Interaction-free dressing of virtual humans

Arnulph Fuhrmann\textsuperscript{a,*}, Clemens Groß\textsuperscript{a}, Volker Luckas\textsuperscript{a}, Andreas Weber\textsuperscript{b}

\textsuperscript{a}Fraunhofer-Institut für Graphische Datenverarbeitung Fraunhoferstr. 5, 64283 Darmstadt, Germany
\textsuperscript{b}Institut für Informatik II, Universität Bonn Römerstr. 164, 53117 Bonn, Germany

Abstract

We describe an interaction-free method for a geometric pre-positioning of virtual cloth patterns with human 3D scans. Combined with the following physically based sewing simulation, a fully automated dressing of virtual humans becomes possible. This reduces the time needed for dressing virtual humans by eliminating the time needed for the interactive placement of the cloth patterns in previous methods, where the patterns have to be pre-positioned by user interaction. The result of our pre-positioning method is computed in less than 0.3 s CPU time for various examples on current PCs and designed to converge efficiently in the simulation step afterwards.

Keywords: Cloth modeling; Physically based modeling; Optimization for animation

1. Introduction

Modeling clothing and textiles has become a major topic in computer graphics research; see e.g. [1] for a survey up to 1996, [2] for a collection of some landmark papers up to 2000, [3] for a recent textbook, or [4,5] for some research papers from the last year.

On the application level, virtual clothing for various kinds of virtual actors has become a reality [6,7].

This has become possible because major advances have been achieved for the simulation times of virtual clothing. This involves simplified cloth models such as the ones used in [8,9], but also more accurate models such as the ones used in [10,4]. A comparison and survey of both approaches can be found in [11]. Also, the area of collision detection and response for cloth, another prerequisite for the clothing of virtual actors, has been the topic of intense research [8,12,7,13,3].

For the initial manipulation of the clothing patterns—placing them around a virtual actor and sewing them together—interactive techniques have been used [14,3]. While interactive manipulation of the patterns is a sufficient technique for some applications, such as virtual actors, it is not acceptable in other areas like a virtual clothing boutique, in which customers want to see the fit of several pieces of clothing on their virtual twins, i.e. figurines obtained from measurement data or 3D scans of individual customers. This boutique can reside in a real shop or even be realized as an internet application. In the latter case the customer may try on garments from home.

Major prerequisites for such a scenario in a virtual boutique exist. Methods of obtaining the geometric surface data of real people by 3D-scanning technologies are well established. Also large collections of CAD data of the patterns of “real clothing” exist, which can be stored in standard databases together with information on sewing, texture, material properties, etc.

However, no system has been described up to now that would allow the fully automated simulation of clothing on an individual figure by taking the clothing information out of a database without requiring any user interaction. The manipulation of the patterns before sewing them together at the right places near a figure required the interaction of a user in all systems that have been described in the literature. After the patterns are sewn together, the various physically based simulation systems then allow an automated simulation.
of virtual clothing in consecutive steps, like movements of the figurine, etc.

In this paper, we present an interaction-free method for pre-positioning the clothing patterns around a virtual figurine. Combined with the following physically based sewing simulation, this allows the first fully automated dressing of virtual humans (see Fig. 1 for an example). This enables clothing simulations on individualized human 3D scans e.g. in virtual clothing boutique scenarios as described above. We give an abstract description of the clothing patterns suitable for database storage and a uniform description for clothes of varying sizes, which can be used on different figurines.

By eliminating the time needed for the interactive placement of the cloth patterns in previous methods [14,3], where the patterns have to be pre-positioned by user interaction, our method reduces the time needed for dressing virtual humans. The result of our pre-positioning method is computed in less than 0.3 s CPU time for various examples on current PCs and designed to converge efficiently in the simulation step afterwards.

2. Preliminaries

2.1. Context of our work

We will assume that a 3D scan of an individual person is available, which is already segmented, and for which certain feature points (such as knees, hips, elbows, etc.) are marked. Several systems for these purposes are available and have been described in the literature [15]. Also, methods exist that allow the insertion of animation skeletons into the meshes generated from the 3D scans and to move some parts of the body, such as the arms, into new positions [16]. We can thus assume that the figurine obtained from the 3D scan is in a position suitable for sewing the clothing patterns together around it. These positions are similar to the ones required of a real person in order to allow a real tailor to appropriately sew real clothing patterns.

From the data stored in a clothing database, we need the CAD data of the patterns and some information on the seams: which parts of the boundary curves of one pattern have to be sewn to corresponding parts of other patterns. To function properly, our algorithm needs additional information. This information is added to the garments in the database one time and then can be reused for different 3D scans. What kind of information is needed will be topic of Section 4, when our algorithm for pre-positioning cloth patterns is described.

2.2. Physically based simulation of clothing

Currently, the standard approach to the physically based simulation of cloth is the use of particle systems [8,9,17,4,5,3,11]. The discretization of a piece of cloth into a system of coupled particles gives a system of ordinary differential equations of dimension six times the number of particles. The trajectories of the particles can be computed according to the laws of Newtonian mechanics by solving the initial value problem that consists of this system of differential equations together with a set of initial values for locations and velocities of the particles. The major problem in numerically solving this initial value problem is that it is a stiff problem [18]. This problem results from the fact, that—for cloth—the forces induced by stretching or compressing are much higher than the ones induced by bending or shearing. Moreover, the stretching forces sharply increase if the stretched distance is higher than a certain threshold, which is relatively small for woven fabrics and somewhat larger for most kinds of knitted fabrics [17] (Fig. 2).
When providing initial values, the locations of the coupled particles should be chosen in such a way that the internal forces are relatively small, i.e. the stretched distances between the particles should be zero or at least relatively small. As the bending of textiles results in much smaller forces, an initial bending of the textiles will cause little problem for a numerical solution procedure.

In almost all existing physically based cloth simulation systems, the chosen initial values are just the unstretched cloth patterns in a plane. There are no initial internal forces that have to be dealt with, but these trivial initial values require user interaction for pre-positioning.

In these systems, the patterns are arranged in a front plane of a virtual figurine and in a back plane [14,3], but no timings are given that were needed to provide these initial values of the cloth patterns. Notice that arranging cloth patterns in a front plane and a back plane of a virtual figurine is possible for artificial clothing, in which the form of the clothing patterns can be freely chosen, but is not possible in the context of real clothing, in which the patterns are predefined and very often cannot be split into some kind of front patterns and back patterns.

### 2.2.1. Sewing by elastics

When using particle system simulation, the particles on different patterns can be connected by some kind of “virtual elastics” in accordance with the seaming information in order to bring the patterns together along the seams, see e.g. [3, Chapter 4.3], to which we refer for more details.

Although the general idea of this method of *sewing by elastics* is relatively simple, there are many subtle points which have to be taken into consideration. For instance, one might want to use quite high virtual forces induced by these elastics in order to shorten the required simulation time for bringing the patterns together. On the other hand, the acceleration induced by the elastics usually has to be corrected due to occurring collisions, which limits the maximum acceleration and requires an appropriate damping, etc. Of course, the shorter the virtual elastics are, the easier the handling becomes for the simulator.

In principle, our technique can be used in combination with any of the physically based simulation techniques described in the literature: we just provide non-trivial initial values, which can then be used by all of the described methods of sewing by elastics and the solution methods of the initial value problem.

### 2.3. Developable surfaces

For the geometric pre-positioning of the patterns, we utilize developable surfaces. These are surfaces whose Gaussian curvature vanishes at every point. Therefore, they can be flattened into the plane without distortion, i.e. the length of any curve drawn onto them and the area of the surface remain the same after the mapping. Clearly, the reverse mapping has the same properties. The simplest examples of developable surfaces are cylinders and cones. An example of a non-developable surface is a sphere, because its curvature is constantly non-zero. Developable surfaces are widely used in computer graphics, e.g. for texture mapping and geometric design [19,20].

### 3. A two-step technique for dressing virtual humans

We address the problem of dressing specific figurines obtained from a 3D scan with virtual clothing. Each of these figurines is different from the others and therefore the pieces of cloth have to be positioned from scratch on each one. This might be the reason that all of the approaches described in the literature rely on interactive techniques.

In order to solve this problem automatically, a suitable abstract description of the garment patterns is needed. Some of the abstract information includes the boundary curves of patterns and how to connect different patterns, i.e the seams. This information is readily available, can be stored in standard databases, and does not cause major technical problems.

However, an abstract classification is also needed to move the patterns to the desired positions near the figurine. To this end, it seems possible—in principle—to use classifications of the patterns according to the standard naming of clothing, such as skirt, short, blouse, dress or trousers. However, it seems very tedious to use this classification to accomplish our proposed task.

Instead, we take a more general approach that is independent from the kind of clothing. The information that we need in order to preposition the panels around the body are just the *relative positions with respect to the body*. This information can be quite similar for different kinds of clothing, e.g. the panels of a shirt and a jacket.
will have the same information. For our abstract description, we use a suitable classification of body segments and assign the garment patterns to the body segments.

3.1. Geometric pre-positioning

The pre-positioning algorithm has to provide the positions of the cloth patterns that can serve as initial values for the physically based sewing process. Thus, the patterns have to be arranged in a way that ensures a correct positioning of the garment after an appropriate sewing process and there should be no intersection between the cloth patterns and the 3D scan. In order to reduce simulation time, the distance between the seams of patterns that have to be sewn together should be as short as possible. The pre-positioning algorithm is explained in more detail in Section 4.

3.2. Physically based end-positioning

After the pre-positioning, we drape the panels by a physically based method based on the work described in [9]. In order to bring the panels together, we use elastics which were introduced in [13]. These elastics provide forces to bring together two particles along a seaming line. In some cases, e.g. when a pattern is not entirely wrapped around a body segment, this method can fail (see top of Fig. 3). Therefore, we extended the original idea by introducing auxiliary sewing particles. In order to pull together consecutive sewing particles and the two cloth particles, elastics are used. If collisions between auxiliary particles and the body are checked, a correct sewing simulation is guaranteed (see bottom of Fig. 3). The initial placement of the auxiliary sewing particles is described in Section 4.6, since it is closely related to our pre-positioning algorithm.

During simulation, the distance between the auxiliary and cloth particles is decreasing. Hence, two particles are merged if their distance is below a certain threshold. If some particles could not be merged, this is a hint that the garment does not fit. In this case, the user can decide to increase the forces provided by the elastics to counteract the internal cloth forces. This would stretch the panels more than would be natural, but the remaining particles could be merged. A better choice for the user would be to try on a larger garment.

The physically based simulation also has to handle self collisions and collisions between the figurine and the cloth. Well-known techniques exist for these purposes, cf. Section 2.2. We have nothing substantial to add to the topic of collisions detection and collision response for cloth per se. By imposing penalty forces to cloth triangles that have already penetrated the figurine, it is even possible to correct small penetrations of the clothing patterns in the figurine imposed by the geometric pre-positioning. Clearly, any other of the well-known physically based techniques for sewing by elastics and simulation of the virtual clothing can be used in conjunction with our geometric pre-positioning.

4. Geometric pre-positioning

4.1. Classification of body segments

We divided the human body into segments suitable for wrapping pieces of cloth around it. Looking at a human body and common clothing, we found that the following segments and their unions are sufficient:

- upper and lower arm (left and right),
- upper and lower torso,
- upper and lower leg (left and right),
- neck, and
- hands, feet and head.

These segments correspond to the skeleton segments commonly used for the animation of human characters. It is not a mere coincidence, but can be explained by the fact that parts of the body, which can be treated as a rigid body in a simple animation system, do not have to be subdivided for the purpose of pre-positioning clothing.

Remark. For some special kind of clothing, such as gloves, a finer classification, e.g. mentioning different fingers, might be necessary. Our approach is general enough so that these special cases can be added later.

The information concerning which specific pattern of cloth belongs to which body part is stored in the database. We will map a piece of cloth to one bounding surface, which surrounds a corresponding body segment. These surfaces enclose the body as tightly as possible and must allow the correct placement of the piece.

4.2. Determining bounding surfaces

For each body segment, we construct a tightly fitting developable bounding surface. To accomplish this task,
we start by determining the principal axis of the segment. We point-sample the surface uniformly and calculate the axis as a solution of a least square fit to these points. Let \( l(t) = s + \lambda \cdot t \) with \( s, t \in \mathbb{R}^3 \) and \( \lambda \in \mathbb{R} \) be the line to be determined and \( p_i \in \mathbb{R}^3, i = 1, \ldots, N \) be the sampled points. In order to fit \( l \) to the \( p_i \), we minimize

\[
E(s, t) = \sum_{i=1}^{N} \left( s + \left( \frac{\langle t, p_i - s \rangle}{\langle t, t \rangle} \right) t - p_i \right)^2.
\]

(1)

Thereafter, \( l \) is capped to a line segment by two orthogonal planes to restrict its length to the extent of the body segment. Then, we project the \( p_i \) along this line and compute the planar convex hull. The polygon defined by the \( p_i \) on the border of the convex hull is interpolated by a spline curve to guarantee smoothness. This curve serves as a basis for our bounding surface by translating it to both ends of the line. The resulting two base curves span a general cylinder, which is the desired bounding surface (Fig. 4).

Often, it is necessary to expand one of the base curves or even both of them to get a proper fit of several pieces of clothing. Therefore, we have chosen the parameterization of a general cylinder by two curves. This allows us to expand each of the curves independently. The cylinder is transformed into a cone by taking the intersection between the plane in which the curve lies and the principal axis of the body segment as the center for a homothetic expansion.\(^1\) With this transformation, we remain in the realm of developable surfaces.

4.3. Mapping between plane and surface

The motivation for using developable surfaces for a geometric pre-positioning is based on the observation that internal forces due to stretching are quite high, cf. Section 2.2. Therefore, we want to avoid stretching of the cloth during the pre-positioning process. By mapping onto developable surfaces, we treat it like paper and it is clear that in this case no stretching or shearing occurs since the area remains constant. Bending forces are much lower and will not cause problems for the subsequent physically based techniques, as long as the curvature of the bounding surfaces remains below a certain threshold, which is easily achieved in our application.

The usual way to flatten a developable surface involves the solving of a system of differential equations \([20]\). As efficiency of the overall pre-positioning process is of importance, we did not choose this method, but rather used a discretization into a triangular mesh. Developable surfaces have the useful property to preserve angles and edge lengths. Knowing this, we start with the first triangle and place it at an arbitrary position in the plane. Then, we take a neighboring

\(^1\)Let \( \mathbf{b}(u) \) be a point on the curve, \( \mathbf{c} \) the center and \( \lambda \) the expansion factor. Then, a point \( \mathbf{b}(u) \) on the transformed curve can be obtained by \( \mathbf{b}(u) = \lambda \cdot (\mathbf{b}(u) - \mathbf{c}) + \mathbf{c} \).
triangle \( t \) and map one of its edges into plane according to Fig. 5(a).

Consequently, the other edge of \( t \) is fixed and we can go for the next triangle. The overall approximation error induced by the discretization can be controlled by the number of triangles used. In practice, several hundred triangles proved to be sufficient. See Fig. 5 for the example of the flattening of an elliptic cone.

After the surface is flattened, we establish a mapping between plane and bounding surface. In order to map a point, we resolve its barycentric coordinates in the according triangle in plane and apply them on the corresponding triangle of the bounding surface. This method allows a fast projection of patterns of cloth onto the bounding surface. Clearly, an approximating error occurs by treating the cloth as a triangulated surface and mapping just the points of it. But this error, i.e. the stretching in some areas, remains small and will cause no problem for the subsequent physically based simulation: It is not necessary that no stretching at all occurs during pre-positioning; it is only important that this stretching remains quite limited, cf. Section 2.2.

Since we map only vertices of the surface, some faces may intersect with the body segment. In order to alleviate this problem, we initially stretch the base curves of our bounding surface. The stretching factor depends on the largest edge of the triangulation. Possible remaining small intersections are then handled by collision detection routines of the physically based end-positioning. A more sophisticated stretching factor could be derived if the curvature of the bounding surface is also taken into account. But this would lead to an increased distance from the body, which would increase the necessary computation time in the physically based end-positioning, and did not offer any advantages in practice.

4.4. Solving the pattern puzzle

As described above, each pattern is assigned to a body segment and a bounding surface is generated around each segment. The remaining task is to determine the correct positions of all patterns on the flattened bounding surface. Then, they are enveloped using our previously defined mapping. As a consequence, the patterns are placed near the figurine and serve as good initial values for the physically based sewing by elastics and further simulations, cf. Section 2.2.1.

The first task that we have to accomplish is the one of an initial placement of the patterns. To be able to arrange these patterns automatically, we add some additional information to our garment database:

- An orientation for each pattern (i.e. which side is visible after dressing).
- The assigned body segment.
- A vector, which points in the direction of the principal axis of the bounding surface of this pattern.
- For one pattern only: the position of one feature point in relation to this pattern.

This information can be added manually to the database or generated by an appropriate garment design tool. Clearly, this has to be done only once independent of the size of the virtual figurine. Some of this information might be specific to our method, but most of it is necessary to ensure the intended solutions, as real world examples show: e.g. a skirt can be rotated around the hip, which necessitates the correspondence to a feature point. The other information is needed by our algorithm to place the patterns at correct locations.

In the following, we first describe which feature points are used for which bounding surfaces and which garments. Thereafter, we present an algorithm for arranging patterns around the pattern by using the seaming information (feature point information is provided). Finally, we describe how to adapt the spacing between these patterns to solve the problem of an intersection-free placement around different sized people resp. bodies.
4.4.1. Feature points

In order to find the correct position of one initial pattern on the flattened bounding surface, we project the relevant feature points of this body segment onto the bounding surface and flatten them, as well. In principle, only one feature point is necessary. The projection of a feature point \( p_f \) is done by first finding the nearest point \( p_n \) on the principal axis and then intersecting the ray starting at \( p_n \) and pointing towards \( p_f \) with the bounding surface. Since the principal axes are lying in the center of the body segments, the feature points are projected to the correct sides of the bounding surface. In both cases, we use an overview of the feature points used and their allocation to the body segments. For the generation of the feature points, see [15].

The number of correspondences between cloth panels and feature points depend on the type of garment and on how many bounding surfaces are necessary to position the patterns. Exactly one correspondence is needed for every bounding surface. For example, the patterns of a skirt are mapped to two different bounding surfaces and therefore two correspondences must be inserted into the database. The waistband is mapped onto the lower torso and the rest are mapped to the union of the lower torso and the legs. This splitting is reasonable, since considering only the lower torso for the waistband results in a closer fitting bounding surface. In both cases, we use the waist girth point lying centered at the back of the body as the feature point. For trousers, we use the same feature point for the waistband and the knees as feature points for the panels of the legs.

4.4.2. Arranging the patterns

We begin with one of the patterns, for which some additional information is stored in the database, namely the one containing position and orientation of the pattern according to some feature point. Obtaining a general solution to this problem seems to be a quite difficult task. Therefore, we consider a simplified problem, which covers a wide range of practical examples. First, we determine a geometrical ordering of the patterns using the seaming information. Then, we begin with the starting pattern and fit the next one to it. After that step, the third pattern is fitted to the first and second. This process is repeated for the remaining patterns. Using this method, we loose generality and we are not able to handle arbitrary clothing, but we can give a fast algorithm suitable for many garments.

The fitting is carried out by moving the corresponding borders of two panels as closely as possibly together. After sewing, these borders become a seam. We note that the two borders may have different geometry, since this is often predetermined by the design of the garment. To bring the two borders together, we determine the line of regression for each of them (left of Fig. 6). The new panel is rotated so that the two lines are parallel. Now, we move the new panel perpendicular to the line of regression of the border towards the old panel until a collision occurs or the two lines match (right of Fig. 6). Collision detection is carried out between the new pattern and the patterns which are already arranged.

4.4.3. Adaptation of the initial bounding surfaces

In most cases, it becomes necessary to adapt the initial bounding surface, because the patterns will not fit into it. This problem commonly occurs for garments which have many folds, like a skirt or, more generally spoken, when its volumetric form resembles a cone. As was mentioned above, the bounding surfaces are parameterized by two base curves, which can be stretched to enlarge the surface. We use this feature to adjust the size iteratively until the patterns are fitting into the boundary of the surface, cf. Fig. 7.

We start with the patterns arranged closely together side-by-side. One seam limits the left side and one seam the right. After being mapped onto the bounding surface, these two seams will be connected. Two cases have to be considered: If the patterns connect with the bounding surface, we have to enlarge the bounding surface. We can approximate the upper and lower length of the arranged patterns measured between the limiting seams along the boundary curves of the patterns. These two measurements correspond to circumferences on the bounding surface. Knowing this value yields an estimate for the length of the two base curves of the bounding surface, which in turn can be made larger. In the second case, the limiting seams are too far away from the border of the bounding surface and we increase the spacing between the patterns. This step is done by leaving a gap between the patterns in the algorithm described in 4.4.2.

### Table 1

<table>
<thead>
<tr>
<th>Feature point</th>
<th>Body segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neck (L,R)</td>
<td>Neck</td>
</tr>
<tr>
<td>Neck front</td>
<td>Neck</td>
</tr>
<tr>
<td>Neck back</td>
<td>Neck</td>
</tr>
<tr>
<td>Biceps (L,R)</td>
<td>Upper arm</td>
</tr>
<tr>
<td>Elbow (L,R)</td>
<td>Upper and lower arm</td>
</tr>
<tr>
<td>Wrist (L,R)</td>
<td>Lower arm, hands</td>
</tr>
<tr>
<td>Waist girth point</td>
<td>Lower torso</td>
</tr>
<tr>
<td>Hip (L,R)</td>
<td>Lower torso</td>
</tr>
<tr>
<td>Nipple (L,R)</td>
<td>Upper torso</td>
</tr>
<tr>
<td>Ankle inseam (L,R)</td>
<td>Lower leg, feet</td>
</tr>
<tr>
<td>Knee (L,R)</td>
<td>Lower and upper leg, lower torso</td>
</tr>
</tbody>
</table>
The number of iterations needed depends mostly on the final form of the bounding surface. The more conical it is the more iterations are done. The patterns of the skirt from our examples are adapted in seven iterations, whereas the shirt patterns already fit after four iterations.

**Input:** initial bounding surface, arranged patterns  
**Output:** patterns lying on bounding surface

\[
B = \text{initial bounding surface} \\
B_f = \text{flattened } B \\
\text{repeat} \\
\quad \text{if patterns intersect with } B_f \text{ then} \\
\qquad \text{increase circumference of base curves of } B \\
\qquad \text{if circumference exceeds threshold then} \\
\qquad \quad \text{modulate base curves, cf. Section 4.5} \\
\qquad \text{end if} \\
\quad \text{end if} \\
\text{else if} \text{ limiting lines are too far away from border of } B_f \text{ then} \\
\quad \text{increase spacing of patterns} \\
\quad \text{rearrange patterns} \\
\text{end if} \\
\text{until} \text{ patterns are lying evenly spaced inside } B_f
\]

Algorithm 1: Adaption of bounding surfaces and pattern spacing.

4.5. Limiting the extent of bounding surfaces

In some cases, the expansion of the bounding surfaces causes collisions with other body segments and the pre-positioned patterns are far away from the enclosed body segment. The adapted bounding surface for a wide skirt, for example, would be a wide open cone. To limit the extent of this bounding surface we developed the technique of modulation of the original base curves with sine functions. The amplitude and the number of oscillations depend on the desired arc length of the base curve. In Fig. 8, the two forms are shown.

The bounding surface with a modulated base curve is more compact by nearly having the same surface area as the original one. Both lower base curves have the same arc length. One minor drawback of this method is that this surface is no longer a developable surface. But our algorithm for flattening developable surfaces can still be applied to these surfaces and, in practice, the distortions of the pattern mesh are still quite small. Remember that we only have to ensure that the initial stretching between particles is quite small, but does not have to be 0. An example of a wide skirt on a modulated bounding surface is given in Fig. 9.
4.6. Generating the auxiliary sewing particles

This section describes the calculation of the auxiliary sewing particles. These particles have to be computed if the physically based sewing simulation with elastics—taken as the direct connection lines between two seams—would not converge to the desired results (see Fig. 3). In order to avoid testing these cases, we generate these particles for every elastic. This is possible, since the computation time is very short. Experiments showed that it is sufficient to compute just one auxiliary sewing particle for each elastic. However, the computation procedure can be easily adapted for generating more than one particle. In these cases, where the two corresponding patterns, which need to be sewn together, are lying on the same bounding surface directly side-by-side, the parameterization of the bounding surface enables us to easily calculate a particle on the bounding surface in the middle between the two endpoints of the elastic. For almost all of the rest of the seams, it was our experience that it is enough to take the middle point of the direct connection line, e.g. seams between arm patterns and upper body patterns at the armpit or seams between the right and the left leg in case of trousers. The sole exception are seams lying along the line between the neck and the shoulders. In these cases, some sewing particles may lie inside the figurine if the positions of the upper body patterns are too far down. For these cases, we generated virtual shoulder–neck lines lying above the scan onto which the particles can be projected.

4.7. The pre-positioning algorithm

A summary of our method in algorithmic pseudocode is given in Algorithm 2. Various steps of the algorithm with different input parameters have been visualized in Fig. 10.

Input: 3D body scan marked with feature points, CAD data of a garment with classification and seaming information
Output: pre-positioned patterns
divide figurine into segments \( s_i \)
for all \( s_i \) do
compute minimal bounding surfaces \( B_i \)
end for
assign each pattern to one \( s_i \)
for all \( s_i \) do
repeat
project feature points onto \( B_i \)
flatten \( B_i \) together with the feature points
place all corresponding patterns in plane
adapt \( B_i \)
until all patterns fit into the flattened \( B_i \)
envelope patterns
end for
generate auxiliary sewing particles and elastics

Algorithm 2: The main steps of our pre-sewing algorithm in pseudocode.

5. Computational results

In Fig. 10, several examples of the pre-positioning of different garments on various 3D scans are given. Each of the 3D scans of real people shown consists of approximately 17,000 triangles. This means a fine sampling of the original body surface is used. Notice that we have restricted the reproduced 3D scans to the ones of male persons for reasons of decency, although we use patterns of skirts in some examples. For the given examples, no normalization of the 3D scans was necessary (cf. Section 2.1), as would have been the case for other examples, such as jackets, in order to sew long sleeves on the arms.

Table 2 gives some timings for our the geometric pre-positioning algorithm on an AMD Athlon 1.3 GHz. Notice that the influence of the complexity of the 3D scan is very small. This is due to the fact that the vertices of the 3D scan are only needed for the computation of the body segments’ bounding surfaces. The time required for the segmenting of the 3D scan is not included, since the segmentation was given as a precondition of the algorithm. For segmenting the 3D scan with 17,000 resp. 60,000 vertices, we needed 260 ms resp. 600 ms computation time. The required cpu time of our pre-positioning algorithm was less than 0.3 s for all examples shown on a 1.3 GHz Athlon PC. This shows that our pre-positioning method is faster than any interactive technique can be. The physically based end-positioning was done in about 7–16 s, depending on the complexity of the garment. The implementation is done in Java and uses Java3D for the visualization of the results.
6. Conclusions and further research

For a variety of examples, cf. Fig. 10, our current implementation worked reliably. However, clothing patterns can be quite complicated and improving the robustness of our method will be the topic of further research.

One possible problem is that intersections between panels of cloth lying on different bounding surfaces are not handled by the geometric part of our pre-positioning system. Therefore, this task must be carried out during the physically based simulation. Although it is not yet included in our implementation, there is an algorithm for resolving incorrectly oriented surfaces of cloth [21,13]. This collision detection routine is able to correct interpenetrations of cloth panels by detecting so-called contact regions. Each region is then labeled as being on the wrong or the right side. Finally, the correct
orientation is computed by the choice of the majority. In our case, only small parts of cloth are likely to intersect and, therefore, can be detected as regions, which are on the wrong side.

Another point of research is the extension of our algorithm to support more types of clothing. Currently, we can handle patterns that fit into the topology of our bounding surfaces. Although this allows us to handle common types of clothing, there are examples, which do not fall into this category. For instance, patterns with initial folds (like collars) cannot be pre-positioned at the moment. One possible solution could be an artificial cutting of problematic patterns.

The major application of our work can be seen in the context of a virtual clothing boutique as described above, but it also has major benefits for the design phase of “real clothing” for “real people”. As our technique—in addition to the sine qua non in the context of a virtual boutique—considerably shortens the developing cycle of designing patterns, sewing them together, and viewing them on individual persons, we hope that it will facilitate the task of an average person, as well as an expert, designing real clothes for real people and this overall problem can be seen as the "Holy Grail" of cloth modelling [2, p. 332].

We believe we have taken a measurable step towards this goal. However, many subsequent steps have to be taken and several problems have to be resolved, such as the ones arising from multiple layers of clothing. Our method seems to be suitable for extensions that handle such additional requirements without sacrificing the goal of an animated procedure requiring no or little user interaction. For instance, by assigning a “dressing order” to different pieces of clothing—a piece of information that can easily be stored in the pattern database—an iterative version of our method should work for this purpose, or by the use of different bounding surfaces—corresponding to the same body segment—lying one upon the other.

Acknowledgements

We thank our Virtual Try-On project partners for providing example input data for our application (www.virtualtryon.de). The 3D scans were provided by tecmath AG, the 2D garment patterns and seam information by the Hohensteiner Institutes. Furthermore, we are very grateful to our partners at the University of Tübingen for helpful discussions about cloth simulation.

This work was partially supported by the bmb+f (Bundesministerium für Bildung und Forschung) Virtual Try-On grant.

References


Table 2
Timings of the geometric pre-positioning algorithm

<table>
<thead>
<tr>
<th>Vertices of garment</th>
<th>3D-scan with 17 000 vertices (ms)</th>
<th>3D-scan with 60 000 vertices (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>180</td>
<td>190</td>
</tr>
<tr>
<td>5000</td>
<td>210</td>
<td>220</td>
</tr>
<tr>
<td>10 000</td>
<td>250</td>
<td>260</td>
</tr>
</tbody>
</table>


