar, B.L., Tung, T., Pons-Moll, G.: SIZER: A Dataset and Model for Parsing 3D Clothing and Learning Size Sensitive 3D Clothing (7 2020), <http://arxiv.org/abs/2007.11610>
  - 31. Ulmer, S.: At cscs, energy efficiency is a key priority, even at high performance (Nov 2022), <https://www.cscs.ch/science/computer-science-hpc/2022/at-cscs-energy-efficiency-is-a-key-priority-even-at-high-performance>
  - 32. Vidaurre, R., Santesteban, I., Garces, E., Casas, D.: Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On. *Computer Graphics Forum* **39**(8), 145–156 (9 2020), <http://arxiv.org/abs/2009.04592>, <https://www.youtube.com/watch?v=BFV85KS3Y6c>, <https://www.youtube.com/watch?v=xUEmB-NWyIQ>
  - 33. Volino, P., Magnenat-Thalmann, N.: Resolving Surface Collisions through Intersection Contour Minimization. *ACM Trans. Graph.* **25**(3), 1154–1159 (2006)
  - 34. Wang, H., O'Brien, J.F., Ramamoorthi, R.: Data-Driven Elastic Models for Cloth: Modeling and Measurement. *ACM Transactions on Graphics* **30**(4), 1–12 (7 2011). <https://doi.org/10.1145/2010324.1964966>
  - 35. Wang, T.Y., Ceylan, D., Popović, J., Mitra, N.J.: Learning a shared shape space for multimodal garment design. *ACM Transactions on Graphics* **37**(6), 1–13 (12 2018). <https://doi.org/10.1145/3272127.3275074>
  - 36. Xu, W., Du, W., Xue, H., Li, Y., Ye, R., Wang, Y.F., Lu, C.: ClothPose: A Real-world Benchmark for Visual Analysis of Garment Pose via An Indirect Recording Solution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 58–68 (2023)
  - 37. Ye, J., Ma, G., Jiang, L., Chen, L., Li, J., Xiong, G., Zhang, X., Tang, M.: A Unified Cloth Untangling Framework Through Discrete Collision Detection. *Computer Graphics Forum* **36**(7), 217–228 (10 2017). <https://doi.org/10.1111/cgf.13287>
  - 38. Zhang, C., Pujades, S., Black, M., Pons-Moll, G.: Detailed, accurate, human shape estimation from clothed 3D scan sequences. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. vol. 2017-Janua, pp. 5484–5493. IEEE, Honolulu, Hawaii, USA (2017). <https://doi.org/10.1109/CVPR.2017.582>
  - 39. Zhou, B., Zhou, H., Liang, T., Yu, Q., Zhao, S., Zeng, Y., Lv, J., Luo, S., Wang, Q., Yu, X., Chen, H., Lu, C., Shao, L.: ClothesNet: An Information-Rich 3D Garment Model Repository with Simulated Clothes Environment. In: IEEE/CVF International Conference on Computer Vision (ICCV) (8 2023), <http://arxiv.org/abs/2308.09987>, <https://sites.google.com/view/clothesnet>

40. Zhu, H., Cao, Y., Jin, H., Chen, W., Du, D., Wang, Z., Cui, S., Han, X.: Deep Fashion3D: A Dataset and Benchmark for 3D Garment Reconstruction from Single Images. In: Computer Vision – ECCV. pp. 512–530. arXiv (3 2020), <http://arxiv.org/abs/2003.12753>, <https://kv2000.github.io/2020/03/25/deepFashion3DRevisited/>
41. Zou, X., Han, X., Wong, W.: CLOTH4D: A Dataset for Clothed Human Reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12847–12857 (2023), [www.github.com/AemikaChow/AiDLab-fAshIon-Data](http://www.github.com/AemikaChow/AiDLab-fAshIon-Data), [https://www.youtube.com/watch?v=8Cc\\_kl55bFo](https://www.youtube.com/watch?v=8Cc_kl55bFo), [https://openaccess.thecvf.com/content/CVPR2023/html/Zou\\_CLOTH4D\\_A\\_Dataset\\_for\\_Clothed\\_Human\\_Reconstruction\\_CVPR\\_2023\\_paper.html](https://openaccess.thecvf.com/content/CVPR2023/html/Zou_CLOTH4D_A_Dataset_for_Clothed_Human_Reconstruction_CVPR_2023_paper.html)

## GarmentCodeData: Supplemental Material

### A Body shape sampling

Principal component analysis (PCA) on the registered meshes (Sec. 3.3) gives us a generative model to sample various body shapes. Let  $V$  be the number of vertices in the template model,  $\mathbf{U} \in \mathbb{R}^{3V \times k}$  contain the principal components scaled by the square root of their respective eigenvalues, and  $\bar{\mathbf{u}} \in \mathbb{R}^{3V}$  correspond to the mean vertex positions of the dataset. New shapes can then be sampled by generating a parameter vector  $\mathbf{w} \in \mathbb{R}^k$  and reconstructing new vertex positions  $\mathbf{X} = \mathbf{U}\mathbf{w} + \bar{\mathbf{u}}$ .

We sample 5000 shapes by independently sampling each PCA parameter  $w_i$  following a normal distribution with  $\mu = 0$  and  $\sigma = 0.7$ , such that the probability of generating  $w_i \in (-2.4, 2.4)$  is 99.94 %. After generating 5000 shapes, we export each body model in two different poses. We vary the arm inclination of the model by drawing a random angle from a uniform distribution over  $[-15^\circ, 15^\circ]$  and altering the standard A-pose of the model accordingly. Additionally, we export the same shape in a pose where the feet are slightly more apart from each other (see Fig. 1). The body measurements described in Sec. 3.3 are all taken in the first of the two poses.

### B Taking measurements

Here, we provide the full description of our measurement-taking algorithms, following the principles outlined in Sec. 3.3. We include the ones already described in Sec. 3.3 for completeness.

To measure the *bust circumference*, *hip circumference*, *waist circumference* and *leg circumference*, we use an edge loop defined on the template mesh as an initial guess. We then compute a measurement plane that is parallel to the ground plane and minimizes the sum of squared distances to the vertices of this edge loop (see Fig. 3). The convex hull of the intersection between the mesh and the measurement plane (a planar polyline) provides the required circumference. This mimics a tailor placing a tape measure around the body. It is possible for the arms to cross the measurement plane, so if multiple closed polylines are detected, the largest circumference is selected.

Due to some amount of tangential drift in the template fitting process, the computed measurement plane may be slightly misplaced. To remedy this, we shift the plane in its up-direction in the range  $[-2\text{ cm}, 2\text{ cm}]$  in 5 mm steps and measure the resulting convex hull length. The final measurement value is defined by the local maximum of these values for the *bust circumference* and *hip circumference* and the local minimum for the *waist circumference*. Subsequently, we recalculate the landmark positions for the *waist side* (left and right), *mid waist back*, *bust* (left and right), *bust side* (left and right), *hip side* (left and right) and *bum* (left and right), so that these landmarks are the closest point on the intersection polyline. For the *leg circumference* measurement, the measurement

plane is shifted down until both legs are separated in this plane. We measure both leg circumferences and record the minimum value of both legs.

For the *back width*, the *hip back width* and the *waist back width* measurements, only the back-facing part of the circumference curve is relevant. We use the corresponding left and right landmarks on the body to define the coronal plane and measure the back-facing part of the convex hull described above.

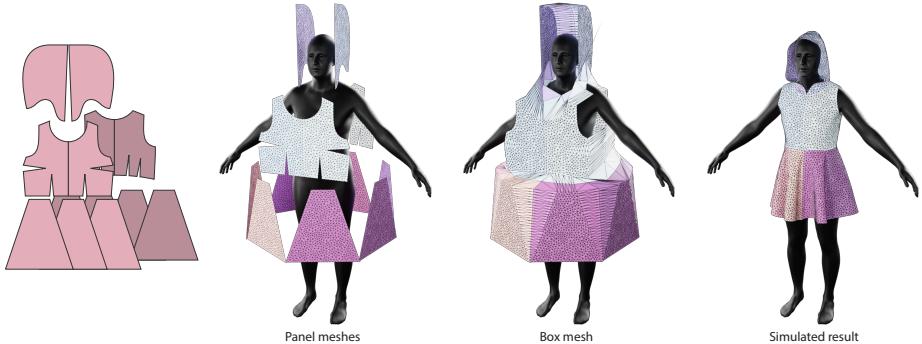
To take the *waist over bust line* measurement, we use a parasagittal plane with the least squares distance to a predefined edge loop on the template mesh. We calculate the polyline of the cut plane and the given mesh and consider a part of this polyline for the final measurement. The intersection plane computed when measuring the *waist circumference* defines the lowest point. The upper boundary is defined by the frontal plane with a minimum distance to the *shoulder* landmarks. We compute the *bust line* similarly, starting at the intersection plane from the *bust circumference* measurement (see Fig. 3).

We use the Euclidean edge loop length to measure the neck width and wrist circumference. For the *armscye depth* measurement and the Euclidean distance between the *bum points*, the *bust points* and the *shoulder width*, we compute the respective distances between the two landmarks for each measurement. The *crotch hip difference*, *hip line* and the *vertical bust line* are measured as distances between two landmarks along the upward axis. For the *height* measurement, the axis-aligned bounding box of the mesh is determined. The final value is the length of the bounding box in the upright dimension. We measure the *head length* as the distance between the neck landmark and the highest point of the head along the upward axis. We calculate the *arm length* as the geodesics distance between the *shoulder* and *wrist* landmarks.

We measure the *arm inclination* and *hip inclination* angles as the opening angle between the median plane of the body and a vector given by the two landmarks. Finally, we calculate the *shoulder inclination* as the angle between the horizontal plane and a vector given by the *neck* and *shoulder* landmarks.

## C Mesh generation

Sewing patterns coming out of GarmentCode [2] are stored in the specialized JSON-based format [1] that represents a sewing pattern as a structure. The pattern is described as a collection of panels, which in turn are specified through the shape of their outline, a looped sequence of directed curved edges connecting vertices on the panel border, with additional specification of the panel 6D placement that locates it in 3D. The specification also contains stitching information as a list of edge pairs connected by a stitch. This representation is unsuitable for simulation, so the first step of the draping process is generating a mesh corresponding to the input sewing pattern. We follow the path of generating a box mesh: generating a fully connected mesh for the whole sewing pattern by directly merging the vertices of the edge pairs connected by a stitch (Fig. C.1). First, we generate vertices on every panel edge by placing them on edge curves according to a chosen mesh resolution (here we aim for a 1 cm average mesh



**Fig. C.1:** Mesh generation process

edge length). Panel edges connected by a stitch may have different lengths (for a gathering/ruching effect), in which case we match the number of vertices on either side by using the maximum number of vertices suggested by the resolution among the two edges, and using this number for both, spacing them equally along the lengths. Once the edge vertices are fixed, we generate internal panel vertices using constrained Delaunay triangulation of CGAL [3]. The vertices are placed in 3D using the panel rotation and translation information from the JSON specification. Triangle normals are set to match the panel normal defined by the counter-clockwise orientation of the edge loop in the specification. Stitches are then processed by merging the matching vertices on either side of a stitch and placing them in the middle between the original locations while keeping the information on the original length of the edges to correctly reduce the stitching edges' lengths during the simulation step.

## References

1. Korosteleva, M., Lee, S.H.: Generating Datasets of 3D Garments with Sewing Patterns. In: Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks. pp. 1–10 (6 2021). <https://doi.org/10.5281/ZENODO.5267549>, <https://github.com/maria-korosteleva/Garment-Pattern-Generator>
2. Korosteleva, M., Sorkine-Hornung, O.: GarmentCode: Programming Parametric Sewing Patterns. ACM Transactions on Graphics **42**(6) (12 2023). <https://doi.org/10.1145/3618351>
3. The CGAL Project: CGAL User and Reference Manual. CGAL Editorial Board, 5.6 edn. (2023), <https://doc.cgal.org/5.6/Manual/packages.html>