

Virtual Garments: A Fully Geometric Approach for Clothing Design

Philippe Decaudin^{1,3} Dan Julius⁴ Jamie Wither^{1,3} Laurence Boissieux³ Alla Sheffer⁴ Marie-Paule Cani^{1,2,3}

¹EVASION[†] ²INPG ³INRIA, Grenoble, France

⁴University of British Columbia, Vancouver, BC, Canada



Figure 1: From sketch to garment. Left to right: sketched contours and darts; 3D shape computed using distance field; piecewise developable surface; final virtual garment, compared with the real one sewn from the 2D patterns we output.

Abstract

Modeling dressed characters is known as a very tedious process. It usually requires specifying 2D fabric patterns, positioning and assembling them in 3D, and then performing a physically-based simulation. The latter accounts for gravity and collisions to compute the rest shape of the garment, with the adequate folds and wrinkles.

This paper presents a more intuitive way to design virtual clothing. We start with a 2D sketching system in which the user draws the contours and seam-lines of the garment directly on a virtual mannequin. Our system then converts the sketch into an initial 3D surface using an existing method based on a precomputed distance field around the mannequin. The system then splits the created surface into different panels delimited by the seam-lines. The generated panels are typically not developable. However, the panels of a realistic garment must be developable, since each panel must unfold into a 2D sewing pattern. Therefore our system automatically approximates each panel with a developable surface, while keeping them assembled along the seams. This process allows us to output the corresponding sewing patterns.

The last step of our method computes a natural rest shape for the 3D garment, including the folds due to the collisions with the body and gravity. The folds are generated using procedural modeling of the buckling phenomena observed in real fabric. The result of our algorithm consists of a realistic looking 3D mannequin dressed in the designed garment and the 2D patterns which can be used for distortion free texture mapping. The patterns we create also allow us to sew real replicas of the virtual garments.

Keywords: *Geometric modeling of garments, developable surfaces, procedural models, buckling.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computing Methodologies / Computer Graphics]: Surface representations, I.3.7 [Computing Methodologies / Computer Graphics]: Three-dimensional graphics and realism

1. Introduction

Within the past few years, much effort has been devoted to enhancing the realism of virtual humans. These characters are essential in both feature films and real-time applications. In addition to realistic bodies, faces, and hair, designing convincing clothing is a crucial element in achieving

character realism. The modeling of virtual garments remains time consuming and requires both technical and tailoring expertise. Even using specialized software, such as Maya-Cloth [May04], the user needs to perform challenging tasks such as designing sewing patterns and using physically-based simulation to obtain the garment rest shape.

A much simpler approach was recently proposed by Turquin et al. [TCH04]. It allows the users to sketch garment

[†] EVASION is a joint project of CNRS, INPG, INRIA, Univ. Joseph Fourier.

- Additional materials available at <http://evasion.imag.fr/Publications/2006/DJWBSC06>

contours directly onto a 2D view of a mannequin. A distance field around the mannequin is then used to compute a 3D surface. The method generates visually appealing clothing in only a few seconds. Unfortunately it does not achieve the level of realism a garment designer would like.

First, real garments are always piecewise developable, as they are made of flat fabric patterns sewn together. However, the surfaces produced by the sketching software are typically not developable, as the generated surface is basically an offset of the mannequin's body, taken at an adequately varying distance. Furthermore, since the surface is not piecewise developable, it is not possible to unfold it into the plane without distortion to obtain the actual 2D sewing patterns necessary to make the garment. Even if the garment is not intended to be produced in the real world, these patterns are necessary for distortion-free texture mapping. Lastly, the method generates a smooth surface while real garments, even at rest, always fold around the body due to collisions and gravity.

Our paper presents an intuitive, fully geometric method for generating virtual clothing. The modeled garments consist of developable surface panels that wrap around the mannequin in a natural manner forming visually realistic folds. The system also provides actual sewing patterns, which can be used for distortion-free texture mapping and for tailoring of real-life replicas of the designed garments.

Our method uses the intuitive sketch based interface, proposed by Turquin et al. [TCH04], to generate an initial 3D surface. The user draws the silhouette of the garment directly on a 2D view of the virtual mannequin. This sketch is converted into a 3D surface using the distance field technique [TCH04]. The difference from the original approach is that the sketching system is augmented to allow the user to specify additional seam-lines on the garment by sketching them on the surface. Given this input, our system uses a novel technique to automatically approximate each surface panel enclosed by seams and borders with a developable surface, while keeping the panels assembled along the seams.

The developable approximation takes only the geometric constraints into account, while the shape of a real garment is also affected by the physical surroundings. A garment worn by a character always has folds, even in a rest position, due to collisions and gravity. Additional folds are created when the wearer's limbs are bent or twisted. We provide a mechanism for modeling the folding using a simple and efficient procedural approach that does not require the user to provide the controls necessary for a complex physical simulation. We start from the key observation that most garments, including trousers, skirts, dresses, and sweaters, are made from pieces of fabric wrapped around roughly cylindrical parts of the human body. This observation allows us to exploit a priori knowledge on the manner in which fabric folds and drapes when wrapped around a cylinder. The resulting, very specific buckling patterns are modeled and controlled procedurally, providing a visually natural shape for the worn garment. This

allows us to validate our garment design system by showing that the virtual models we create closely resemble the real garments made from sewing patterns computed by unfolding the developable surface panels into the plane.

The rest of the paper is organized as follows. Section 2 reviews relevant prior work. Section 3 reviews the sketching system we use, enhanced by the ability to draw seam-lines and darts. Section 4 presents our method for generating a piecewise developable surface from the rough 3D shape created from the sketch. Section 5 studies the different buckling modes of cylindrical cloth swatches and uses the results to set up a procedural model for garment folds. Section 6 describes the application of this model to the sewing patterns computed for the garments, and demonstrates the modeled natural looking clothing containing realistic folds. Section 7 discusses results and future work.

2. Related work

Geometric modeling of garments: As pointed out, the standard way of designing realistic virtual garments [May04] is very similar to the sewing techniques used in the real world. It begins with the definition of all the necessary 2D fabric patterns, thus requiring knowledge of tailoring that is not necessarily possessed by most computer artists. The users then need to specify sewing constraints for the patterns. This tedious process is made even more complex by the need to set adequate values for a set of physical parameters and run a simulation to obtain the final shape for the garment, even when the character is at rest. The alternative is to model the garments directly in 3D, as proposed by [TCH04], who developed a sketch based garment modeling tool. However, using direct modeling in 3D, the generated shape is typically not piecewise developable, and thus depicts a garment which is not physically plausible. Furthermore, the modeled garment is unlikely to depict fold patterns specific to real garments in any rest position. Our work enhances this alternative approach by making it usable as a starting point for modeling the static shape of realistic garments. In particular, our method approximates the garment surface mesh obtained from a sketch by a piecewise developable surface.

Modeling with developable surfaces: The most widely known characteristic of developable surfaces is that the Gaussian curvature at any point on the surface is zero. Another characteristic, used in projective geometry, is that both the normal map of a developable surface on the unit sphere and its Blaschke image are one-dimensional [PW01, Pet04]. Developable approximation is typically performed using one of these characteristics [WT04, WWY04, PW99, CLL*99, Pet04].

Much of the existing approximation research [WWY04, PW99] focuses on developable approximation of free-form input surfaces. Since the methods typically manipulate the polynomial or rational descriptions of the surfaces, they have no clear extension to the approximation and modeling of

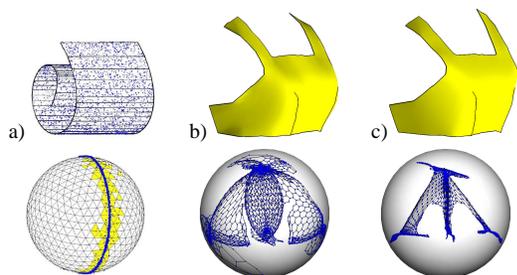


Figure 2: (a) Developable ruled surface approximating a set of points and the corresponding normal maps [Pet04]. (b) Typical input to our approximation. The normal map shows the triangle normals and the dual graph edges between the triangles. (c) Approximation output, the normal map is close to one dimensional.

mesh surfaces. Developable approximation of point clouds [CLL*99, Pet04] is typically performed using the projective geometry surface characteristics. These methods consider only ruled developable surfaces, that is surfaces for which the normal map is a 1D curve. They work well when the input is very close to developable, namely when the Blaschke image or the normal map (Figure 2a) of the input are very close to one-dimensional. In our case the input meshes have non-negligible Gaussian curvature, and their normal maps often cover a large area of the normal sphere (Figure 2b).

Wang and Tang [WWY04] increase NURBS surfaces developability by minimization of Gaussian curvature. They then extend this method to the mesh setting by approximating the Gaussian curvature as the difference between the sum of angles around each vertex and 2π [WT04]. Section 4.5 presents a comparison of results generated by our method with those of [WT04].

Folds and wrinkles: The generation of folds and wrinkles on virtual clothing resulting from collisions and gravity is generally performed using physically-based simulation; see for instance [CK02, BMF03, BWK03] for recent work on cloth animation. However, Choi and Ko [CK02] stressed that most physically-based cloth models are ill-conditioned in buckling situations, when specific folds should appear, such as when a sleeve bends. They use knowledge about the buckling process to solve the numerical problem. Bridson et al. [BMF03] note that collision response with the character's body may remove most of the folds and wrinkles generated by the simulation process. They propose a collision handling method better suited to preserve them.

A few geometric methods were introduced for generating wrinkles on virtual garments. In their groundbreaking work, Kunii and Gotoba [KG90] use a physically-based simulation as a reference to capture the geometric shape of cloth wrinkles and then model the wrinkles procedurally. Tsopelas [Tso91] relies on the buckling features of thin-walled structures [GS66, Den76, AH86] to generate wrinkles. The method, although limited to quasi-static diamond buck-

ling patterns illustrated on a pant leg, was the motivation for our approach. Hadap et al. [HBVMT99] and Larbourlette et al. [LC04] take into account constant length constraints along curves to generate sinusoidal wrinkles on clothes, while Kimmerle et al. [KWH04] use a strain measure to generate similar wrinkles. However, none of these methods consider the specific buckling features of cloth wrapped around cylindrical body parts. Our approach is primarily geometric. Like Tsopelas [Tso91], we rely on the well studied buckling properties of cloth to set up a solution. Contrary to previous approaches, our method is not limited to either sinusoidal or diamond-shape wrinkle patterns. Instead, it creates general fold shapes by locally combining the effects of the different active buckling modes, which generally include twisting, sinusoidal folds and diamond buckling.

3. Sketching a rough garment surface

Our method builds on the intuitive sketching system introduced by [TCH04], who deduces a 3D shape for a virtual garment from contour lines drawn by the user in 2D. As in [TCH04], the user draws over a front (or back) view of a virtual mannequin to create the front (or back) side of a garment. The lines are classified as boundary lines if they cross the mannequin's body or silhouette lines if they do not cross it.

The 3D position of the lines is inferred as follows. Silhouette lines are set to lie in the medial plane \mathcal{P} that faces the user and cuts the mannequin between its front and back parts. The height of the boundary lines above this plane is computed following the assumption that their distance to the body interpolates the distances to the body at their endpoints, computed in the plane \mathcal{P} . A 3D surface for the garment is calculated by extending the height field above the plane \mathcal{P} to the full region delimited by the silhouette and border lines in the 2D drawing, using a simple propagation algorithm based on a smoothing mask. All the computations involving distances to the mannequin surface are accelerated thanks to a precomputed distance field around the body, stored in a 3D grid aligned with the axes.

We enhanced this method by enabling the user to draw seam-lines and darts within the sketching system (Figure 1). In our terminology, the seam-lines are the boundaries between different fabric panels, while darts correspond to cuts within a given panel. Darts are commonly used in clothing design since they create locally non-developable regions to help a garment fit the body shape. We infer the 3D position of the seam-lines and darts by using the previously computed height field above the plane \mathcal{P} .

Finally, the sketching process outputs a 3D surface mesh, which we cut along the seam and dart lines. Computed as a distance-varying offset of the mannequin, this surface is obviously quite far from being piecewise developable. The next section describes our method for turning it into a much better model for the virtual garment.

4. Generation of a piecewise developable approximation

We now describe our algorithm for modifying the sketched input surface to make it piecewise developable. We assume that the user has specified all the desired seams on the surface. Our goal is thus to take each surface panel bounded by the seams and make it developable, with as little modification as possible. To quantify the improvement in developability we measure the distortion introduced by unfolding the surfaces into the plane. The distortion is a natural metric as our final goal is to generate planar patterns for the surface panels. In the ideal case we would like to obtain zero-distortion flattening. However since real fabric can stretch a bit, small distortion is often acceptable. The metrics we use and the unfolding algorithm are discussed in Section 4.4.

To make the panels developable, we use an approach inspired by the moving-least-squares (MLS) approximation [Lev98]. For each triangle on the surface we find the locally best-fitting developable surface and move the triangle to that surface. Similar to MLS we found that local approximation provides a global solution, making the modified surface more developable. We perform the local approximation using the projective geometry setting.

As mentioned, the normal map of any sufficiently smooth developable surface is a union of curves. We observe that therefore the normal map of any sufficiently small region on such surface can be accurately approximated by an arc or a point. The family of surfaces whose normal map is an arc or a point are known as *developable surfaces of constant slope* [PW01]. Following the observation above, any sufficiently smooth developable surface can be approximated locally by a developable surface of constant slope.

Our algorithm uses the following two-step procedure to perform the local approximation.

- For each triangle on the input mesh it computes a local neighborhood and finds a developable surface of constant slope that best approximates this neighborhood (Section 4.1). It then computes the transformation which moves the triangle to the approximating surface with minimal change of its vertex positions (Section 4.2). This computation is performed independently for each triangle. As shown in Figures 3b and 3c the footprint of the normal map on the sphere shrinks, making it nearly one-dimensional. Note that after the transformation the triangles are no longer connected (Figure 3d).
- The second stage of the algorithm (Section 4.3) glues the triangles together, while trying to preserve the newly computed normals and positions. The gluing is applied simultaneously to all the surface panels as they must remain connected.

The process can be repeated to further improve developability (Figures 3e and 3f). We note that as expected the improvement comes at the expense of higher deviation from the original surface, thus it is up to user to decide on the right tradeoff. Finally, we perform planar parameterization

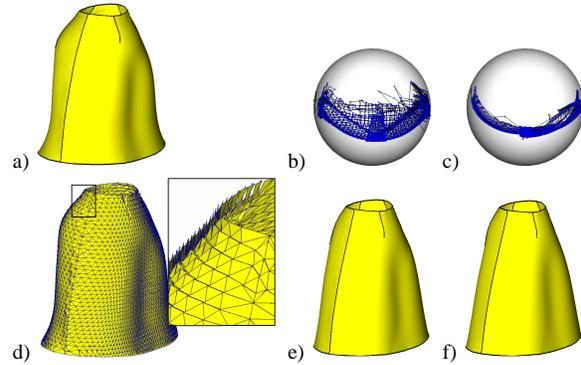


Figure 3: Developable approximation stages: (a) input; (b) normal map of the front panel; (c) normal map after transformation; (d) mesh triangles after transformation; (e) glued mesh after one iteration; (f) mesh after three iterations. Between one and three iterations the distortion decreases from $L_2^{stretch} = 1.00053$, $L_2^{shear} = 5.57e - 6$ to $L_2^{stretch} = 1.00035$, $L_2^{shear} = 2.99e - 6$.

to obtain the actual 2D patterns for the surface panels (Section 4.4). One iteration of approximation on the 7K faces skirt (Figure 3e) takes 29 seconds on a 3GHz Pentium IV.

We now describe the components of our approximation algorithm in more details.

4.1. Local Approximation

As mentioned above, we locally approximate the neighborhood around each mesh triangle by a developable surface of constant slope. The neighborhood of a triangle consists of several rings of triangles, surrounding it. We do the approximation in terms of the normal map, as it controls the developability of our surface and then translate the results into Euclidean space when performing the actual triangle transformation. As noted in [JKS05] a developable surface of constant slope can be described by an *axis vector* N and an angle θ , such that at each point on the surface the angle between the normal to the surface n at that point and the axis is equal to θ . We compute the axis and the angle by solving the following constrained optimization problem,

$$\min_{N, \theta} \sum_j ((n_j \cdot N - \cos \theta) / 2)^2 \text{ subj. to } \|N\| = 1, \quad (1)$$

where n_j are the normals to the faces in the neighborhood. In contrast to other surfaces of constant slope, planes have an infinite number of axis and angle pairs that describe them. Thus we explicitly test if the local neighborhood can be approximated by a plane by finding the axis N that approximates the neighborhood best for $\theta = 0$.

The formulation in Eq. 1 supports surface approximation with cones and does not distinguish between cones whose apexes lie on the surface and those whose apexes lie outside the approximated surface. The apex of a cone is clearly not developable. Thus, to make a surface containing an apex developable we will need to add a dart to the apex. This case

can be detected during parameterization and the darts can be added as described in [JKS05]. Since in our setting extra darts are undesirable, we would like to avoid creating cones whose apex lies on the surface when possible, namely when those regions are sufficiently flat to support a different local approximation. To achieve this we introduce an inequality constraint into the minimization, requiring that $|\cos\theta| < 0.5$.

4.2. Triangle Transformation

We first recompute the normal n of the triangle to move it to the arc on the unit sphere defined by N and θ . If $\theta > 0$ we rotate n around the vector $n \times N$ such that, after rotation, we have $n \cdot N - \cos\theta = 0$, otherwise we set $n = N$. We then compute the new positions for the triangle's vertices v_1, v_2, v_3 , which are as close as possible to the original v'_1, v'_2, v'_3 , such that the new triangle normal is equal to n . The new positions are computed as the solution of a simple quadratic minimization problem,

$$\min_{v_i, d} \sum_{i=1}^3 (v_i - v'_i)^2 \text{ subj. to } v_i \cdot n - d = 0, \quad (2)$$

where d is the scalar offset of the triangle's plane. It is often useful to keep some of the vertices of the input garment mesh in place in order, for instance, to keep the waist-line of a skirt unchanged. We support this requirement by removing the specified vertices from the minimization formula. The system will have a solution as long as one vertex in a triangle remains free.

4.3. Gluing

After all the triangles are moved to their respective local approximating surfaces, the triangles are no longer connected (Figure 3d) and need to be glued back together. We glue the triangles by applying an appropriate linear transformation to each triangle. The transformations are formulated using the local frames of the triangles. We define each local frame using the three triangle vertex positions v_1, v_2 and v_3 , and a fourth point v_4 found by offsetting one of the vertices by the triangle normal [SP04]. The local coordinate frame V is then defined as $V = (v_4 - v_1, v_4 - v_2, v_4 - v_3)$. The transformation gradient expressed in terms of the local frames before and after the transformation (V and \tilde{V}) is $\tilde{V}V^{-1}$ [SP04]. To preserve the normal and remain close to the original surface, the transformation gradient should be close to identity,

$$\min_{\tilde{V}} \sum_j \|\tilde{V}_j V_j^{-1} - I\|_F^2, \quad (3)$$

where I is a 3×3 identity matrix, and the index j goes over the mesh triangles. This linear system is solved simultaneously for the entire garment mesh, since we not only want the triangles within each panel to be glued together, but we also want the panels to remain connected across seams. A skirt after one iteration of developable approximation is shown in Figure 3e. As evident from the metrics (Table 1) the process makes the surface significantly more developable. The resulting surface can be seen as an idealized version of the

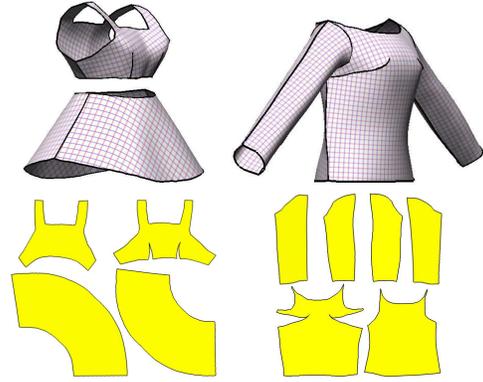


Figure 4: Texture mapped garments and corresponding patterns.

garment, corresponding to the way it would look without the action of gravity and collisions with the body. The next stage of the method makes the garment appear more natural by introducing folds caused by such interaction.

4.4. Unfolding

To obtain planar patterns, the algorithm unfolds the surface panels (Figure 4). For garment design, since fabric can stretch, the input to the unfolding algorithm does not have to be completely developable. We note that shearing during unfolding is highly undesirable since when sewing real garments shearing leads to visible and undesirable wrinkles. We therefore use the ABF++ [SLMB05] conformal (shear minimizing) parameterization to unfold the panels. ABF++ generates zero-distortion parameterization for truly developable surfaces, and minimizes shear when distortion is inevitable.

4.5. Developable approximation outputs

Example of developable outputs produced by our method are shown in Figures 1center, 2c, 3f, and 4. As discussed, we measure the degree of developability of each surface by measuring the distortion created by the unfolding (Table 1). To evaluate the parameterization distortion we measured the $L_2^{stretch}$ [SSGH01] and the L_2^{shear} [SLMB05] of the parameterization. Note that the minimum for $L_2^{stretch}$ is one and for L_2^{shear} is zero. Table 1 lists the statistics for the models shown in the paper. To test the method on an example with higher initial distortion we ran it on the skirt (Figure 3) model without any darts. As Table 1 shows, the distortion before the approximation is quite high, but is drastically reduced after applying our approximation algorithm.

Figure 5 provides a comparison of our developable outputs with those provided by the authors of [WT04], who address the same problem as we do. Since their local optimization method adjusts vertex positions without constraining the surface normals, it tends to result in a wrinkled surface. This effect is undesirable since well fitting garments should ideally have no wrinkles beyond those caused by external physical forces.

	skirt	shirt	bra	mini skirt	pant leg*	skirt no darts
#faces	6818	7016	1366	2923	936	6818
#panels	2	6	2	2	1	2
$L_2^{stretch}$ before	1.003	1.005	1.0015	1.0012	1.008	1.01
L_2^{shear} before	4.6e-5	3e-4	1.5e-4	3.5e-5	8.4e-5	1.4e-4
$L_2^{stretch}$ after	1.00035	1.0014	1.0005	1.0002	1.0001	1.0005
L_2^{shear} after	2.99e-6	6.21e-5	3.2e-5	9.2e-7	2.4e-6	2.9e-6

Table 1: Developable approximation statistics. *For comparison, the distortion for the output generated by Wang [WT04] is $L_2^{stretch} = 1.00015$, $L_2^{shear} = 3.4e-6$.

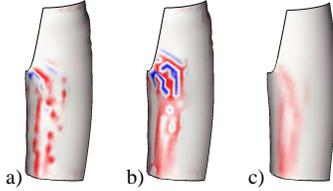


Figure 5: Comparison with Wang et al. [WT04]: (a) input mesh; (b) result provided by Wang; (c) our output. The coloring shows the mean curvature of the surface. The statistics for the models are listed in Table 1.

5. Modeling fabric folds

A comparison of a developable garment output from the previous section with a real, worn garment would not be relevant at this stage as most worn garments exhibit folds due to the pose of the model and gravity. The aim of this section is to add folds to the garment.

Our approach for the efficient, yet accurate, modeling of folds on clothing relies on the geometric buckling properties of fabric, a problem well studied in the physics literature [GS66, AH89, KJL04, AP04]. Garments such as sweaters, trousers or skirts are made of fabric panels wrapped around roughly cylindrical parts of the human body. The specific buckling properties of cylindrical shells [HLP03] should thus be considered. More precisely, in addition to the vertical axis aligned folds that appear when fabric hangs under gravity, characteristic diamond and twist fold patterns also appear on a fabric cylinder under axial compression or twist, respectively. This section studies and models these folds. Rather than using a physically-based simulation, which would be costly and would just recompute well known static shapes, we propose a procedural model for the folds, called the *buckling mesh*. Note that the following is not limited to non-stretchable materials because stretchable cloth would also buckle under axial compression and twist.

5.1. Axial compression: Diamond buckling

The compression of a cylinder of cloth while maintaining a zero Gaussian curvature is made possible by a specific buckling behavior, which we call *diamond buckling* (Figure 6). The fold patterns that appear in this situation have a characteristic diamond shape. Let us characterize this feature geo-



Figure 6: Buckling phenomena for a cylinder of fabric. Diamond shape patterns (left and middle) appear under axial compression. Twist patterns (right) made of parallel oblique wrinkles appear when one end of the panel is twisted.

metrically for a cylinder of radius R . First, since the diamond pattern allows axial compression of the fabric, each diamond can be seen as two triangles folded on each other while the cylinder is compressed [JSAH77]. When it is fully compressed, the triangles are superposed (Figure 7, right). This implies that the diamond's width a and half-height b are linked by the following geometric constraint,

$$R^2 = (a/2)^2 + (R-b)^2. \quad (4)$$

Second, our observations of real fabric show that, independent of the fabric thickness and the cylinder's radius, the number of diamond patterns that appear around the circumference of the cylinder is always an integer: we let n denote this value. The geometric relationship between a , R and n is

$$a/2 = R \sin(\pi/n). \quad (5)$$

In our experiments with real fabric, n varies from 2 to 8 for a cylinder of a few centimeters of radius and different thicknesses of cloth; the thicker the cloth, the smaller n .

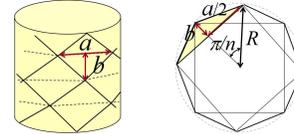


Figure 7: Estimation of the parameters of the diamond buckling patterns for a cylindrical fabric swatch. *Left:* Front view of the patterns; *Right:* Top view when the cylinder is totally compressed.

Last, the half-height b of the diamond is related to both the radius R and the physical properties of the fabric [HLP03]. This relationship can be written in the form

$$b = \lambda \sqrt{R}, \quad (6)$$

where λ is a parameter which depends on the physical properties of the fabric only (not its geometry). We obtain λ by performing simple measurements on a real fabric sample. These three equations enable us to extract all the parameters necessary to characterize the diamond buckling geometry for this fabric. In detail the process is:

1. Place a real cylindrical fabric swatch of known radius R_0 around a slightly smaller rigid cylinder. Observe the diamond buckling mode under axial compression and record the associated number of diamonds n_0 around the circumference, for this specific radius.
2. Estimate λ using the following relation:

$$\lambda = \sqrt{R_0} \left(1 - \cos \frac{\pi}{n_0}\right), \quad (7)$$

obtained by substituting b with a (using Equations 4 and 6) and then replacing a by its value from Equation 5.

3. For any radius R , compute the geometric parameter n of the diamond pattern by re-writing Equation 7:

$$n = \lfloor \frac{\pi}{\arccos(1 - \lambda/\sqrt{R})} \rfloor, \quad (8)$$

and use Equations 5 and 6 to obtain a and b .

These equations for n , a and b will be used to generate a buckling mesh for a more general case than the ideal cylinder, as explained in Section 6.

5.2. Twisting and axis aligned folds

The diamond buckling patterns we described usually appear when an elbow or a knee of a character is bent, or when the sleeves of a sweater are pulled up. Other typical deformations are the parallel folds generated when the body twists or when a loose skirt hangs under gravity. We call these phenomena *twist buckling* and *axis aligned folds*, respectively.

If we study twisting on a cylindrical fabric swatch (Figure 6, right), we observe that a twist motion at one end of the cylinder produces parallel fold patterns along an oblique direction. Moreover, if the cylinder has already been compressed along its axis, this twist mode takes place along the diagonal contour lines of the diamond pattern. The diamond patterns of the last section thus give us the direction and frequency of the twist buckling folds. When twisted, the mean radius of the cylindrical fabric swatch decreases, and the height of the oblique folds is such that the length of an imaginary thread on the fabric remains constant while twisting.

The axis aligned folds are those generated by compression in a perpendicular direction to the cylinder axis. These folds, captured by many previous geometric models [HBVMT99, LC04], are similar to the sinusoidal folds observed on a hanging piece of fabric, such as a curtain.

5.3. Procedural cloth model: The buckling mesh

We now propose a procedural method for reproducing these folds on an ideal fabric cylinder. The extension to an arbitrary garment will be detailed in Section 6. Our model is based on a mesh aligned with the main fold directions, called the *buckling mesh*.

Since a fabric swatch can be compressed and twisted simultaneously, and the amount of the different buckling modes varies from place to place, we need a model that blends the different deformations locally. To this end, we define the buckling mesh in the 2D space of the unfolded fabric as a grid of buckling primitives, i.e. rectangular patches controlled by nine control points, depicted as black and white small squares and numbered from 1 to 9 in Figure 8a. Each patch covers four adjacent half diamond shapes and is centered on the point at which they meet.

This coarse grid, called a *control mesh*, is later refined

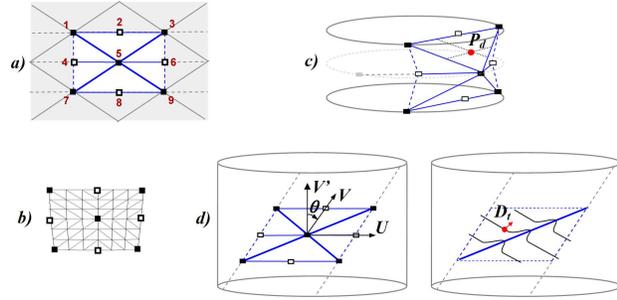


Figure 8: (a) Buckling patch made of 9 control points that guide the procedural deformation. (b) Refined mesh interpolating the control points. (c) Transformed triangles of a diamond buckling pattern. (d) Fold across the selected diagonal of a twist buckling pattern.

into a higher resolution mesh, called the buckling mesh (Figure 8b) whose edges are aligned with the possible folds, enabling us to model a smoothly folding cloth surface convincingly with relatively few facets. Using the vertex correspondence between the unfolded 2D pattern and the fabric panel in 3D, a 3D version of the control mesh is generated. The 3D version of the control mesh serves for controlling the procedural generation of folds. The degrees of diamond, twist, and axis aligned buckling are computed as parameters based on the difference between the current lengths of edges between neighboring control points and the corresponding lengths on the patterns. These parameters are used to calculate the positions of the vertices within each patch of the buckling mesh, such that the surface mimics the different buckling phenomena. These positions are computed as follows:

1. Cardinal spline interpolation is used to create a smooth patch from the positions of the nine control points. To avoid the creation of excessively flat or curved regions when the control points move, we set the tension parameter τ to a linear function of the distance between control points, chosen so as to reduce the variation of length of the associated splines when the control points move. Let P_s denote the position we get on this smooth surface for a given buckling patch point P with parameters (u, v) .
2. We compute the position P_d the point would have if diamond buckling was the only activated mode. The diamond buckling pattern consists of six triangular parts (Figure 8c). During compression, these triangles move according to the points of the control mesh. To ensure correct folding of the diamonds, only five of the control points are used (the black dots 1, 3, 5, 7, 9 in Figure 8a) while the remaining four points (the white dots 2, 4, 6, 8 in Figure 8a) are recomputed from them. We use (u, v) to locate the triangle that P belongs to and its relative position inside it. From this we obtain its position P_d on the transformed triangles. Let $D_d = \alpha(P_d - P_s)$ be the displacement that moves P_s onto the diamond buckling pattern, where α is a compression coefficient: α varies from zero, if the patch is uncompressed, to one, if it is totally compressed.

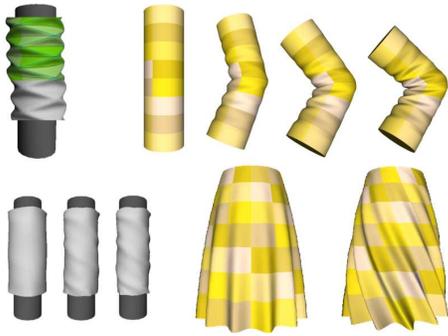


Figure 9: Procedural folds. Left: diamond and twist buckling. Right: complex folds during progressive bending and when axis aligned folds are combined with twisting.

3. Assuming twist buckling only, we compute the displacement D_t that moves the point P_s onto the twist buckling pattern. The direction of the fold produced by twist buckling is one of the two diagonals of the buckling patch. The appropriate diagonal is determined by the sign of the angle θ between the direction V and the direction V' orthogonal to U , where U and V are defined by the control points (Figure 8d, left). A sinusoidal fold across the chosen diagonal is created on the patch (Figure 8d, right) so that the length of a thread remains unchanged compared to its length measured on the unfolded 2D pattern. Thus, $D_t = \beta N_s$, where N_s is the normal to the interpolating smooth patch at point P_s . β is a scalar function of u and v that is equal to the height of the fold on the chosen diagonal and decreases smoothly to zero elsewhere.
4. Axis aligned folds, like the folds in hanging curtains, are modeled directly during the initial calculation of the smooth surface by simply displacing the middle column of three control points (dots 2, 5, 8 in Figure 8a) in each buckling patch along the direction of the normal N_s . The degree of displacement is chosen to ensure that the length of a horizontal thread remains unchanged. Thus the effects of these folds are included implicitly in the term P_s .
5. We combine the buckling modes by defining the final 3D position of the point as: $P' = P_s + D_d + D_t$.
6. Finally we smooth the resulting surface, recomputing the position of the buckling mesh vertices as a linear combination of their neighbors using a standard square filter with coefficients $(\frac{1}{4}, \frac{1}{8}, \frac{1}{16})$. This smooths out the tangent discontinuities introduced while computing P_d from a coarse triangle mesh in Step 2. The folds we obtain after applying these buckling effects are depicted in Figure 9.

6. Generation of the 3D garment

6.1. Fitting a control mesh to the sewing pattern

Let us now consider the 2D sewing patterns output for an arbitrary garment using the method in Section 4. During the initial sketching of the garment the seam lines between panels were chosen so as to divide the body into a set of roughly cylindrical parts (typically the arms, the legs and the torso).

We wish to fit a control mesh to the 2D patterns of each part, as was done for ideal cylinders in Section 5.3. To ease the computations, we first rotate the pattern so that the vertical axis corresponds to the major axis of the associated body part. We then fit smooth curves to the bottom and top contours of the pattern. Intermediate, smooth curves computed by interpolation between the top and bottom curves are used to calculate the control point positions in order to obtain a grid of buckling patches.

We first choose the simulated fabric by selecting a value for the parameter λ from among those measured on real samples, as explained in Section 5.1. We get the number of buckling patterns, n , around the axis for this piece of garment from Equation 8, using an approximate, global value for the radius R of the pseudo-cylinder.

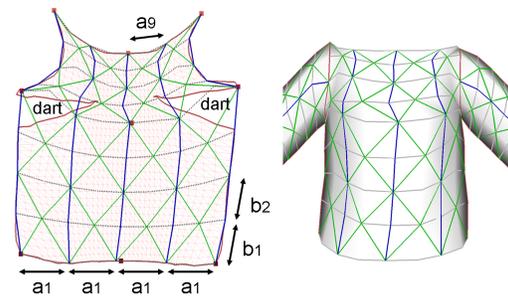


Figure 10: 2D computation of the buckling patches (left). Note that the patches overlap the darts, but since the original lengths of the edges are computed on the 3D developable surface (right) this is not a problem.

We then compute the set of buckling primitives as follows. Starting at the bottom of the 2D pattern, we successively compute the local radius R_i of the pseudo-cylinder along an intermediate curve interpolating the top and bottom curves, using $2\pi R_i = l_i$ (the length of the curve) if there is a single panel for this body part, and $\pi R_i = l_i$ if there are a front and a back panel. We set $a_i = l_i/n$ and compute b_i from a_i and R_i using Equation 4. This gives us the position of the next interpolating curve at the top of the next row of buckling primitives. We continue this way until we reach the top of the sewing pattern. If needed, we scale our control mesh in order to get an integer number of rows of patches covering the pattern (Figure 10, left). Lastly, the mesh points at the edges of the buckling grid are snapped to the exact edges of the corresponding sewing pattern, so that there is no loss of precision at the border of the garment and along the seam lines. Note that using this method, the relative height of the buckling patches adequately varies depending on the local variation of the pseudo cylinder radius (Figure 10, right). To have a good fit of buckling primitives across seam lines, we use the same series of b_i values for the front and back panels of a piece of garment. In addition, we adjust the number n of buckling primitives on the sleeves so that they fit with the buckling primitives on the body along the seams.

This control mesh is converted into 3D, using the mapping between the pattern and the developable surface panel. Finally, the original lengths of the edges required by our procedural folding method (Section 5.3) are measured on the 3D developable surface rather than on the 2D pattern. This has the advantage of handling darts with no extra processing (Figure 10), since two control points separated by a dart will simply be set to be at a smaller distance than on the unfolded 2D patterns. The resulting garment surface is now ready for the procedural generation of folds.

6.2. Final 3D garment

Once we have the 3D *control mesh* of buckling patches for the whole garment, we procedurally generate the refined buckling mesh, as detailed in Section 5, to get the final geometry. The control points can then be used as handles to place the garment in a more general position. Note that any reasonable positioning of the control points will generate a visually convincing cloth surface, since the buckling mesh accurately models the way fabric folds.

We procedurally alter the positions of the control points in order to address phenomena such as collisions with the body and the action of gravity, producing more natural illustrations of the virtual garment.

- **Handling collisions:** The first reason for moving the control points is to remove any local penetration inside the mannequin. The developable approximation of the initial surface has altered the surface shape, so that it may have caused such penetrations. We use the precomputed distance field around the mannequin as an efficient method to find penetrations and to move the points outside, along a direction defined by the gradient of the distance field. We check and move both the control points and the vertices output by the refined buckling mesh.
- **Mimicking the action of external forces:** In general cases, a physical simulation should be processed on control points to handle the action of external forces. But in more particular cases like the action of gravity on a loose skirt, it can be mimicked by creating compression perpendicular to the major axis. Twist can be used to model a sideways wind effect. Both phenomena were combined at the bottom right of Figure 9.
- **Generating a more natural configuration:** More natural configurations of the garments can be obtained by bending the limbs of the mannequin or twisting the body. This is done by moving the positions of the control points of the buckling mesh according to the pose of the mannequin's skeleton, using a standard skinning technique. See Figure 11. Nice folds are also obtained by pulling the sleeves up using a procedural axial displacement of the sleeves' control points, as shown in Figure 1.

Lastly, we validate our method by comparing the virtual garments with real clothing sewn using the patterns we output, shown in Figures 1 and 12.

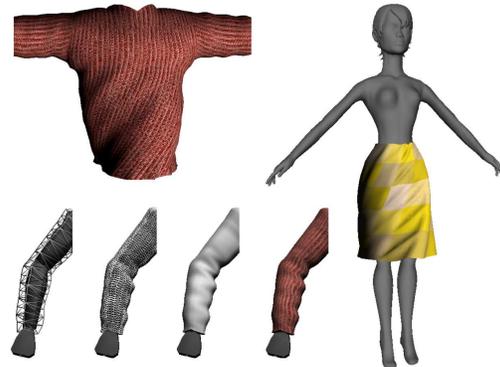


Figure 11: Garment deformation under different body poses.



Figure 12: Comparison between the virtual clothing we designed and real replicas sewn from the patterns we output.

7. Conclusion

We have proposed a novel solution for the design of virtual garments. An initial rough geometry is approximated by a piecewise developable surface. The folds that appear on a garment worn by a character are added procedurally, taking advantage of the well studied buckling properties of fabric. Since we compute the corresponding 2D sewing patterns, distortion-free texture mapping becomes straight-forward, and real replicas of the designed garments can be sewn.

Our methods for developable approximation and for generating cloth folds can be used in several other contexts. In particular, the developable approximation method can be used for a variety of engineering applications requiring developable surfaces, such as forming or forging [Pet04]. Our procedural method for fold generation can be applied to generate folds on any input mesh representing a garment.

Limitations and future work: Dynamic animation of garments was outside the scope of this paper. However, our

method could be used in two different ways toward this goal. The 2D patterns we generate can be used by a cloth animation system to compute the rest lengths of the springs that model the cloth material. Thus, the garments we design can be animated using standard techniques. Another option would be to take advantage of our procedural modeling of fabric folds during animation. Then, only the control mesh would need to be animated using physically-based simulation, while fine details such as folds, costly to simulate since they are caused by stiff, buckling phenomena [CK02], would be added procedurally at no cost prior to rendering. We plan to explore this approach in the near future.

Another possible application of our work is the prototyping of real garments. A fashion designer could use our sketching system to quickly sketch some clothing, automatically get different 3D views of the garment, edit the model as necessary, and finally print the corresponding 2D patterns. This would require a number of enhancements to our method, such as introducing gussets and enabling axial symmetry specification.

Acknowledgments

We would like to thank C.L. Leavitt for assistance in editing, and C.C.L. Wang for the comparison results. This project was partially funded by NSERC. We also thank B. Thomaszewski for having previously worked with us on an alternative approach to parts of this work, and E. Turquin for providing us the sketching software and answering our questions. Jamie Wither is supported by a grant from the European Community under the Marie-Curie project VISITOR.

References

- [AH86] AMIRBAYAT J., HEARLE J. W. S.: The complex buckling of flexible sheet materials, part I: Theoretical approach. *Int. Journal of Mechanical Science* 28, 6 (1986), 339–358.
- [AH89] AMIRBAYAT J., HEARLE J. W. S.: The anatomy of buckling of textile fabrics : Drape and conformability. *Journal of Textile Institute* 80, 1 (1989), 51–70.
- [AP04] ATHIANNAN K., PALANINATHAN R.: Experimental investigations on buckling of cylindrical shells under axial compression and transverse shear. *Sadhana* (Feb. 2004), 93–115.
- [BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. *ACM-EG Symposium on Computer Animation* (July 2003), 28–36.
- [BWK03] BARAFF D., WITKIN A., KASS M.: Untangling cloth. *ACM Transactions on Graphics* 22, 3 (2003), 862–870.
- [CK02] CHOI K.-J., KO H.-S.: Stable but responsive cloth. *ACM Transactions on Graphics* 21, 3 (July 2002), 604–611.
- [CLL*99] CHEN H.-Y., LEE I.-K., LEOPOLDSEDER S., POTTMANN H., RANDRUP T., WALLNER J.: On surface approximation using developable surfaces. *Graphical Models and Image Processing*, 61, 2 (1999), 110–124.
- [Den76] DENBY E.: The deformation of fabrics during wrinkling, a theoretical approach. *Textile Research J.* 46 (1976), 667–670.
- [GS66] GROSBERG P., SWANI N. M.: The mechanical properties of woven fabrics, part III: The buckling of woven fabrics. *Textile Research Journal* 36 (1966), 332–338.
- [HBVMT99] HADAP S., BANGARTER R., VOLINO P., MAGNENAT-THALMANN N.: Animating wrinkles on clothes. *Proc. IEEE Visualization '99* (Oct. 1999), 175–182.
- [HLP03] HUNT G., LORD G., PELETIER M.: Cylindrical shell buckling: a characterization of localization and periodicity. *Discrete and continuous dynamical systems* (Nov. 2003), 505–518.
- [JKS05] JULIUS D., KRAEVOY V., SHEFFER A.: D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum, Proceedings of Eurographics 2005* 24, 3 (2005), 581–590.
- [JSAH77] JOHNSON W., SODEN P. D., AL-HASSANI S. T. S.: Inextensional collapse of thin-walled tubes under axial compression. *Journal of Strain Analysis* 12, 4 (1977), 317–330.
- [KG90] KUNII T. L., GOTIBA H.: Modeling and animation of garment wrinkle formation processes. *Proc. Computer Animation '90* (1990), 131–147.
- [KJL04] KANG T. J., JOO K. H., LEE K. W.: Analysis of fabric buckling based on nonlinear bending properties. *Textile Research Journal* (2004).
- [KWH04] KIMMERLE S., WACKER M., HOLZER C.: Multilayered wrinkle textures from strain. *Proceedings VMV* (2004).
- [LC04] LARBOULETTE C., CANI M.-P.: Real-time dynamic wrinkles. *Proc. Computer Graphics International* (2004).
- [Lev98] LEVIN D.: The approximation power of moving least-squares. *Math. Comput.* 67, 224 (1998), 1517–1531.
- [May04] Maya cloth. In *Maya cloth user manual, Alias* (2004).
- [Pet04] PETERNELL M.: Developable surface fitting to point clouds. *Computer Aided Geometric Design* 10 (2004), 785–803.
- [PW99] POTTMANN H., WALLNER J.: Approximation algorithms for developable surfaces. *Computer Aided Geometric Design* 16, 6 (1999), 539–556.
- [PW01] POTTMANN H., WALLNER J.: *Computational Line Geometry*. Springer Verlag, 2001.
- [SLMB05] SHEFFER A., LÉVY B., MOGILNITSKY M., BOGOMYAKOV A.: ABF++: fast and robust angle based flattening. *ACM Trans. Graph.* 24, 2 (2005), 311–330.
- [SP04] SUMNER R. W., POPOVIC J.: Deformation transfer for triangle meshes. *ACM Trans. Graph.* 23, 3 (2004), 399–405.
- [SSGH01] SANDER P. V., SNYDER J., GORTLER S. J., HOPPE H.: Texture mapping progressive meshes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 409–416.
- [TCH04] TURQUIN E., CANI M.-P., HUGHES J. F.: Sketching garments for virtual characters. *EG Workshop on Sketch-Based Interfaces and Modeling* (Aug. 2004).
- [Tso91] TSOPELAS N.: Animating the crumpling behaviour of garments. *EG W. on Animation and Simulation* (1991), 11–23.
- [WT04] WANG C. C. L., TANG K.: Achieving developability of a polygonal surface by minimum deformation: a study of global and local optimization approaches. *The Visual Computer* 20, 8-9 (2004), 521–539.
- [WWY04] WANG C. C. L., WANG Y., YUEN M. M.: On increasing the developability of a trimmed nurbs surface. *Eng. Comput. (Lond.)* 20, 1 (2004), 54–64.