Fashion Transfer: Dressing 3D Characters from Stylized Fashion Sketches

Amélie Fondevilla, Damien Rohmer, Stefanie Hahmann, Adrien Bousseau, Marie-Paule Cani

Fashion Transfer : Dressing 3D Characters from Stylized Fashion Sketches

A. Fondevilla[†1], D. Rohmer[2], S. Hahmann[3], A. Bousseau[4], M-P. Cani[2].

[1] IRIT, Université de Toulouse, CNRS, INPT, UPS, UT1, UT2J, [2] LIX, Ecole Polytechnique/CNRS, IP Paris,
[3] Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, [4] Inria, Université Côte d'Azur

Figure 1: *Our approach takes as input a single, annotated fashion drawing and synthesizes a 3D garment of similar style over any existing 3D character. The method handles deep folds and automatically adapts the garment to the morphology and pose of the target, allowing to easily dress human-looking to cartoon-style models.*

Abstract

Fashion design often starts with hand-drawn, expressive sketches that communicate the essence of a garment over idealized human bodies. We propose an approach to automatically dress virtual characters from such input, previously complemented with user-annotations. In contrast to prior work requiring users to draw garments with accurate proportions over each virtual character to be dressed, our method follows a style transfer strategy : the information extracted from a single, annotated fashion sketch can be used to inform the synthesis of one to many new garment(s) with similar style, yet different proportions. In particular, we define the style of a loose garment from its silhouette and folds, which we extract from the drawing. Key to our method is our strategy to extract both shape and repetitive patterns of folds from the 2D input. As our results show, each input sketch can be used to dress a variety of characters of different morphologies, from virtual humans to cartoon-style characters.

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics; Shape modeling;***

1. Introduction

Fashion designers often use sketches to illustrate the look of a garment. However, these sketches rarely depict human figures with accurate anatomical proportions, which prevents the direct 3D reconstruction of the depicted garments to dress virtual characters. We propose to use fashion design sketches as an indication of the *style* of a garment, which we transfer to virtual characters of different anatomy and proportions. Our approach thus differs from existing sketch-based modeling systems that require users to draw garments with accurate proportions over a specific target character [TWB*07, RMSC11].

† afondevilla@laposte.net

Accepted to COMPUTER GRAPHICS *Forum* (7/2021).

Recent work on garment modeling in Computer Graphics [BSBC12, LL14, BSK*16] shows that the factors having the most impact on the visual style of a garment are *proportionality*, which expresses the relative proportions between the garment and the mannequin - such as covered portions of the limbs and body; *fit*, which indicates if a garment is worn tight or loose; *folds patterns*, which directly provide information about the material and physical behaviour of the garment, and *overall shape*, which is well captured by surface normals. The key observation in this work is that these four style characteristics can be directly extracted from fashion design sketches and used to generate 3D garments.

Our method takes as input a fashion sketch with a few user annotations. In particular, we ask users to over-trace the free borders of loose parts, which convey folds; the silhouette of the garment, which conveys surface normals; and skeletal bones of the character, which convey the relative location of the garment with respect to the limbs and body of the character.

The first step of our method consists in extracting geometric information from these annotations. While silhouette lines provide direct knowledge of the surface normals at grazing angle, hemlines suffer from perspective deformation and occlusions. We describe how to reconstruct fold patterns by rectifying the hemline and completing occluded folds using a smoothness assumption. Fold information is then stored as a set of generic pose-independent features, allowing to apply them to a character of different morphology.
In a second step, we create the overall shape of the virtual garment over a target rigged character. We represent garment patches surrounding limbs as generalized cylinders with proportions and silhouette normal features extracted from the sketch. This information being generic, the synthesized shape is not limited to the one depicted on the input sketch, and is used to match target characters with different poses and morphologies, the resulting surface being automatically adjusted to prevent collisions with the body. Garment patches are represented as parametric cubic Bézier patches, which brings a compact representation, parameterized along limbs directions.
Finally, the last step of our method consists in synthesizing folds over the garment. We achieve this by transferring the fold patterns extracted from the sketch, scaled to fit the character's size. Thanks to our patch-based parameterization, loose garments exhibiting deep folds along the limbs can easily be modeled.

Note that this work does not directly target fashion product designers, as we do not integrate real-world fabrication constraints. Targeted users are: (1) Hobbyists wishing to generate new 3D garments for their favorite character (ex. a game character) from an illustration found on the web (eg. a manga dress);

(2) Digital artists/designers in 3D production willing to efficiently visualize how garments depicted in a rough concept-art sketch would appear in 3D. Small details are not required, but the general shape and folds are of primary importance;

(3) 3D modeling artists needing to quickly dress one or several virtual character(s) from some example sketch. Our approach would allow them to quickly get a first virtual garment, to be later refined if needed.

We illustrate the versatility of our method by creating a variety of garments, including dresses, shirts and pants. More importantly, our results were obtained from sketches of various styles, from fashion sketches to cartoon sketches with extreme proportions. Similarly, we transferred the extracted garments to realistic as well as to stylized 3D characters. We further confirmed the usability and efficiency of our sketch-based modeling system through a user study.

2. Related Work

Our work builds on garment sketching methods, algorithms for fold and wrinkle modeling, as well as style and garment transfer approaches.

Garment design. The most common approach for *computer-aided garment design* mimics the traditional fabrication process. The user designs a 2D pattern, chooses seam curves among its borders, and places the pattern pieces around the virtual body. Physically-based simulation is then used to automatically compute the 3D draping of the garment dressing the character's body [VCMT05, WLL*09, UKIG11]. Commercial software, such as Clo3D [Clo], Marvelous Designer [Mar], or Optitex [Opt] employ these 2D pattern-based modeling approaches to create very realistic garments. A more detailed overview on CAD methods for garment design can be found in [LZY10]. 3D-to-pattern interfaces [UKIG11, BSK*16] ease the modeling process by supporting coarse-scale edits in 3D, such as elongation, cutting or garment merging which are propagated to the patterns, while the pre-positioning of cloth patterns can be computed automatically [GFL03]. These approaches however require significant tailoring expertise. Moreover, they require the design or computation of 2D patterns in the garment creation loop, whereas in the case of fashion design sketches, patterns are not readily available.

Sketch-based modeling systems aim at simplifying the modeling process to make it accessible to non-expert users. The challenging problem is to infer the 3D geometry of garments from only one or two 2D sketches representing them, as detailed in different surveys [CSE*16, OSSJ09]. Various geometric cues have been leveraged to constrain this problem, such as known surface orientation along occluding contours [IMT99, NISA07, OSJ11, SKv*14], curve planarity and orthogonality [SKSK09, XCS*14], symmetry [CSMS13], and developability [FBR*17]. Sketch-based modeling systems dedicated to garment modeling often require users to draw over the 2D view of a pre-defined 3D mannequin, which provides additional context about the position of the drawn strokes in the 3D space [WWY03, TCH07, DPS15, RMSC11]. In contrast, a strength of our method is its ability to analyse *pre-existing* sketches which may or may-not explicitly depict the mannequin, and use them to generate garments over 3D characters of different morphology and size. Casting sketch-based garment modeling into a style transfer problem makes our approach robust to the exaggerated proportions often observed in fashion design sketches.

Folds and wrinkles are essential to the appearance of a garment, so reproducing them already received a lot of attention. Jung et al. [JHR*15] reconstruct folds from a sketch by constraining surface orientation along fold lines, assuming that they are drawn

from a grazing viewing angle. Their method aims at exactly reproducing the set of folds represented in the sketch, while we rather aim at generating similar-looking folds over different garments. Physically-based simulations holds the promise of generating realistic folds on arbitrary geometry [RPC*10, WHRO10]. However, reproducing sketched folds using simulation would require solving the difficult inverse problem of jointly recovering the 2D pattern of the garment and the physical properties of its fabric. While Li [LSGV18] propose a first solution to this challenging problem, they assume that the initial fold-free garment is provided as input. Closer to our method, Decaudin et al. [DJW*06] propose a geometric method to synthesize procedural folds in areas where a garment compresses or twists. We follow a similar strategy by generating procedural folds that exhibit the same distribution as the ones depicted on the sketch.

Single-image reconstruction. Our problem relates to computer vision methods that aim at reconstructing a 3D garment from a single input photograph. Recent methods employ deep learning [DDÖ*17, JZH*20] or optimization [YPA*18] to adjust parameters of template garments so that they best fit the input. These methods however heavily rely on physically-based simulation to generate training data or to regularize the optimization, and as such are not suited to design sketches that do not necessarily follow the laws of physics. Wang et al. [WCPM18] propose a set of neural networks to bring sketches, 2D patterns and draped 3D garments into a shared latent space, which allows users to design garments by manipulating any of these modalities. However, their system was trained using synthetic drawings generated from simulated garments, which do not exhibit the variety of poses and exaggerated proportions present in the fashion design sketches we handle.

Style transfer. The seminal work of Hertzmann et al. [HJO*01] on image analogies has inspired a number of methods that seek to transfer the style of an exemplar onto a target image or shape. Representative work includes the transfer of artistic styles to realistic renderings [FJL*16], the transfer of geometric details on furniture while preserving functionality [LKWS16], and the transfer of exaggerated cartoon animation effects to physically-based simulations [DBB*17]. Closer to our application domain, Brouet et al. [BSBC12] propose a method to transfer garments between different characters while preserving their 3D style, captured using proportionality, surface orientation and fit criteria. We take inspiration from this approach, but in contrast we describe how to extract geometric style features from 2D fashion design sketches, and we include fold style among the extracted features. Finally, a number of methods have been proposed to resize garments to fit various body shapes [LYW*10, BSBC12, JYSL19, MWJ12, GRH*12, PMPHB17]. All these methods take an existing garment model as input that is subsequently deformed. We instead directly synthesize a different garment for each target character while preserving the style captured from the input sketch.

3. User input

The input of our algorithm is twofold,as illustrated in Figure 2: Firstly the 3D rigged character on which a 3D garment is to be synthesized. Secondly a pre-existing drawing depicting the garment to be transferred. Even though much progress has been recently done in the vectorization and cleaning of paper drawings [FLB16] and in learning-based modeling from sketches [DAI*18, SBS19], automatic line drawing interpretation is still a challenge. In our implementation as in many similar works [LPL*17, FBR*17], we rather ask the user to annotate the sketch to ease interpretation. This is done as follows:

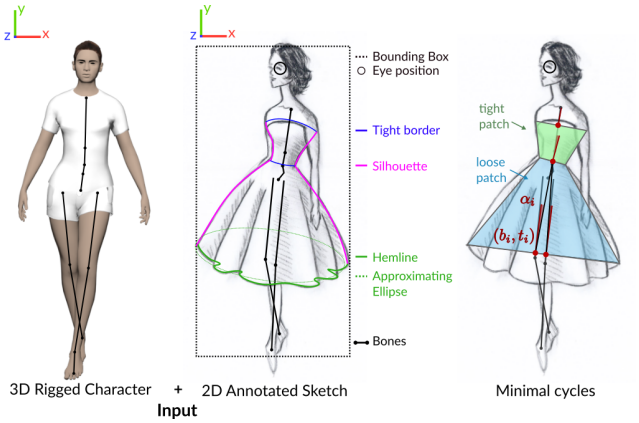


Figure 2: *The input of our algorithm includes a 3D rigged character and a 2D sketch. The user is asked to annotate the sketch with virtual bones in correspondence with the bones of the 3D character. Borders and silhouettes of the garment are also over-sketched. They are used to compute a decomposition of the garment into patches and infer their relative location and orientation with respect to the 2D skeleton.*

Garments contours are over-drawn using Bézier curves of different colors representing silhouette (pink), tight border (blue), and loose border, also called hemline (green). In the case of a loose border with folds, the user draws an approximate medial curve simplifying out the folds, to which an ellipse is automatically fitted (dotted green curve in Figure 2, middle). Based on these annotations, our method generates a two-layered parametric model that separates the coarse shape of the garment from the detailed folds.

To retrieve the local orientation of the different pieces of garment, we also ask the user to outline the bone directions of the limbs locally wrapped by the garment (black segments in Figure 2, middle). These bones are manually matched to their counterparts in 3D rigged character. Finally, the bounding box of the character and the approximate location of the eyes (black circle) are also provided, which will be used as a proxy for the height of the viewpoint from which the sketch has been drawn.

As a precomputation step, we decompose the set of garment contours as a list of minimal cycles bounded by curves alternating between *silhouette-type* and *border-type* (see Figure 2-left). Each cycle defines a 2D patch depicting a piece of garment to be transferred. A patch is automatically identified as *loose* when at least one of its border is loose, or *tight* otherwise.

Note that several steps of this manual annotation pipeline could be automatized using standard vision-based tools, towards an automatic computation of the skeleton, contours, bounding box, and

eyes-position. Still, we aim at being able to handle sketches with high stylization levels, which would hardly fit to a single, general automatic approach, but instead can fully benefit from subjective interpretation. As such, our user-based annotations allow the user to freely explore different possibilities (eg. exploring loose versus tight annotations of a given border, leading to different shapes) while requiring reasonably fast and simple interaction.

4. Extracting style features from fashion sketches

The first step of the method is to extract style features from the annotated sketch. As we aim at transferring the garment to 3D characters of diverse morphologies, the extracted features should be independent from the input character's size and shape.

Our definition of *garment style* inspires from Brouet et al. [BSBC12]. They define the three following criteria to be preserved between two garments, for them to have the same style:

(I) **Scale** (or proportionality), expressed through the *relative location* of the garment w.r.t. the character's body and limbs.
(II) **Fit**, expressed through the preservation of the *tight regions* where the garment should fit the body.
(III) **Shape**, expressed using normals in loose areas.

In our case, to be body-shape and pose independent, we express the normals of feature (III) in the skeleton's frame of the associated body or limb. Moreover, since we want to allow style transfer between radically different characters, we choose to consider the shape criteria at the scale of the *overall shape* described by the simplified hemline, and add the following fourth's similarity criterion to capture details:

(IV) **Folds**, expressed as the shape and frequency of wrinkles along the hemline.

This choice allows characters of different sizes not to have the same number of folds around their dresses, when the same sketch is transferred, enabling to give the impression that the same fabric has been used. The remainder of this section details the extraction of these four features from a 2D sketch.

4.1. Scale, fit and shape of garment patches

Let us consider each of the garment patches extracted from the sketch as described in Section 3. Scale (criteria (I)), described as the relative location of the garment patch onto the body and limbs, is determined using the drawn 2D skeleton. For each border of the patch, we compute and store all intersections between the 2D skeleton and the line linking the border's extremities. This method allows us to process all bones crossed by the border while not suffering from the bias in the shape of the hemline curve due to perspective (see for instance Figure 2-right), where the front of the skirts looks longer than the silhouettes due to a camera position located higher than the hemline). Each intersection j is associated with a triplet:

$$I_j = (b_j, t_j, \alpha_j), \tag{1}$$

where b_j is the bone's identifier, $t_j \in [0,1]$ is the linear coordinate of the intersection point along the bone's segment, and the angle α_j expresses the orientation of the border in the patch. If the border is tight, then α_j is the angle between the normal vector of the border line and the bone. If the border is loose, then it is the angle between the normal vector of the border line and the vector linking the intersection point and the middle point of the opposite tight border of the patch, as illustrated with the blue patch in Figure 2-right.

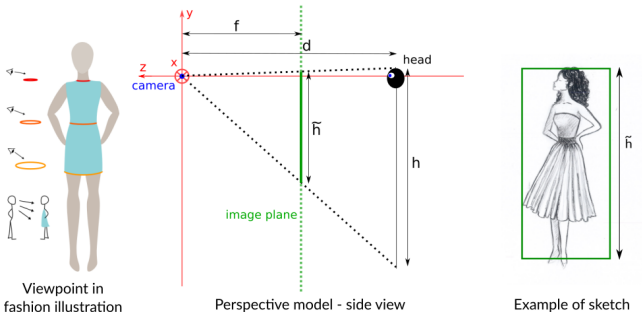


Figure 3: *According to fashion design literature, most fashion designers draw with a perspective effect as if they were looking at the model from the height of their eye (left). We use this assumption to design a virtual camera model (middle) that interprets the folds drawn in the sketch (right).*

Criteria (II) can be fully satisfied with the sole knowledge of parameters I_j and the annotation of tight borders on the input sketch. The *coarse shape* criteria (III), ie. normal orientations, is extracted from silhouette curves. Since the later depict the set of points where the 3D normal vectors are orthogonal to the view direction, the 2D normal vectors along the silhouette curves actually correspond to the projected 3D counterpart in the image plane orthogonal to the view direction. For each loose patch, we compute a local 2D axis by linking the middle point of each border in the patch. We use this axis together with the borderline as a local frame to compute a generic representation for the orientation of silhouette curves, independent from the character pose. We neglect perspective on the silhouette curves as we assume that they are depicted in a plane facing the camera.

Note that the ellipse extracted from the sketch for each garment patch is not used at this stage, but considered while extracting folds, as described next.

4.2. Extracting fold patterns from a sketch

The *folds* criteria (IV) is finally extracted from the detailed shape of the hemlines. In this work, we focus on folds aligned along a given axis typically corresponding to the one produced by garment falling under gravity along limbs, and call them *tubular folds*. Note that if arms or legs are horizontal the limb axis may not be the vertical axis. We detail in the following how we extract fold features that we apply later onto any synthesized 3D garment.

Perspective model

According to fashion literature, designers commonly draw their models as if they were looking at them at their eyes height, at a distance of two to three meters [Wat09], as shown in Figure 3-left.

Given this viewpoint, hemlines at the bottom of long garments are seen from above under perspective, revealing their folds. As a result perspective should be taken into account for analysing folds to accurately retrieve their depth.

Our algorithm assumes a prescribed camera-to-eye distance d, and a character height h. As depicted in Figure 3, the character is seen by a *virtual* camera and displayed in the image plane with a height $\tilde{h}$ that can be measured on the 2D sketch. The focal distance f of the *virtual* camera can therefore be expressed as $f = \frac{d\tilde{h}}{h}$.

Following this perspective model, any 3D position $p = (x, y, z)$ has an image $\tilde{p}$ in the sketch at the 2D coordinates

$$\tilde{p} = \left(\frac{fx}{z}, \frac{fy}{z} \right) . \tag{2}$$

As in Robson et al. [RMSC11], we make the assumption that the hemline is planar and lies within a plane $\mathcal{P}$ orthogonal to the image plane. Let us now consider the ellipse approximating the coarse, fold-free hemline (dotted green curve in Figure 2). This 2D ellipse corresponds to the projection of a 3D ellipse lying in $\mathcal{P}$, and centered at depth $z = -d$. We call $\Omega = (\Omega_x, \Omega_y, -d)$ and $n = (n_x, n_y, 0)$ its respective center point, and unit normal. We show in appendix A that these 3D parameters can be explicitly computed from its 2D projection.

Finally the 3D correspondence of the detailed folded hemline can be retrieved using this perspective model. Let us consider the 2D annotated hemline point $\tilde{p}_h = (\tilde{x}_h, \tilde{y}_h)$ and its 3D correspondence $p_h = (x_h, y_h, z_h)$ on the folded hemline. As p_h is assumed to lie within the plane $\mathcal{P}$, we have

$$\begin{aligned} & (p_h - \Omega) \cdot n = 0 \\ \Leftrightarrow \quad & (\tilde{x}_h \, z_h / f \, n_x + \tilde{y}_h \, z_h / f \, n_y + z_h \, n_z) - \Omega \cdot n = 0 \\ \Leftrightarrow \quad & z_h = f \, \frac{\Omega \cdot n}{\tilde{x}_h \, n_x + \tilde{y}_h \, n_y + f \, n_z} . \end{aligned} \tag{3}$$

Finally, z_h can be used in Eq. (2) to retrieve x_h and y_h coordinates. We therefore end up with a 3D curve for the folded hemline.

Character-independent parameterization

The depth extraction process we just described outputs a 3D planar folded curve wrapped around a specific ellipse. As we aim at a generic representation of folds, independent from the shape of the garment's hemline, we normalize the curve by first applying an affine transform mapping the ellipse onto a unit circle and then choosing an angular parameterization, Fig. 4a-c. The curve is then converted to polar coordinates which allows to represent it as a parametric curve of a generic angular parameter (Fig. 4d).

As shown in Figure 4d, this curve called $\mathcal{D}$ is only a partial representation of the complete border. This comes for two reasons: (a) occlusions caused by deep folds may occur under the depicted viewpoint, (b) only the front part of the curve was visible on the sketch. Under the assumption that we have a continuous fold curve, we propose to solve (a) by introducing a similarity-based completion algorithm to infer the missing occluded parts of this curve as detailed in the next section, and (b) by decomposing the curve into a serie of single folds at the inflexion points, and duplicating them in the same order until the entire circular domain is filled. To ensure

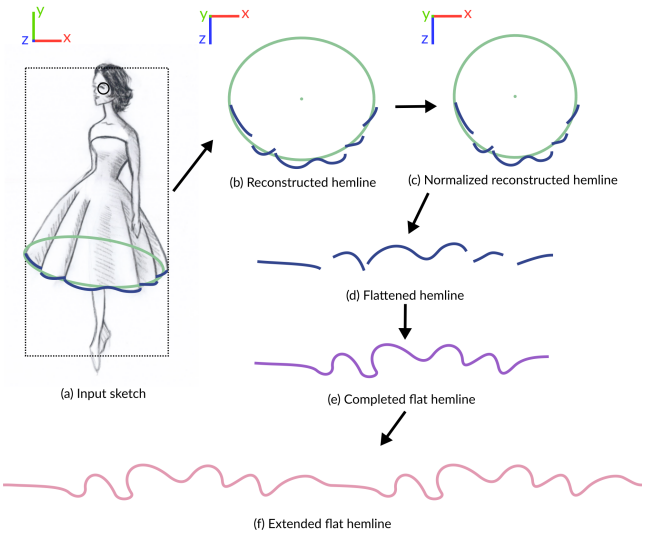


Figure 4: *Example of folded hemline reconstruction. Our algorithm takes as input the 2D image of a folded hemline and an ellipse approximating the image of the hemline without folds (a). First it computes a reconstruction of the hemline, using plane estimation and reverse perspective (b). Then, this curve is normalized by applying the affine transformation for which the image of the ellipse is a unit circle (c). Third, our algorithm computes a polar representation (d) of the normalized curve. If the hemline is discontinuous, we apply our completion algorithm (e). This partial representation for folds is then extended to a complete curve by duplicating single folds (f) to fill the entire angular domain* $[0, 2\pi]$.

a smooth periodicity, the entire curve is scaled to exactly span 2π, and a G^1 continuity constraint is applied at the end-points.

Completion of occluded parts

Our goal is to complete the normalized parametric curve $\mathcal{D}$ toward a continuous and smooth curve S, by inferring the missing parts. We model S as a spline composed of N cubic Bézier curves S_i, with

$$S_i(t) = \sum_{j=0}^{3} b_j^i \, B_j(t) \, , \tag{4}$$

where B_j are the cubic Bernstein polynomials basis and b_j^i are the control points of S_i. The general idea is to consider that non-visible curve portions should be similar to visible ones. We thus force S to approximate $\mathcal{D}$ in the visible region, while inferring non-visible part using similarity as well as continuity criteria. Note that S only approximates $\mathcal{D}$ in the visible part in order to remain robust to possibly imprecisely sketched folds depicted by the user, especially at grazing angle. Our method works as follows.

We first generate an initial guess Bézier spline $\hat{S}$ interpolating $\mathcal{D}$ in the visible region, and completing the missing part using C^1 continuity (see Fig. 5a). To ease further analysis, $\hat{S}$ is resampled such that individual junctions between *polynomial segments* occur at inflexion points of the spline (an example of such decomposi-

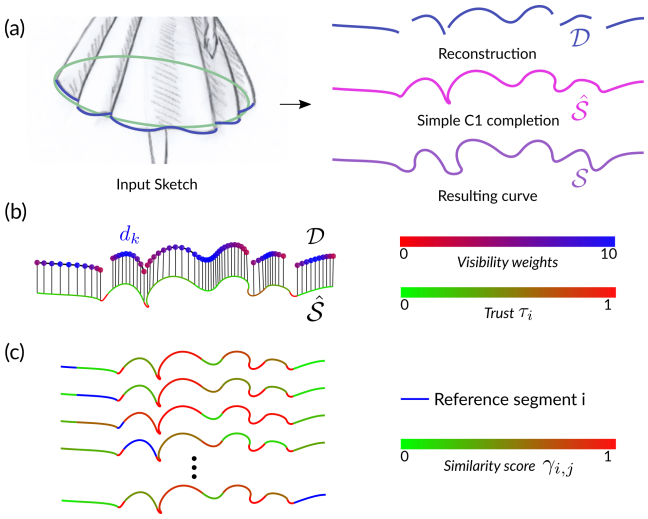


Figure 5: *Completing occluded portions of the folded hemline. a) The input curve $\mathcal{D}$ is incomplete. A first initial continuous guess curve is generated $\hat{S}$ to fill the missing parts, and is further deformed into the final approximating folded curve S. b) The projection of samples of $\mathcal{D}$ onto $\hat{S}$ are used to measure a trust score τ_i and a per-sample visibility weight. c) Bézier segments are compared together using a geometrical similarity score $\gamma_{i,j}$.*

tion is illustrated in Fig. 5c) We also consider a set of uniformly distributed samples $\{d_k\}$, $k = 1,\dots,K$, of $\mathcal{D}$ and associate to each sample its orthogonal projection $\hat{S}_i(t_k)$ belonging to the polynomial segment $\hat{S}_i$ at some parameter value t_k.

Secondly, we associate a *trust score* $\tau_i \in [0,1] = t_k^{\max} - t_k^{\min}$ to each segment of S given by the size of the range of parameters $[t_k^{\min}, t_k^{\max}]$ on which samples $\{d_k\}$ are projected to $\hat{S}_i$. A fully visible segment will therefore be associated to the trust score 1, while a fully occluded one will be associated to 0 (see Fig. 5b).

Third, we define a *shape similarity score* $\gamma_{i,j}$ between two polynomial segments (i,j), as illustrated in Fig. 5c, in computing the maximal angle between corresponding tangent vectors over a discrete uniform sampling of the curve $\hat{S}$ at parameters $t_r = r/(R-1)$, with $R = 10$.

$$\gamma_{i,j} := \max_{t_r} \operatorname{acos}\left(\frac{\hat{S}_i'(t_r)}{\|\hat{S}_i'(t_r)\|} \cdot \frac{\hat{S}_j'(t_r)}{\|\hat{S}_j'(t_r)\|} \right) . \tag{5}$$

The map $Sim : i \to j$ associating a segment i to its most similar segment j is precomputed by taking into account the *shape similarity score* weighted with respect to the *trust score* to favor the most visible segments

$$Sim(i) = \underset{\substack{j\in\{0,\dots,N-1\}\\ j\neq i}}{\arg\min}\ 1 - e^{-\gamma(i,j)^2/\tau_j^2}. \tag{6}$$

Note that we only consider a segment to be similar to another one if its associated value $1 - e^{-\gamma(i,j)^2/\tau_j^2} > 0.5$, otherwise we set $Sim(i) = -1$ allowing to prefer completion based on other criteria such as smoothness in the next optimization.

The final curve S is globally computed by minimizing the following quadratic energy E_S, expressed with respect to the control points of the spline:

$$E_S = E_{visibility} + E_{similarity} + E_{distance} + E_{G^1} . \tag{7}$$

- $E_{visibility}$ enforces the curve to match the discontinuous projection

$$E_{visibility} = \sum_{k=0}^{K} \omega_{visibility}^k \|S_{i_k}(t_k) - d_k\|^2,$$

where the visibility weights $\omega_{visibility}^k$ are defined using a Gaussian function centered on the middle index of each continuous part of $\mathcal{D}$, such that $\omega_{visibility}^k = 1$ at discontinuities and $\omega_{visibility}^k = 10$ at the center of continuous parts, see an example in Figure 5b. This energy component allows to locally limits clearly visible part of the curve to be deformed, while discontinuous part, often seen at grazing angles, may be more easily modified.
- $E_{similarity}$ encourages segments that are not clearly visible to locally reproduce the shape of their associated similar segment,

$$E_{similarity} = \sum_{i=0}^{N-1} \omega_{similarity}^i \sum_{r=0}^{R-1} \|S_i'(t_r) - S_{Sim(i)}'(t_r)\|^2,$$

where $\omega_{similarity}^i = (1-\tau_i)$ if $\tau_j < \frac{1}{2}$ and $Sim(i) \geq 0$, and 0 otherwise.
- E_{G^1} enforces the tangent-continuity at segment junctions

$$E_{G^1} = \omega_{G^1} \sum_{i=0}^{N-2} \|(b_2^i - b_3^i) - \alpha_i(b_0^{i+1} - b_1^{i+1})\|^2,$$

where $\omega_{G^1} = 1$, $\alpha_i = \|\hat{b}_2^i - \hat{b}_3^i\| / \|\hat{b}_0^{i+1} - \hat{b}_1^{i+1}\|$ with $\hat{b}_i$ being the control points of the initial guess curve $\hat{S}$.
- $E_{distance} = \omega_{dist} \sum_{i=0}^{N-1} \|b_0^i - \hat{b}_0^i\|^2$ prevents the points from varying too much from their initial position. This criteria is only useful to initialize the general placement of the curve and we set a low weight $\omega_{dist} = 0.01$.

We end up with a continuous, closed planar curve, expressed in polar coordinates.

5. Synthesizing 3D folded garments over target characters

Once style and folds features have been extracted from a 2D sketch, we are ready to synthesize a 3D garment satisfying these features over any target, rigged characters, and while matching its individual morphology and pose.

Garments are processed by creating a surface patch for each cycle extracted from the sketch, and processing them in the decreasing vertical order, while enforcing continuity between neighboring patches along shared borders.

In the following, we explain how we model a 3D surface corresponding to each surface patch of the garment. We proceed in four steps (see Figure 6). (1) garment pieces matching the relative location of the borders along the limbs are tightly synthesized on top of the mannequin. If the patch shares a boundary with another one, we set the shared border to match its twin in the previous patch. (2) loose parts are deformed such that their normals match those from

the sketched silhouettes. (3) collisions between garment and body surface are handled. (4) geometrical folds are finally added along the loose pieces of garment.

5.1. Positioning

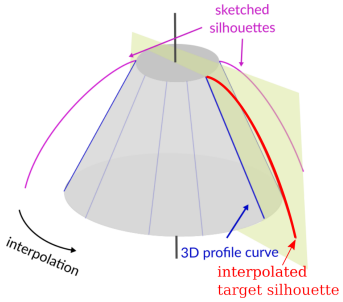
The goal here is to compute a canonical surface as a generalized cylinder (2 boundary curves connected by straight lines) and to position it onto the 3D characters by wrapping it around the relevant body and limbs and by preserving the proportions of the 2D sketch. In practice, we make the assumption that 3D boundary curves of the surfaces are all planar. We therefore first need to compute the planes in the target character's space, and second to compute the boundary curves within these planes.
The intersection parameters (Eq. (1), Fig. 2) of the 2D straight border with the 2D bone $I_j = (b_j, t_j, \alpha_j)$ are used to define a point in the plane and a normal vector. The point belonging to the 3D bone, where the plane intersects, is obtained by affine interpolation of the bone's endpoints at parameter t_j. The plane's normal vector is computed by rotating the 3D bone direction vector by α_j around the z-axis.
Note that borders may contain several intersections when surrounding multiple limbs. In this case we compute the border's plane using the mean rotated 3D normals and 3D bone intersections points.

The 3D boundary curves of the generalized cylinder, acting as garment boundary curves, should best fit the character's body. They are computed inside each plane as the convex hull of the intersection points between the 3D plane and the mannequin's surface mesh. To enforce a 3D smooth parametric model, we convert this curve into a cubic Bézier spline by first, subsampling the convex-hull curve using Ramer-Douglas-Peucker algorithm [DP73], and approximating the coarse samples in a least-squares sense as illustrated in Figure 7. The generalized cylinder is then generated by adding the rulings joining the two boundary curves at the spline junctions, see Fig. 6 for a result. Note that both boundary curves are reparameterized in order to redistribute initial junctions, while correspondence between samples is computed in polar coordinates.

5.2. Basis Surface

From the generalized cylinder a smooth 3D *basis surface* (without folds) satisfying the target proportions (I), style criteria fit (II) in tight regions, and shape (III) in loose ones is computed.

Let us first focus on loose garments, where the normal orientations are a key shape feature, since they express the global shape of the garment, e.g. straight, round, bell or tulip shaped. The generalized cylinder serves as scaffold. Its rulings, called *initial profile curves*, will be deformed into *final profile curves*, see blue and red curves in Figure 8 (middle,right). The idea is to rotate the sketched silhouette curves around the cylinder axis by keeping its anchor point (one of its endpoints) on the cylinder boundary curve, and by deforming it slightly in order to interpolate the left and right sketched silhouette curves, which are not identical.



The rotating silhouettes will therefore describe the final shape as a generalized surface of revolution, modeled using a Bézier surface in our implementation. To achieve this, we transfer the normals and thus the coarse shape from the sketched curves onto the final profile curves, as follows. Given the set of M initial profile curves (rulings of the cylinder), the corresponding final profile curves are obtained by interpolating the sketched 2D silhouette curves (after scale and rotation according to the cylinder axis length and orientation). They are thus planar and placed in the 2D section plane spanned by the cylinder axis and the corresponding straight profile curve (see the inset), so that their endpoint is anchored at the tight cylinder boundary curve.

In the case of tight garments, the shape is fully guided by the body on which it is transferred. In this case the direction of the tangent vector at the extreme position of each profile curve is defined by the intersection of the plane containing the profile curve, and the tangent plane of the body mesh at the corresponding closest point. A cubic tensor product Bézier surface interpolating the final profile curves is then computed, so that the surface is tangent-plane continuous across these curves.

5.3. Collisions avoidance

At this step, the synthesized garment surface modeled as a Bézier surface matches the proportion, fit and shape criteria with respect to the sketch, but it may be in collision with the body surface (as in Fig. 9.middle), stored as a triangular mesh. Collision handling between a meshed surface and a body is a well studied problem for garment modeling [BWK03, JYSL19]. Interpenetration is typically solved in pushing penetrating vertices outside of the body volume along a vector normal to the closest body position [RMSC11, GRH*12]. Even though inspired by these approaches, our problem states differently, since we have to solve penetrations between a parametric surface and a mesh. We propose to interactively subdivide the Bézier patches where maximal penetrations occur, and to deform the newly inserted control points in order to push the garment surface outside the body as illustrated in Figure 9.

Collisions are detected by marching along the generalized cylinder's axis while sampling both surfaces radially within a plane orthogonal the bone segment. When a collision occurs, the penetration depth within the body can be computed in this plane. For each plane, we evaluate a collision distance by computing the cubic spline curve Q of intersection with the garment surface, and the polyline P of intersection with the character's body. Q is composed of C Bézier curves Q_i of control points denoted $q^i_j, j = 0,..,3$.

$$\forall t \in [0,1], Q_i(t) = \sum_{j=0}^{3} q^i_j B^3_j(t)$$

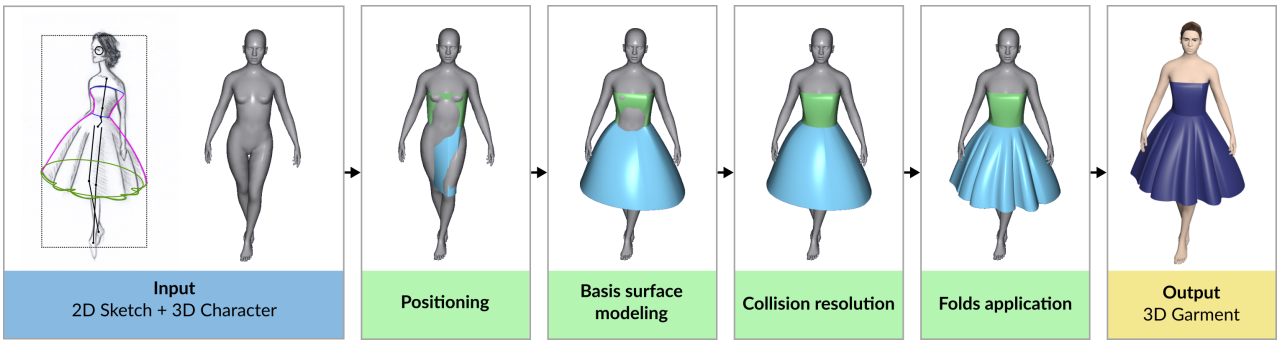


Figure 6: *The four steps of our garment synthesis pipeline satisfying the style-transfert criteria defined in Sec. 4.*

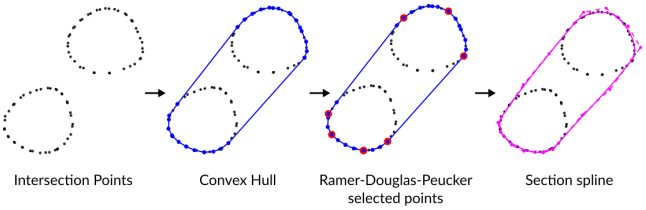


Figure 7: *Construction of a Bézier spline fitting the convex hull of a set of points representing the cut of the body mesh along a plane.*

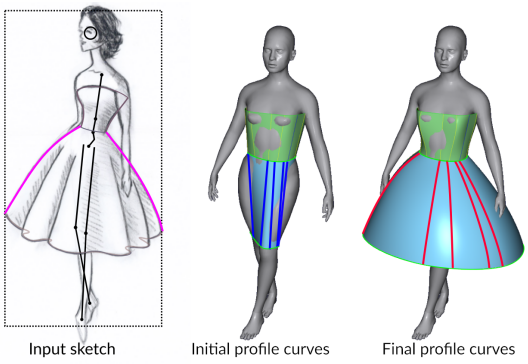


Figure 8: *Transferring normals of an input sketch (left) to the basis garment (right). Initial profile curves are displayed in blue. Silhouettes drawn in the input sketch (pink curves) are scaled and adaptively rotated around a vertical axis in order to serve as target curves for the final shape.*

We compute the collision distance as

$$d_{col} = \max_{p,i,t} \left\{ \|\overrightarrow{pQ_i(t)}\| \mid \overrightarrow{pQ_i(t)} \cdot Q_i'(t) = 0 \text{ and } Q_i(t) \text{ is interior to } P \right\}$$

We denote $(i_{col}, t_{col}, p_{col})$ the parameters of the deepest collision of Q in P : $d_{col} = \|\overrightarrow{p_{col}Q_{i_{col}}(t_{col})}\|$ (see an example in Figure 9).

In the case of loose garments, the action of gravity makes the fabric fall on the highest part of limbs first. We model this behavior by marching along bones from top to bottom, while stopping the sweeping process at the first locally maximal penetration encountered. The parametric surface is then subdivided by inserting Q as an extra section curve in the surface, in order to gain sufficient degrees of freedom to allow its deformation. The actual deformation of the Bézier curve is computed as a quadratic energy minimization problem, weighting between repulsion of the deepest colliding position and preservation of the current shape as measured by the preservation of curve derivatives.

Using the previous notations, we set $\tilde{p}_{col} = p_{col} + \varepsilon \frac{\mathbf{d}_{col}}{||\mathbf{d}_{col}||}$, where $\varepsilon = 10^{-1}$, as the target point the curve needs to reach, while keeping at best its initial shape. The spline is closed and C^0 continuous, so the control points of Q_i verify: $q_3^{i-1} = q_0^i$, and $q_0^i = q_3^{C-1}$. We also denote by $\hat{q}_j^i$ the initial coordinates of the control points. The collision is then solved by minimizing the following quadratic energy on the control points of all Bézier segments Q_i of the spline Q in the plane.

$$E_{collision} = E_{col} + E_{stretch} + E_{G^1} + E_{data}\ , \tag{8}$$

where

- $E_{col} = ||\tilde{p}_{col} - Q_{i_{col}}(t_{col})||^2$
  aims at pushing the spline outside of the body at the location computed as the point of deepest penetration
- $E_{stretch} = \sum_{i=0}^{C-1} \sum_{j=0}^{K-1} ||\hat{Q}_i'(\frac{j}{K}) - Q_i'(\frac{j}{K})||^2$
  to minimize variations in the derivative. We fix the number of tangents sampled in each Bézier curve $K = 4$, which is enough to depict the shape of the degree 3 curve.
- $E_{G^1} = \sum_{i=0}^{C-1} ||(q_2^i - q_3^i) - \alpha_i(q_0^{i+1} - q_1^{i+1})||^2$,
  with $\alpha_i = \frac{||\hat{q}_2^i - \hat{q}_3^i||}{||\hat{q}_0^{i+1} - \hat{q}_1^{i+1}||}$ to enforce tangent continuity at junctions. Here, we extend the notations to account for the fact that the section curve is closed : $q_0^N := q_0^0$ and $q_1^N := q_1^0$.
- $E_{data} = \sum_{i=0}^{C-1} \omega_i ||q_0^i - \hat{q}_0^i||^2$ with $\omega_i = e^{-\frac{d_i^2}{2\sigma^2}}$
  where d_i is the projected distance of q_0^i to P if it lies inside P, and 0 otherwise, and $\sigma = 0.2 \max_{i \in \{0..N-1\}}\{d_i\}$, keeps the initial position of the control points that are far from p_{col}.

. Once the collision is corrected, the marching algorithm is pursued until reaching the lowest extremity of the bone.

Contrary to loose garments, the geometry of tight ones can be considered as independent from the action of gravity. Therefore, we

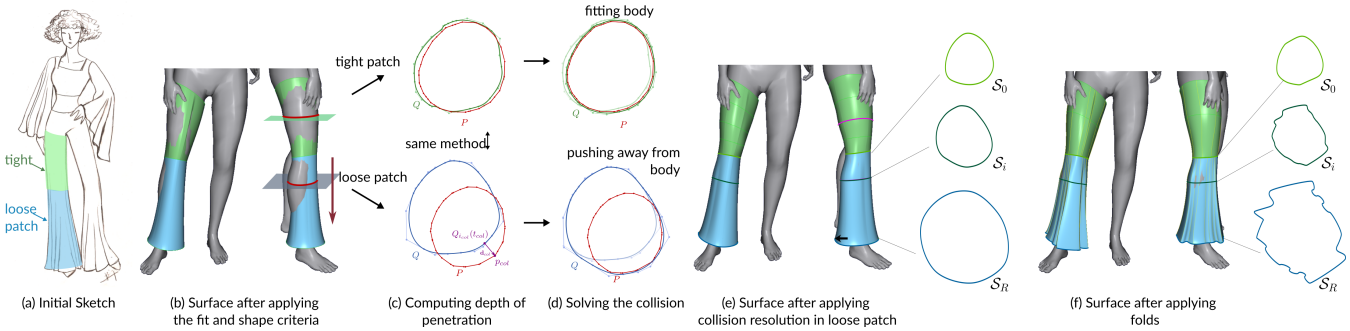


Figure 9: *Illustration of our algorithm for solving collisions with body, and applying folds in the garment surface. Collisions are detected by marching along the generalized cylinder's axis (b), and are solved by inserting new sections within the surface patch. These sections are then deformed to fit the body at best (c,d,e). Folds are then applied to each loose section curve of the surface, and interior section curves are deformed to create a smooth transition in loose patches (f).*

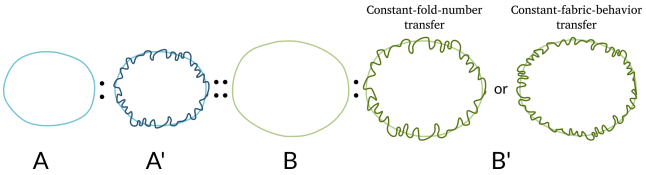


Figure 10: *Example of transferring folds from a border A to a border B. We propose two different strategies to produce B', one preserving the number of folds in A', and another preserving their shape.*

iteratively process the deepest detected intersection along the entire bone. The Bézier surface is also subdivided and set to match the convex hull of the body section, following the algorithm in Fig. 7.

5.4. Applying folds to the surface

To achieve criterion (IV) for style similarity, we finally deform the loose parts of the garments by applying folds that match the detailed fold features extracted from the sketch in Section 4.2 (namely, a 2D curve parameterized with a generic angular parameter). We detailed below our algorithm to transfer this fold distribution to a different curve used as border of the target surface.

As depicted in Figure 10, we propose two different transfer strategies, depending whether we want to maintain the same number of folds as the input, or to keep the number of folds per arc length constant in order to mimic a similar fabric. Comparative results of these two strategies can be found in Fig. 13.

Constant-fold-number strategy. The first strategy maps the fold curves along the angular parameter of the target section curve, independently of its length. As a result, the number of folds remains constant, while their sizes scale with the curve length. This approach allows to visually give the appearance of a similar garment, as was done in [BSBC12].

Constant-fabric-behavior strategy. The second strategy seeks to adapt the number of folds with respect to the arc-length of the target section curve, while maintaining a constant size of folds. This is done by scaling first the distribution by the arc-length of the ellipse reconstructed using the perspective model. Then, the fold distribution can be extended or cut out in order to be sized with the length of the target curve with a method similar to the completion of the hemline in Sec. 4.2. In this model, the visual style of the entire garment may not be preserved when mapped onto a very different character but the local arc-length preservation of folds enables to mimic the transfer of a garment made of the same fabric as the initial one, to the target character.

In the case of loose patches, we can use one of these methods to transfer the folds to the section curve $\mathcal{S}_R$ corresponding to the loose boundary of the patch. If the loose patch contains a tight boundary, we want the garment to smoothly fold along the patch as it becomes loose to the body (e.g. loose patch in the pants of Fig. 9). Therefore, we resample the tight boundary of the patch $\mathcal{S}_0$, so that extremities of the Bézier path in the spline have same polar coordinates in the cylinder than the ones in the loose boundary curve. For each interior section curve $\mathcal{S}_i, i \neq 0, i \neq R$ in the patch, we compute a linear interpolation between a resampled version of the curve $\mathcal{S}_i^U$ (obtained like the tight boundary curve), and a folded version of the curve $\mathcal{S}_i^F$ (obtained like the loose boundary curve), with an interpolation factor γ_i depending on the position of the section curve along the cylinder's axis (0 near the tight border, 1 near the loose border)

$$\mathcal{S}_i = (1-\gamma_i)\mathcal{S}_i^U + \gamma_i \mathcal{S}_i^F, \text{ with } \gamma_i = \frac{\sum_{j=0}^{i-1} \|\overrightarrow{o_j o_{j+1}}\|}{\sum_{j=0}^{R-1} \|\overrightarrow{o_j o_{j+1}}\|},$$

where the o_j are the barycenters of each section curve $\mathcal{S}_j$.
As illustrated in Fig. 9, this interpolation ensures a smooth transition between the loose boundary curve which is completely folded, an the tight boundary curve which is not folded.

6. Results

We firstly tested the ability of our algorithm to qualitatively reconstruct the 3D coordinates of a folded hemline curve from a single view sketch when the ground truth model is known. We modeled a synthetic idealized conical dress-like shape falling under gravity while fixing the top circular border of radius 1 using a physically-based cloth simulation. Using four different materials pre-set from

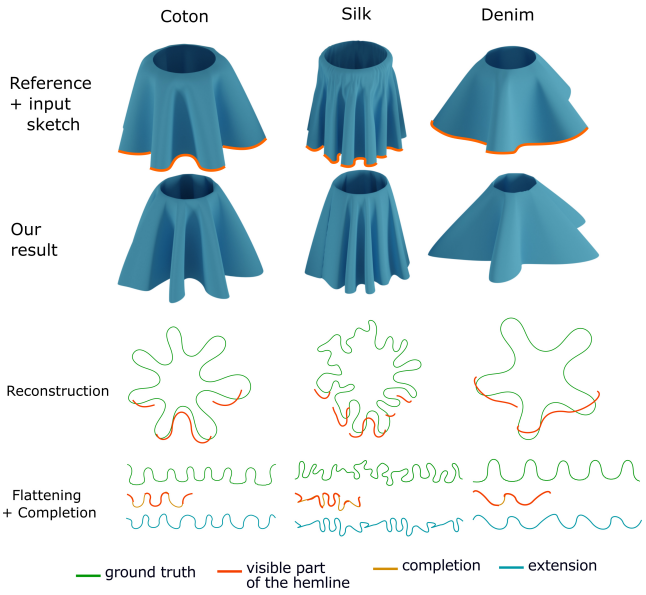


Figure 11: *First row: Reconstruction of fold curves generated from physics-based simulation with three different parameters (cotton, silk and denim) around a cylinder. The result of the simulation is rendered on an image, on top of which we annotated the visible part of the folded hemline. 2nd row: reconstructed garment. 3rd row: reconstructed visible hemline coordinates curve. 4th row: automatic completion of the missing occluded parts of the hemline in normalized angular coordinates.*

the Blender software †, we computed different variations of folded-shapes. Each simulated mesh is rendered as an image, and the user draws on top of the resulting image the visible parts of the folded border, and the approximated ellipse. Our algorithm generates from these annotations a 2D curve in the normalized space that we compare to the true trace of the 3D border curve projected orthogonally onto the horizontal plane. This 3D planar curve is then flattened, completed and extended using the method described in Sec. 4.2. The results displayed in Fig. 11 show that the reconstructed hemlines are in general visually close to ground truth, even with irregularly shaped folds, such as the silk model. One can note that results that look the least accurate correspond to garments with only a few, large folds. Indeed, the true 3D folded border curve deviates more, in this case, from the planar hypothesis assumed during the reconstruction process.

Let us now compare several garments synthesized from a single, annotated sketch. To generate all such results, we made the assumption that the sketched character, of height $h = 170$ cm, was seen by an observer at a distance $d = 250$ cm [Wat09]. This enabled us to set the camera position used to interpret the sketches. When not indicated otherwise, the constant-fold-number strategy was used to apply folds to the synthesized garments.

The first four examples of Figure 1 show transfer to a 3D mannequin under the same pose and view angle as in the sketch, thus illustrating the ability of our method to preserve the visual appearance of the different types of cloth and folds. In contrast, the models at the right side of Figure 1 illustrate our garment synthesis method on characters with various, possibly extreme, poses and morphologies, such as the dancing girl and the cartoon characters. More results are shown in Figure 12, where a single sketch is used to dress two different 3D characters. While their morphologies are drastically different, from human to cartoon style, our method is able to efficiently synthesize adapted 3D dresses of similar style. Note that the tight top and top tight part of the pants and the on the second human model was manually added, but could be generated using former methods offsetting body parts for instance [TWB*07].

As our approach handles the overall silhouette of the garment separately from the detailed folds, synthesized models can also be set to mix the styles from multiple sketches. For example, Figure 13 illustrates the option to synthesize the general silhouette appearance from a given sketch –here a dress (a)–, while applying folds extracted from another sketch –here some pants (b)–.

We also illustrate in Figure 13 the twofold transfer strategies described in Section 5.4, namely the transfer with *constant-fold-number*, which re-scales the folds to fit to the border length of the new garment (c), versus the transfer with *constant-fabric-behavior*, which preserves the frequency and magnitude of folds by duplicating them such that they adapt to a longer border (d).

Lastly, our algorithm also allows the input sketch to be more expressive than realistic. For example, the second sketch of Figure 14 (as well as the manga-style girl in Figure 1) show garments drawn on unrealistic sketched morphologies, but that are still plausibly transferred to human bodies. In Figure 15 we also show how the surface varies whether it is treated as tight or loose. We can note that garment on the shoulder has a local orientation following the silhouette of the sketch when the corresponding patch is annotated as loose, while the same garment follows the surface body only if the patch is annotated as tight.

This approach was implemented on non-optimized C++ code and runs from a standard laptop model. All shown examples were annotated by the authors in 5-10 minutes and generated in less than 15s. Roughly half of the time was spent on the collision correction step.

Comparison with state-of-the-art methods

Our work adapts and extends the notion of "style preservation" initially proposed by Brouet et al. [BSBC12]. Note, that our input garment is provided only as a 2D drawing instead of a complete 3D surface. We show in the following that our approach is, first, able to mimic the type of clothing and folds shape achieved by the approach from Brouet et al.from only 2D input, and second, able to achieve a similar "style transfer" between characters.

To this end, we annotated an image of the article exhibiting the transfer of a dress from an adult toward a child. Figure 16 illustrates our results where the adult dress is annotated, and used to generate the 3D meshes on both the adult and child mannequin. Figure 16-middle shows the use of our *constant-fold-number strategy*, similar to the notion of "style transfer" defined in Brouet et

† www.blender.org

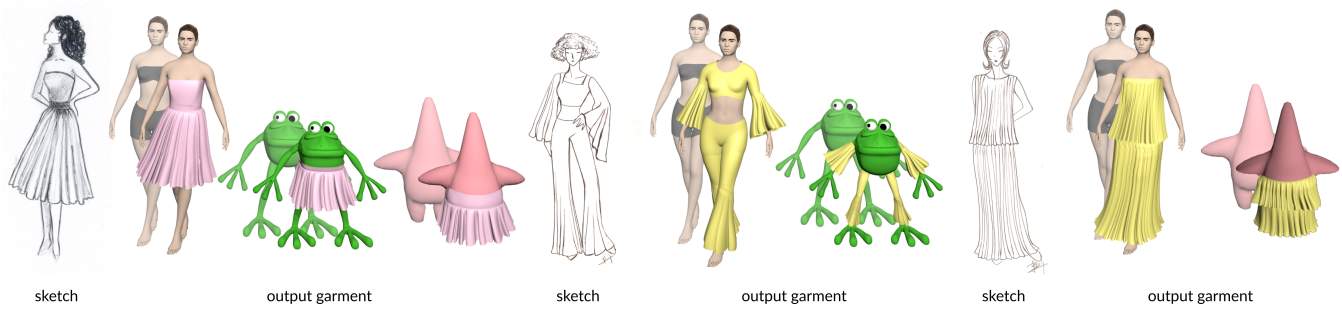


Figure 12: *Example of garments generated from a unique sketch to different characters.*

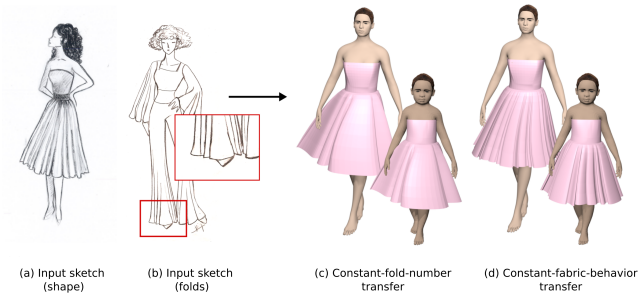


Figure 13: *Example of fold transfer from a garment to another. Our approach allows the computation of a garment using the shape of one sketch (a) and the folds drawn in another sketch (b). Different strategies may be applied, whether we want to preserve the number of folds in the sketch (c), or their size and frequency modeling the fabric behavior (d).*

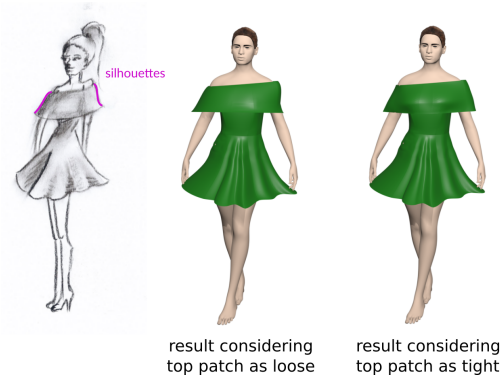


Figure 15: *Example of garments generated from different interpretation of a sketch. In the first version, the top garment patch is considered as loose, whereas in the second version it is considered as tight.*

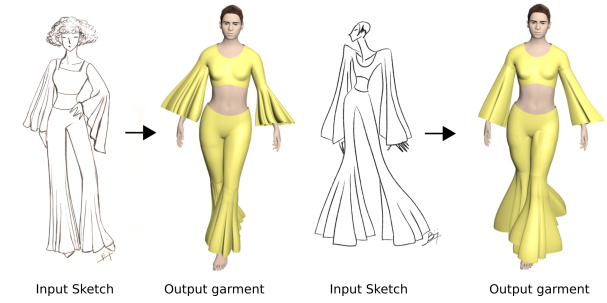


Figure 14: *Example of garments generated from different sketches. Even if the sketch on the right shows less realistic morphology and more expressiveness, both are correctly positioned and modeled around a realistic character.*

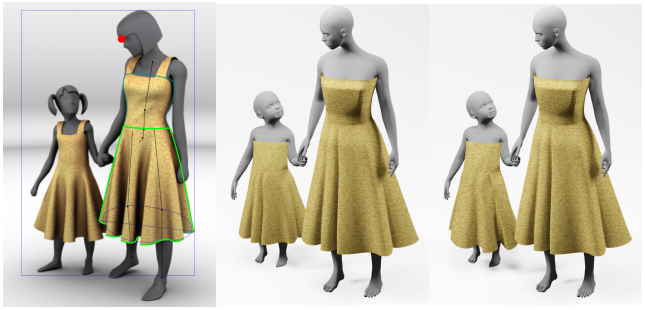


Figure 16: *Garments modeled with our method, using an image of [BSBC12] as 2D input (left). We show the use of our "constant-fold-number" (middle) similar to this reference approach, and our extension to "constant-fabric-behavior" (right) applied from the adult reference toward the child model.*

al. [BSBC12] by reproducing the same number of folds between the adult and the child. Figure 16-right uses the *constant-fabric-behavior* and extends the notion of transfer where fewer folds appear on the child's dress while preserving the same fold width with respect to the adult's dress.

We also compared our results with some examples from Yang et al.'s data driven algorithm to reconstruct a 3D garment from an image [YPA*18]. As illustrated in Fig. 17, we annotated each image of Yang et al.'s result (cf Sec. 3), and generated the corresponding garment on a generic mannequin. Note that the view angle of the camera of the three last images was set to be parallel to the dress'

Figure 17: *Comparison with [YPA*18]. Note that due to the lack of significant hemline in the inputs b,c,d, we applied folds extracted from the examples of Fig. 11.*

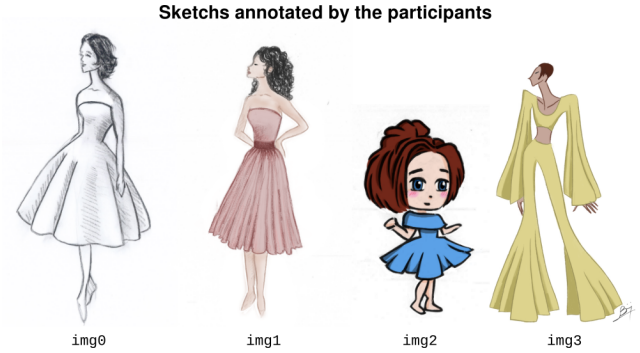


Figure 18: *Drawing used in our user study.*

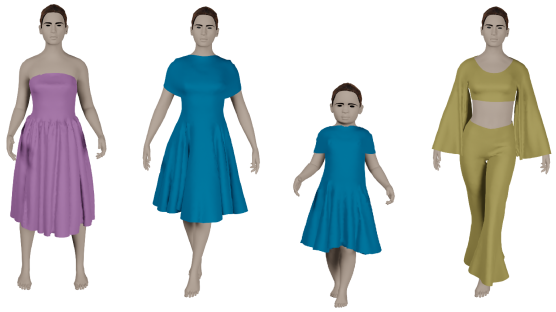


Figure 19: *Garments manually modeled by a 3D artist in 3DS-Max, using sketches (Fig. 18.img1-3) as references. Each model took more than one hour of work from an experienced artist. Note that the style does differ from the one in the sketch, such as a bell shape instead of a tulip shape for the pink dress, and a quite different rest angle for the blue one.*

hemline, which therefore does not fit to our input requirements. As such, the ripples of the hemline were only annotated in the first image, while we used the synthetic examples shown in Fig 11 for the three last images. As depicted in the comparison, our method is able to output garments of comparable fidelity, however our approach is an order of magnitude faster as it requires less than ten minutes per garment (mostly spent in user-interaction), in comparison to an average of 5 hours for the reference method.

Usability

We conducted a user study, in order to evaluate the usability and efficiency of the system using the drawings displayed in Fig. 18.

A first reference experiment was conducted by asking an experienced 3D artist to recreate the outfits from img1, 2 and 3, with a particular emphasis on the folds appearance, using a traditional 3D modeling tool [‡]. The artist choose to model an unfolded mesh model around the mannequin and used the embedded physically-based cloth simulator of the software to simulate the folds falling under gravity and colliding with the body. The results are provided in Fig. 19 and took respectively 3h30, 2h15, 1h30 and 1h30 to model. While the folds have a natural appearance and handle well collision with the body thanks to the use of a physically-based simulation, sharp folds exhibited in img2 could not be be well reproduced, and the silhouettes of the outfit are mostly guided by gravity and not following the reference input sketch. A complete manual modeling could indeed be used to improve the fidelity to the sketches, but would lead to even higher modeling time.

In parallel, we asked 9 participants, that were not expert-artists, to annotate the four drawings each, using the software Inkscape [§], while reporting the time they spent annotating each drawing. A video and text tutorial (see supplementary material) were first presented to the participants. These documents depict a reference annotation shown for the case of img0, which is therefore used to train the participants. No reference annotation was provided for the three

[‡] 3DS Max: www.autodesk.com/products/3ds-max

[§] www.inkscape.org

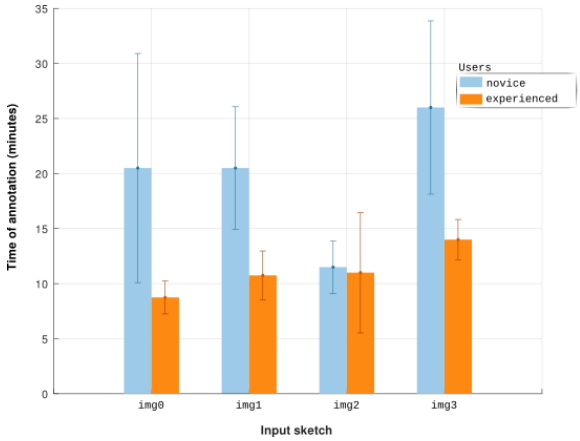


Figure 20: *Sketches to be annotated and annotation time from the user study. Participants are divided in 2 categories : beginners and experienced users for the Inkscape software on which the study was made. For each sketch, and each group of users, we display the mean and standard deviation values of the time taken by the participants to make the annotations.*

other images. As half of the participants were not familiar to the use of Inkscape software, and in particular to the use of its Bézier curves tools, we also asked whether they considered themselves as beginner or experienced user of the software. The corresponding timings are displayed in Fig. 20.

We note a significant difference between timings of the novices versus experienced users of Inkscape. Overall, the experienced users required only half of the time, i.e. 8-12 minutes on average instead of 12-26 minutes for novice users. In all cases, the time spent using our system is an order of magnitude faster than the use of standard 3D modeling software from an experienced graphics artist.

We then used our method to dress virtual characters with the annotated sketches from the user study. In 3 of 36 cases, the annotations were displaying a topology breaking the hypothesis we made for our method, such as silhouette curves covering actual borders of the garment, and crossing the character's limbs in the 2D sketch so that our system could not generate an output garment. Among the 33 generated results we observe different interesting cases highlighted in Fig. 21 :

- Variations in the annotation of the skeleton. For example, different lengths and directions of the bones may lead to garments of different lengths, see `img1`(a) and `img1`(b), or different orientations, see `img0 (a)` and `img0`(b). It is an inherent feature of our method to take the lengths and bone orientations into account and therefore to provide a versatile and direct control enabling for subjective sketch interpretations.
  In other cases, our method manages to recover a coherent set of garments from variations in the orientations of the bones: for example, while the legs were annotated very differently in `img0`(a) and `img0`(b), silhouettes in the resulting garment remain consistent.

Figure 21: *Some highlighted examples of garments generated with our algorithm using sketches of the user study.*

- Variations in the annotation of the folded hemline. Sketches in which some parts of the folded hemline are occluded were interpreted by the annotators in different ways: some drew one continuous curve including both visible parts of the garment's hemline and occluded folds silhouettes (`img3`(b)), and others drew discontinuous curves with only the visible parts of the hemline (`img3`(a)). Our method handles robustly the two types of inputs and generates a continuous hemline in all cases. The main difference between the two versions can be noted by the sharper folds in the first case as the completion algorithm interprets the fold silhouettes as the hemline.
- Variations in the interpretation of loose and tight areas. For some garments parts, such as the legs in `img3` or the dress in `img2`, some participants considered the cloth as fully loose (`img3`(b), `img2`(b)), while others distinguished a tight area and a loose one

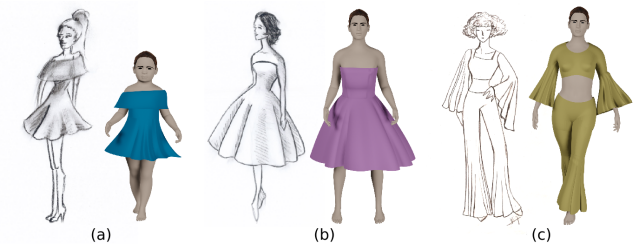


Figure 22: *Images used in our second user study as naive models to evaluate quality of the garments created with our method.*

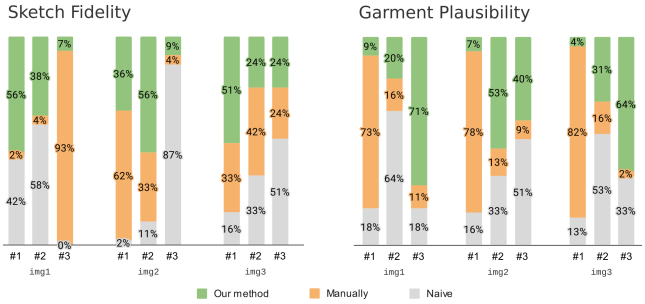


Figure 23: *Results of our second user study. For each sketch, users had to rank three 3D models based on fidelity to the sketch and plausibility of the garment. The graph displays the proportional distribution of rank for each sketch and each model evaluated.*

(`img3`(a), `img2`(a)). These variations lead to different resulting garments, which are both consistent with the intent given by the annotation.

In conclusion, all users (non-experts and non-artists) who experimented our system were able to annotate efficiently the sketches and obtained quickly a first virtual garment, which could be later refined by an artist if needed.

Quality

We also evaluate the results we obtain in terms of fidelity to the input sketch, and plausibility of the garment. To do so, we conducted a second study.
In this perceptual study, each of the 45 participants successively saw three sketches (`img1-3` in Fig. 18). For each sketch, the participant saw three images representing renderings of 3D models: one created with our method, one created manually by our experimented artist (Fig. 19), and another one - called *naive model* - generated with our method but using a slightly different sketch as input (Fig. 22). The result of our method was chosen randomly among three garments, two resulting from the user study (Fig. 23) and one resulting from annotations by the first author. The three models were displayed in a random order to avoid bias.
Participants had to rank the 3D models successively in terms of each of the following criteria:

- Fidelity to the associated sketch. A faithful model is one that visually represents the same garment as the one in the sketch.
- Plausibility of the garment they represent. A plausible model is one that represents a garment which could exist in real life.

We illustrate in Fig. 23 the results of this study. For each image and each task, we display the proportional distribution of rank.
Regarding sketch fidelity, we interpret results in each three images distinctively. For `img1`, our method obtains the best results (56% of 1st rank), followed by the naive solution (42% of 1st rank) which was obtained from a very similar sketch (Fig. 22.a). The manually-created model is ranked 3rd 93% of the time. For `img2`, the manually-created model is considered as the most faithful (62% of 1st rank), followed by our solution (36% of the 1st rank). The naive model here is obtained from a sketch a bit more different than `img2` (cf Fig. 22.b) and is considered as less faithful by 87% of the participants. For `img3`, our method is ranked as most faithful by 51% of the participants, followed by the manually-created model (33% of 1st rank). The naive model - here obtained from a sketch very similar to `img3` (cf Fig. 22.c), and ranks last in this criteria, the results being more balanced than in the first two cases (only 51% of the participants ranked it as the worst model).
Regarding garment plausibility, the result is homogeneous. The garment which was manually modeled using physics-based simulation is considered as most plausible for each set of images by more than 70% of the participants, although this higher plausibility is often achieved at the cost of deviating from the stylized input sketch.

Limitations and discussion

While our approach can efficiently synthesize folded garments over a large variety of characters, from a single annotated sketch, it is currently restricted to garments with planar borders (which is usually the case for real garments). The resulting 3D garment can be seen as an initial guess surface that could be further refined using another sketch-based approach, such as cutting out a part of it as to follow a non-flat border, if needed. The overall shape of the garment is also restricted to a surface of cylindrical topology due to the use of tensor product spline surfaces, which may lead to surface collision when modeling separate parts, such as sleeves and torso. To overcome this limitation of topological type, T-splines [SCF*04] could be considered.
This work also only addressed the capture and generation of tubular folds. If the user wants to add other folds to the garment, e.g. those caused by seams or notches, an interactive modeling tool [UKIG11] can also be used afterwards. However, these kind of folds need tailoring expertise, which we do not expect from users of our system.
As already mentioned, some tights parts were modeled manually using standard tools, e.g. junction of the pants, top-collar, etc, which correspond to more complex geometry. We did not focus our algorithm to handle these parts as they can already be well handled using existing approaches, e.g., [TCH07, RMSC11].
Finally, our approach does not guarantee physical plausibility of the garment shape such as taking into account fabric properties, nor its overall developability. As shown in the perceptual study, our result will be generally perceived as *less plausible* than a model fully generated using physical simulation. An extreme case of non-physical example is visible in the teaser with the little girl walking upside-down on her hands, therefore opposite to gravity direction, while

her garment still remains straight along her limbs. At the opposite, the lack of physical constraints allows to remain fidel to the reference sketch, even if the later is itself physically implausible, such as the blue dress of the manga girl from Fig. 21-img2 corresponding to human imaginary representations, possibly in contradiction with real physics constraints.

7. Conclusion and Future work

We presented a method to dress virtual character of various morphologies and poses from a single, annotated fashion sketch. This method extends style transfer methodology, usually defined between two surfaces in 3D, toward style transfer from a 2D sketch to a 3D model. To this end, we reformulated the definition of style as a set of four features that can be computed from a 2D drawing. In particular, we splitted the notion of shape preservation into preservation of silhouette on one side, and preservation of folds appearance on the other side. Our feature extraction method includes the completion of the occluded parts of folds, using a priori knowledge on the camera, extracted from garment design literature. This enabled us to correct for perspective distortions. As a result, our method can robustly synthesize plausible garments mimicking the style of the depicted sketch on various models ranging from human-like to cartoon-style characters. As discussed in the limitations, our approach takes the side of following the input sketch as much as possible, despite being associated to possible physical inaccuracies. However, various trade-off could be envisioned to increase the physical realism of the generated surface. Improving the developability of the meshed surface could be proposed using numerical optimization approaches [Wan08], while iterations can be stopped when a maximal deformation of the surface is reached. Plugging naively our resulting surface to a cloth simulator using standard parameters for in-plane and bending stiffness would deform the surface toward a physically-plausible result, but would not preserve the sketch and fold appearance when the action of gravity is suddenly applied to the vertices of the mesh. We believe that the use of inverse thin elastic shell modeling [LCBD*18] could allow a good trade-off between fidelity and physical-plausibility. Indeed, assuming that the sketch is depicted in static pose, inverse modeling would allow to compute stiffness parameters associated to equilibrium given our input surface. While these optimized parameters may be far from standard real-cloth fabrics stiffness, adjusting them toward more standard values should allow our surface to continuously vary from strong sketch fidelity to a more natural fabric appearance.

8. Acknowledgements

This work was supported in part by the ERC starting grant D^3 (ERC-2016-STG 714221) and by research and software donations from Adobe. This research was also partially funded by the ANR project Structures (ANR-19-CE38-0009-01).
We thank Laurence Boissieux for the help and the creation of 3D models. We also thank all the participants of the two user studies.

References

[BSBC12] BROUET R., SHEFFER A., BOISSIEUX L., CANI M.-P.: Design preserving garment transfer. ACM Trans. Graph. 31, 4 (2012). 2, 3, 4, 9, 10, 11

[BSK*16] BARTLE A., SHEFFER A., KIM V. G., KAUFMAN D. M., VINING N., BERTHOUZOZ F.: Physics-driven pattern adjustment for direct 3D garment editing. ACM Trans. Graph. 35, 4 (2016), 50–1. 2

[BWK03] BARAFF D., WITKIN A., KASS M.: Untangling cloth. ACM Trans. Graph. 22, 3 (2003), 862–870. 7

[Clo] CLO3D: Clo Virtual Fashion. URL: https://www.clo3d.com/. 2

[CSE*16] CORDIER F., SINGH K., ETEM E., CANI M.-P., GINGOLD Y.: Sketch-based modeling. In Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Tutorials (2016), EG '16, pp. 7:1–7:1. 2

[CSMS13] CORDIER F., SEO H., MELKEMI M., SAPIDIS N. S.: Inferring mirror symmetric 3D shapes from sketches. Computer-Aided Design 45, 2 (2013), 301 – 311. Solid and Physical Modeling 2012. 2

[DAI*18] DELANOY J., AUBRY M., ISOLA P., EFROS A., BOUSSEAU A.: 3D sketching using multi-view deep volumetric prediction. Proc. ACM on Computer Graphics and Interactive Techniques 1, 1 (2018). 3

[DBB*17] DVOROŽŇÁK M., BÉNARD P., BARLA P., WANG O., SỲKORA D.: Example-based expressive animation of 2D rigid bodies. ACM Trans. Graph. 36, 4 (2017). 3

[DDÖ*17] DANECEK R., DIBRA E., ÖZTIRELI A. C., ZIEGLER R., GROSS M.: Deepgarment : 3D garment shape estimation from a single image. Computer Graphics Forum (Proc. Eurographics) 36, 2 (2017). 3

[DJW*06] DECAUDIN P., JULIUS D., WITHER J., BOISSIEUX L., SHEFFER A., CANI M.-P.: Virtual garments: A fully geometric approach for clothing design. Computer Graphics Forum 25, 3 (2006), 625–634. 3

[DP73] DOUGLAS D. H., PEUCKER T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: the international journal for geographic information and geovisualization 10, 2 (1973), 112–122. 7

[DPS15] DE PAOLI C., SINGH K.: Secondskin: sketch-based construction of layered 3D models. ACM Trans. Graph. 34, 4 (2015), 126. 2

[FBR*17] FONDEVILLA A., BOUSSEAU A., ROHMER D., HAHMANN S., CANI M.-P.: Patterns from photograph: Reverse-engineering developable products. Computers & Graphics 66 (2017), 4–13. 2, 3

[FJL*16] FIŠER J., JAMRIŠKA O., LUKÁČ M., SHECHTMAN E., ASENTE P., LU J., SỲKORA D.: Stylit: illumination-guided example-based stylization of 3D renderings. ACM Trans. Graph. 35, 4 (2016). 3

[FLB16] FAVREAU J.-D., LAFARGE F., BOUSSEAU A.: Fidelity vs. simplicity: A global approach to line drawing vectorization. ACM Trans. Graph. 35, 4 (2016). 3

[GFL03] GROSSC., FUHRMANN A., LUCKAS V.: Automatic prepositioning of virtual clothing. In Proceedings of the 19th Spring Conference on Computer Graphics (New York, NY, USA, 2003), SCCG '03, ACM, pp. 99–108. 2

[GRH*12] GUAN P., REISS L., HIRSHBERG D., WEISS A., BLACK M.: Drape: Dressing any person. ACM Trans. Graph. 31, 4 (2012). 3, 7

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In of the 28th Annual Conference on Computer graphics and interactive techniques (2001), SIGGRAPH '01, ACM, pp. 327–340. 3

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3D freeform design. In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (1999), SIGGRAPH '99, pp. 409–416. 2

[JHR*15] JUNG A., HAHMANN S., ROHMER D., BEGAULT A., BOISSIEUX L., CANI M.-P.: Sketching folds: Developable surfaces from non-planar silhouettes. ACM Trans. Graph. 34, 5 (2015), 155. 2

[JYSL19] JIANG L., YE J., SUN L., LI J.: Transferring and fitting fixed-sized garments onto bodies of various dimensions and postures. Computer-Aided Design 106 (2019), 30 – 42. 3, 7

[JZH*20] 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 (Cham, 2020), Vedaldi A., Bischof H., Brox T., Frahm J.-M., (Eds.), Springer International Publishing, pp. 18–35. 3

[LCBD*18] LY M., CASATI R., BERTAILS-DESCOUBES F., SKOURAS M., BOISSIEUX L.: Inverse elastic shell design with contact and friction. ACM Transactions on Graphics 37 (2018). 15

[LKWS16] LUN Z., KALOGERAKIS E., WANG R., SHEFFER A.: Functionality preserving shape style transfer. ACM Trans. Graph. 35, 6 (2016), 209. 3

[LL14] LI J., LU G.: Modeling 3D garments by examples. Computer--Aided Design 49 (2014), 28 – 41. 2

[LPL*17] LI C., PAN H., LIU Y., TONG X., SHEFFER A., WANG W.: Bendsketch: modeling freeform surfaces through 2D sketching. ACM Trans. Graph. 36, 4 (2017), 125. 3

[LSGV18] LI M., SHEFFER A., GRINSPUN E., VINING N.: Foldsketch: enriching garments with physically reproducible folds. ACM Trans. Graph. 37, 4 (2018). 3

[LYW*10] LI J., YE J., WANG Y., BAI L., LU G.: Fitting 3D garment models onto individual human models. Computers & Graphics 34, 6 (2010), 742 – 755. 3

[LZY10] LIU Y.-J., ZHANG D.-L., YUEN M.: A survey on CAD methods in 3D garment design. Comp. in Industry 61 (2010), 576–593. 2

[Mar] MARVELOUSDESIGNER: Clo Virtual Fashion. URL: https://www.marvelousdesigner.com/. 2

[MWJ12] MENG Y., WANG C. C., JIN X.: Flexible shape control for automatic resizing of apparel products. Computer-Aided Design 44, 1 (2012), 68 – 76. 3

[NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Fibermesh: Designing freeform surfaces with 3D curves. ACM Trans. Graph. 26, 3 (2007). 2

[Opt] OPTITEX: EFI Optitex. URL: https://optitex.com/. 2

[OSJ11] OLSEN L., SAMAVATI F., JORGE J.: Naturasketch: Modeling from images and natural sketches. IEEE Comput. Graph. Appl. 31, 6 (2011), 24–34. 2

[OSSJ09] OLSEN L., SAMAVATI F. F., SOUSA M. C., JORGE J. A.: Sketch-based modeling: A survey. Computers & Graphics 33, 1 (2009), 85 – 103. 2

[PMPHB17] PONS-MOLL G., PUJADES S., HU S., BLACK M.: Clothcap: Seamless 4D clothing capture and retargeting. ACM Trans. Graph. 36, 4 (2017). 3

[RMSC11] ROBSON C., MAHARIK R., SHEFFER A., CARR N.: Context-aware garment modeling from sketches. Computers & Graphics 35, 3 (2011), 604–613. 1, 2, 5, 7, 14

[RPC*10] ROHMER D., POPA T., CANI M.-P., HAHMANN S., SHEFFER A.: Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles. ACM Trans. Graph. 29, 6 (2010), 157–157. 3

[SBS19] SMIRNOV D., BESSMELTSEV M., SOLOMON J.: Deep sketch-based modeling of man-made shapes. ArXiv abs/1906.12337 (2019). 3

[SCF*04] SEDERBERG T. W., CARDON D. L., FINNIGAN G. T., NORTH N. S., ZHENG J., LYCHE T.: T-spline simplification and local refinement. ACM Trans. Graph. 23, 3 (2004), 276–283. 14

[SKSK09] SCHMIDT R., KHAN A., SINGH K., KURTENBACH G.: Analytic drawing of 3D scaffolds. ACM Trans. Graph. 28, 5 (2009). 2

[SKv*14] SÝKORA D., KAVAN L., ČADÍK M., JAMRIŠKA O., JACOBSON A., WHITED B., SIMMONS M., SORKINE-HORNUNG O.: Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. ACM Trans. Graph. 33, 2 (2014). 2

[TCH07] TURQUIN E., CANI M.-P., HUGHES J. F.: Sketching garments for virtual characters. In Proc. Eurographics Workshop on Sketch-Based Interfaces and Modeling (2007), pp. 175–182. 2, 14

[TWB*07] TURQUIN E., WITHER J., BOISSIEUX L., CANI M.-P., HUGHES J. F.: A sketch-based interface for clothing virtual characters. IEEE Computer graphics and applications 27, 1 (2007). 1, 10

[UKIG11] UMETANI N., KAUFMAN D. M., IGARASHI T., GRINSPUN E.: Sensitive couture for interactive garment modeling and editing. ACM Trans. Graph. 30, 4 (2011). 2, 14

[VCMT05] VOLINO P., CORDIER F., MAGNENAT-THALMANN N.: From early virtual garment simulation to interactive fashion design. Computer-aided design 37, 6 (2005), 593–608. 2

[Wan08] WANG C. C. L.: Towards flattenable mesh surfaces. Computer-Aided Design 40, 1 (2008), 109–122. 15

[Wat09] WATANABE N.: Contemporary fashion illustration techniques. Rockport Publishers, 2009. 4, 10

[WCPM18] WANG T. Y., CEYLAN D., POPOVIĆ J., MITRA N. J.: Learning a shared shape space for multimodal garment design. ACM Trans. Graph. 37, 6 (2018). 3

[WHRO10] WANG H., HECHT F., RAMAMOORTHI R., O'BRIEN J. F.: Example-based wrinkle synthesis for clothing animation. ACM Trans. Graph. 29, 4 (2010). 3

[WLL*09] WANG J., LU G., LI W., CHEN L., SAKAGUTI Y.: Interactive 3D garment design with constrained contour curves and style curves. Computer-Aided Design 41, 9 (2009), 614 – 625. 2

[WWY03] WANG C. C., WANG Y., YUEN M. M.: Feature based 3D garment design through 2D sketches. Computer-Aided Design 35, 7 (2003), 659–672. 2

[XCS*14] XU B., CHANG W., SHEFFER A., BOUSSEAU A., MCCRAE J., SINGH K.: True2Form: 3D curve networks from 2D sketches via selective regularization. ACM Trans. Graph. 33, 4 (2014). 2

[YPA*18] YANG S., PAN Z., AMERT T., WANG K., YU L., BERG T., LIN M. C.: Physics-inspired garment recovery from a single-view image. ACM Trans. Graph. 37, 5 (2018). 3, 11, 12

Appendix A: Computing 3D ellipse parameters

We show in this appendix that the 3D parameters of an ellipse $\mathcal{E}$, namely its center $\Omega = (x_0, y_0, z_0)$ and normal vector $n = (n_x, n_y, n_z)$, can be retrieved from its projection $\tilde{\mathcal{E}}$ onto a 2D image.

As explained in Sec. 4.2, the 3D ellipse is supposed to lie in a plane orthogonal to the camera view direction, leading to $n_z = 0$. In addition, the center is supposed to be at a known distance d. In calling φ the angle such that $n = (\cos(\varphi), \sin(\varphi), 0)$, the 3D ellipse can be parameterized as

$$\forall t \in [0, 2\pi] \ , \ \mathcal{E}(t) = \begin{pmatrix} a\ \cos(\varphi)\cos(t) + x_0 \\ -a\ \sin(\varphi)\cos(t) + y_0 \\ -b\ \sin(t) + z_0 \end{pmatrix} \tag{9}$$

Its image under perspective transformation with focal distance f is

$$\tilde{\mathcal{E}}(t) = \frac{f}{b\sin(t) - z_0} \begin{pmatrix} a\cos\varphi\cos(t) + x_0 \\ -a\sin\varphi\cos(t) + y_0 \end{pmatrix} . \tag{10}$$

We show that this curve is an ellipse satisfy the following algebraic

equation

$$A\tilde{x}^2 + 2B\tilde{x}\tilde{y} + C\tilde{y}^2 + 2D\tilde{x} + 2F\tilde{y} + 1 = 0\,, \tag{11}$$

with

$$\begin{aligned}
A &= \frac{a^2(z_0^2 - b^2)\sin^2\varphi + b^2 y_0^2}{a^2 f^2 (x_0\sin\varphi + y_0\cos\varphi)^2} \\
B &= \frac{a^2(z_0^2 - b^2)\cos\varphi\sin\varphi - b^2 x_0 y_0}{a^2 f^2 (x_0\sin\varphi + y_0\cos\varphi)^2} \\
C &= \frac{a^2(z_0^2 - b^2)\cos^2\varphi + b^2 x_0^2}{a^2 f^2 (x_0\sin\varphi + y_0\cos\varphi)^2} \\
D &= \frac{z_0\sin\varphi}{f(x_0\sin\varphi + y_0\cos\varphi)} \\
F &= \frac{z_0\cos\varphi}{f(x_0\sin\varphi + y_0\cos\varphi)}
\end{aligned} \tag{12}$$

The 3D parameters of the ellipse can be expressed with respect to the coefficients A, B, C, D, F

$$\tan\varphi = \frac{D}{F}\,, \qquad \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \frac{z_0}{(\tilde{x}_0 D + \tilde{y}_0 F) f} \begin{pmatrix} \tilde{x}_0 \\ \tilde{y}_0 \end{pmatrix}, \tag{13}$$

where $\begin{pmatrix} \tilde{x}_0 \\ \tilde{y}_0 \end{pmatrix} = \frac{1}{B^2 - AC} \begin{pmatrix} CD - BF \\ AF - BD \end{pmatrix}$ is the origin of the projected ellipse $\tilde{\mathcal{E}}$.

Demonstration Let $c_\varphi = \cos\varphi$, $s_\varphi = \sin\varphi$.

$$\begin{aligned}
&(a^2 f^2 (x_0 s_\varphi + y_0 c_\varphi)^2)(A\tilde{x}^2 + 2B\tilde{x}\tilde{y} + C\tilde{y}^2) \\
&= a^2(z_0^2 - b^2)(\tilde{x}^2 s_\varphi^2 + 2\tilde{x}\tilde{y} c_\varphi s_\varphi + \tilde{y}^2 c_\varphi^2) + b^2(\tilde{x}^2 y_0^2 - 2\tilde{x}\tilde{y} x_0 y_0 + \tilde{y}^2 x_0^2) \\
&= a^2(z_0^2 - b^2)(\tilde{x} c_\varphi + \tilde{y} s_\varphi)^2 + b^2(\tilde{x} y_0 - \tilde{y} x_0)^2
\end{aligned}$$

Replacing $\tilde{x}$ and $\tilde{y}$ with their corresponding expression from Eq. (10) leads to

$$\begin{aligned}
&(a^2 f^2 (x_0 s_\varphi + y_0 c_\varphi)^2)(A\tilde{x}^2 + 2B\tilde{x}\tilde{y} + C\tilde{y}^2) \\
&= \frac{a^2 f^2}{(b\sin(t) - z_0)^2}(z_0^2 - b^2\sin^2(t))(x_0 s_\varphi + y_0 c_\varphi)^2 \\
\Rightarrow\ & A\tilde{x}^2 + 2B\tilde{x}\tilde{y} + C\tilde{y}^2 = \frac{z_0^2 - b^2\sin^2(t)}{(b\sin(t) - z_0)^2}
\end{aligned}$$

On the other side,

$$\begin{aligned}
&f(x_0 s_\varphi + y_0 c_\varphi)(2D\tilde{x} + 2F\tilde{y}) = 2z_0(\tilde{x} s_\varphi + \tilde{y} c_\varphi) \\
&= \frac{2z_0 f}{b\sin(t) - z_0}(x_0 s_\varphi + y_0 c_\varphi) \\
\Rightarrow\ & 2D\tilde{x} + 2F\tilde{y} = \frac{2z_0}{b\sin(t) - z_0}
\end{aligned}$$

Finally

$$\begin{aligned}
&A\tilde{x}^2 + 2B\tilde{x}\tilde{y} + C\tilde{y}^2 + 2D\tilde{x} + 2F\tilde{y} + 1 \\
&= \frac{z_0^2 - b^2\sin^2(t)}{(b\sin(t) - z_0)^2} + \frac{2z_0}{b\sin(t) - z_0} + 1 = 0
\end{aligned}$$