Virtual garments based on geometric features of fabric buckling

Philippe Decaudin, Bernhard Thomaszewski, Marie-Paule Cani

▶ **To cite this version:**

R INRIA

Virtual garments based on geometric features of fabric buckling

Philippe Decaudin — Bernhard Thomaszewski — Marie-Paule Cani

N° 5549

R apport de recherche

INRIA
RHÔNE-ALPES

Virtual garments based on geometric features of fabric buckling

Philippe Decaudin[*] , Bernhard Thomaszewski[†] , Marie-Paule Cani[‡]

Abstract:

We present a new, procedural approach for modelling and animating the garments worn by virtual characters. Our method takes into account the properties of fabric to generate the geometry of local folds that are so crucial for visual realism.

Modelling is performed by approximating an initial free-form surface roughly representing a garment by a piecewise developable mesh called the *buckling mesh*. The latter models the specific buckling features of fabric: more precisely, it automatically folds according to pre-computed patterns, depending on the 3D positions of a few control points. The approximation enhances the realism of the garment surface since the wrinkles that make it piecewise developable also give it the appearance of fabric. Moreover, the set of 2D fabric panels required to sew the garment is computed during the process, enabling distortion-free texture mapping.

During animation, the buckling mesh procedurally folds and unfolds according to the positions of its control points. The latter are animated through a simple physically-based model that accounts for the floating behaviour of cloth. Our solution differs completely from the standard physically-based approach, since it does not require the use of many mass nodes nor of stiff equations to generate non-extensible cloth.

As our results show, our method can be used for the real-time, visually aesthetic animation of dressed characters, making it applicable to enhance the realism of virtual humans in real-time applications.

Key-words: cloth modelling, procedural models, real-time animation

[*] GRAVIR/IMAG-INRIA, projet EVASION – 655 avenue de l'Europe – 38334 Montbonnot Saint-Ismier – France
Email: philippe.decaudin@imag.fr – Web: http://www.antisphere.com

[†] University of Tübingen, WSI/GRIS – Sand 14 – 72076 Tübingen – Germany
Email: b.thomaszewski@gmx.de

[‡] GRAVIR/IMAG-INRIA, projet EVASION – 655 avenue de l'Europe – 38334 Montbonnot Saint-Ismier – France
Email: marie-paule.cani@imag.fr – Web: http://www-evasion.imag.fr/Membres/Marie-Paule.Cani/

Vêtements virtuels basés sur les propriétés de flambage du tissu

Résumé :

Nous présentons une nouvelle approche procédurale pour modéliser et animer des vêtements portés par des acteurs virtuels. Notre méthode se base sur les propriétés des tissus pour générer la géométrie des plis qui apparaissent localement sur les vêtements. Ces plis sont d'une grande importance pour le réalisme visuel.

La modélisation est obtenue en approximant une surface initiale arbitraire représentant approximativement le vêtement par un maillage développable par morceaux appelé le *maillage de flambage*. Ce dernier modélise les propriétés de flambage spécifiques au tissu : plus précisément, il se plie automatiquement en fonction de motifs pré-définis, qui dépendent de la position d'un petit nombre de points de contrôles. Ceci augmente le réalisme du vêtement car ces plis qui rendent sa surface développable par morceaux lui confèrent aussi l'apparence d'un tissu. De plus, les patrons 2D de tissu nécessaires pour coudre le vêtement sont calculés pendant le processus, permettant ainsi de le texturer sans distorsion.

Au cours de l'animation, le maillage de flambage se plie et se déplie procéduralement en fonction des positions de ses points de contrôle. Ceux-ci sont animés par un modèle physique simple qui simule le comportement des parties flottantes du vêtement. Notre solution diffère totalement des approches physiques standards, car elle ne nécessite pas l'utilisation de beaucoup de masses ni de ressorts de fortes raideurs pour obtenir un vêtement non-extensible.

Comme le montre nos résultats, notre méthode peut être utilisée pour l'animation en temps-réel de personnages habillés, la rendant utile pour améliorer le réalisme d'humains virtuels dans les applications temps-réelles.

Mots-clés : modélisation de vêtements, modèles procéduraux, animation en temps-réel

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 3D applications. In addition to realistic bodies, faces and hair, providing a way to animate fine details such as dynamic wrinkles on garments is crucial for realism.

The computer graphics literature classifies the garments worn by virtual characters into three categories [MTCK$^+$04]: *stretch cloth* such as underwear, for which animation through standard skinning techniques is sufficient; *floating cloth* such as large skirts, for which using a full physically-based simulation is mandatory; and intermediate, *loose cloth*, such as trousers, shirts, sweaters and jackets. The latter dynamically fold and unfold during motion, but without showing a very strong dynamic behaviour. As the authors note, these intermediate garments have either been animated through skinning or through physically-based modelling, depending on the application. However, none of these solutions are satisfactory in terms of both efficiency and visual realism. Our work specifically addresses the modelling of these loose, but not floating garments,which undoubtfully constitute the most generally used type of clothes. When worn, these garments fold according to specific patterns that mostly depend on the body posture, as those depicted in Figure 10. We believe that their limited dynamic behaviour does not justify the use of a costly, full physically-based simulation.

Creating virtual garments is usually a quite tedious task: using standard software such as MayaCloth [May04], computer artists first need to design the set of 2D fabric panels[1] necessary to create the garment – as if they were to sew it in the real world. Once these panels are placed around the character, and sewing constraints specified, they need to run a simulation to obtain the garment's rest shape. During both this first simulation step and during subsequent animation, generating realistic cloth that forms nice folds as in Figure 10 requires the use of a fine mesh. This involves stiff equations, due to the quasi inextensibility and buckling properties of fabric. This in turn leads to a fair amount of time spent in collision detection and response. An alternative, of general use in real-time applications such as video-games, consists in directly modelling the 3D geometry of a garment (around or even instead of the character's body) and animating it through simple methods such as skinning. This process is much easier for the artist, but the result then lacks realism: even if some geometric wrinkles were modelled, they will remain static during motion, unless specific deformers or blend-shapes are set up. Moreover, there is little chance that these user-defined folds meet the geometric constraint of making the garment piecewise developable onto a plane, nor that they fit the buckling patterns of fabric.

Now, with a rough 3D shape as an input, can we turn this arbitrary geometry into a convincing garment? To do so, we need to automatically make it piecewise developable, while adding folds and wrinkles that make the surface look like cloth. Can we compute the set of 2D fabric panels needed to sew this garment? Even if the latter is not to be sewed in the real world, getting the shape of these panels would be of great help for distortion-free texture mapping. Can we provide a real-time method for animating the garment, while combining both dynamic effects and convincing local details? This paper solves these three problems in the specific, but useful case of loose clothes such as trousers, shirts and sweaters. The key observation is that these loose garments are made of pieces of fabric wrapped around roughly cylindrical parts of the human body. This leads us to exploit some a priori knowledge on the way fabric folds when wrapped around a cylinder. The resulting, very specific buckling patterns are modelled and animated procedurally, enabling us to account for the dynamic effects at a much coarser scale. This approach yields a stable, efficient, yet realistic way to model and animate common types of garments.

1.1 Related work

Geometric modelling of garments:
As already emphasized, the standard way of designing realistic virtual garments [May04, SyF04] is very close to the technique one would use to sew them in the real world: it starts with the definition of all the necessary 2D fabric panels, thus requiring specific know how, not necessarily acquired by all computer artists. The tedious process of specifying these panels and their sewing constraints is made more complex by the need of setting up adequate values for a set of physical parameters and to run a simulation, even when only a static garment shape is needed. The alternative is to use a solution that directly outputs a 3D shape for the garment such as interactive design or automatic generation from sketch [TCH04]. Then, the generated shape has little chance of being piecewise developable and will not depict the fold patterns that are specific to real garments in any rest position. Our work enhances these alternative approaches by making them usable as a starting point for modelling the static shape of worn garments.

---

[1]As in MayaCloth, we use the term *panels* for the 2D patterns that form a garment, to avoid confusion with the fold patterns corresponding to the different buckling modes of fabric.

Our approach has been inspired by the very original work of Mitani [MS04] on mesh approximation with a developable surface. In this paper, the authors approximate any existing 3D mesh by a developable surface made of triangle strips. The resulting 2D patterns enable one to build paper-craft toys that roughly approximate the initial geometry. These strip-based patterns are indeed very different from the ones a tailor would use to sew a garment.Nevertheless, the idea of loose mesh approximation under the geometric hypothesis of a surface being locally developable will be re-used in our case.

Animation:
Improving the physically-based simulation of cloth has attracted a lot of attention during the past few years. Important progress has been made in the robustness, stability and efficiency of both integration schemes and collision processing algorithms [VMT01, CK02, BFA02, BMF03, BWK03, BA04][2]. When used for modelling loose garments, these physically-based approaches sometimes exhibit specific problems: Choi stressed that cloth models are ill-conditioned in buckling situations, when specific folds should appear (such as when a sleeve bends). He uses some a priori knowledge on the shape of the buckling patterns to solve the numerical problem. Baraff proposes a history-free algorithm for resolving the tangles that frequently occur in cloth self-collisions, in particular when different body parts intersect. Bridson notes in 2003 that collision response with the character's body may incidently remove most of the dynamic folds and wrinkles. A geometric approach is used to solve the problem.

At the other end of the spectrum, geometric methods have been used to generate wrinkles on virtual garments. Some of these techniques have been combined with a physically based simulation applied at a coarser scale. Kunii [KG90] uses a physically-based simulation as a reference to capture the geometric parameters of cloth wrinkles. These parameters are then used to re-create 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 buckling patterns illustrated on a trousers' leg, has been inspiring for us.

[HBVMT99, LC04] take into account constant length constraints along curves to generate sinusoidal wrinkles on clothes, without considering the specific buckling features of cloth wrapped around cylindrical limbs. Kimmerle [KWH04] rather uses a strain measure to generate similar wrinkles.

Our approach is primarily geometric. Like Tsopelas, we rely on the well studied buckling properties of cloth to set up a solution, since we doubt an existing physically-based model can be a reference as good as a real material. Contrary to previous approaches, our method is not limited to either sinusoidal or diamond-shape wrinkle patterns. It rather creates general fold shapes by locally combining the effects of the active buckling modes, which generally include some amount of twist and of diamond buckling.

1.2   Overview

This paper exploits geometric constraints – namely the buckling features of cloth wrapped around body parts – for the real-time, yet visually pleasing animation of the garments worn by virtual characters. The user just needs to input an arbitrary 3D mesh roughly representing the garment's shape and to specify the seams. Our constrained model for cloth meshes, called the *buckling mesh*, is used to approximate the initial geometry, automatically generating the wrinkles that make the garment piecewise developable onto a plane. Since this process computes the 2D fabric panels needed to form the garment, distortion-free texture mapping is made possible at no extra cost. During animation, the buckling mesh automatically folds and unfolds according to the displacement of its control points, tailored by the underlying body motion. Detecting collisions with the body becomes straightforward since the animated mesh is aligned with the main fold directions. Lastly, the model can be coupled with a simple physically-based model to account for dynamic effects at low cost.

Section 2 reviews the buckling properties of fabric and derives a procedural cloth model dedicated to worn garments. Section 3 proposes a method for approximating any given 3D mesh with our cloth model. Section 4 addresses animation, explaining how dynamic effects and collision processing can be combined with the procedural cloth model. We then discuss results and directions for future research.

---

[2]Reviewing these methods, which are crucial for the generation of realistic floating cloth, is beyond the scope of this paper. Good overviews can be found in [HB00, MTCK+04]

2 Cloth Buckling on Cylinders

Our approach for the efficient, yet detailed modelling of cloth is to rely on the specific buckling properties of fabric, well studied in the physics literature [GS66, AH89, KJL04, AP04]. The best known type of folds for cloth are probably the ones that appear when it lies on a plane, namely the sinusoidal, parallel folds modelled by Hadap [HBVMT99] and the more complex "three-folds" (T-shape folds) that, according to Amirbayat [AH89], distinguish cloth from standard thin-shell material. However, when worn, the loose garments we are interested in (such as sweaters or trousers) are made, to a first approximation, of a set of fabric panels wrapped around roughly cylindrical parts of the human body. The specific buckling properties of cylindrical shells [HLP03] should thus be considered, instead of the features exhibited by roughly planar cloth. More precisely, very characteristic diamond and twist fold patterns appear on a cylinder of fabric under axial compression or twist, respectively. This section studies these phenomena and proposes a procedural model in the case of a single, ideal cylinder. This model, called the *buckling mesh*, will be extended in the next section to allow the modelling of arbitrary loose garments.

2.1 Axial compression: diamond buckling

Although sinusoidal, parallel folds have been used for modelling cloth compression along cylindrical limbs, this representation fails to capture the complex fold patterns one can observe with real fabric (see Figure 1). Indeed, wrinkling with sinusoidal folds only clamps the surface's Gaussian curvature[3] to zero (i.e. keeps the surface locally isometric to a plane) when the large-scale shape is planar. If applied to a cylinder, sinusoidal folds perpendicular to the compression axis yield two non-zero principal curvatures, so the resulting surface is no longer developable. In real life, the axial compression of cylindrical panels of paper or cloth is made possible by a very specific buckling behaviour, which we call *diamond buckling*. The fold patterns that appear in this situation have a very characteristic diamond shape.

Figure 1: *Buckling behaviour of real cylindrical panels of fabric. Diamond shape patterns (left and middle) appear under axial compression, since the surface maintains a zero Gaussian curvature everywhere during deformation. Twist patterns (right) made of parallel oblique wrinkles appear when one end of the panel is twisted.*
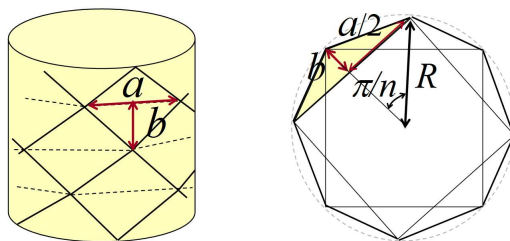

Figure 2: *Estimation of the parameters of the diamond buckling patterns for a cylindrical fabric panel. Left: front view of the patterns; Right: view from the top when the cylinder is totally compressed.*

Let's try to characterize this feature geometrically for a cylindrical panel of radius R: Since the diamond pattern allows the axial compression of the fabric, each diamond can be seen as two triangles folded on each other while the cylinder is compressed [JSAH77], and when it is fully compressed, the triangles are superposed (see Figure 2). This implies that its width a and height b are linked by the following geometric constraint:

$$R^2 = (a/2)^2 + (R-b)^2 \qquad (1)$$

---

[3] Defined as the product of the principal curvatures. See [Kre91]

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

$$a/2 = R\sin(\pi/n) \tag{2}$$

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

Lastly, the height b of the diamond patterns is also related to the radius R and to the physical properties of the fabric [HLP03]. This relationship can be written in the form:

$$b = \lambda\sqrt{R} \tag{3}$$

where λ is a parameter related to the physical properties of fabric.

These three equations enable us to extract, from a very simple measure on a real fabric sample, all the parameters necessary to characterize the diamond buckling geometry for this specific fabric. We proceed as follows:

1. We place a real cylindrical fabric panel of known radius R_0 around a slightly smaller rigid cylinder. We observe the diamond buckling mode under axial compression and get the associated value n_0, for this specific radius.

2. We get λ, which only depends on the fabric, from this measure, by expressing b from a in (3) using (1), and then replacing a by its value from (2). This gives us:

$$\lambda = \sqrt{R_0}(1 - \cos\frac{\pi}{n_0}) \tag{4}$$

3. For any radius R, the geometric parameters of the diamond pattern are computed by re-writing equation (4) to get n:

$$n = \lfloor \frac{\pi}{\arccos(1 - \lambda/\sqrt{R})} \rfloor \tag{5}$$

and then using eq. (2) and (3) to respectively get a and b.

2.2   Twist buckling

During the animation of a character, the diamond buckling patterns we just described typically appear near the elbow and knee when they bend. Another typical motion is a twist along the body, producing oblique parallel folds, as depicted on Figure 10 (right).

If we study this phenomenon on a real cylinder of fabric with radius R (see Figure 1, 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[4]. 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 panel decreases, and the height of the oblique folds is so that the length of an imaginary thread on the fabric remains constant while twisting.

2.3   Buckling mesh

The key feature of our model is to use a procedural, geometric model to reproduce the buckling modes we just characterized, rather than trying to simulate them using physically-based modelling. There are several good reasons to do so; the buckling patterns are quasi-static shapes: they appear and disappear as a series of rest shapes. Buckling is a very intricate phenomenon to simulate, as emphasized by [CK02]. Lastly, a physically-based simulation is still useful to obtain dynamic behaviour of cloth such as floating effects, but our method will alleviate the load of the simulation module, by giving it only large scale control points to animate.

Our approach for modelling buckling of a cylindrical fabric panel is based on a mesh aligned with the main fold directions, called the *buckling mesh*. Since the panel can be compressed and twisted at the same time, the amount of compression and twist varying from place to place, we need a model that provides a local control of the amount of each deformation rather than providing control through global compression and twist parameters.
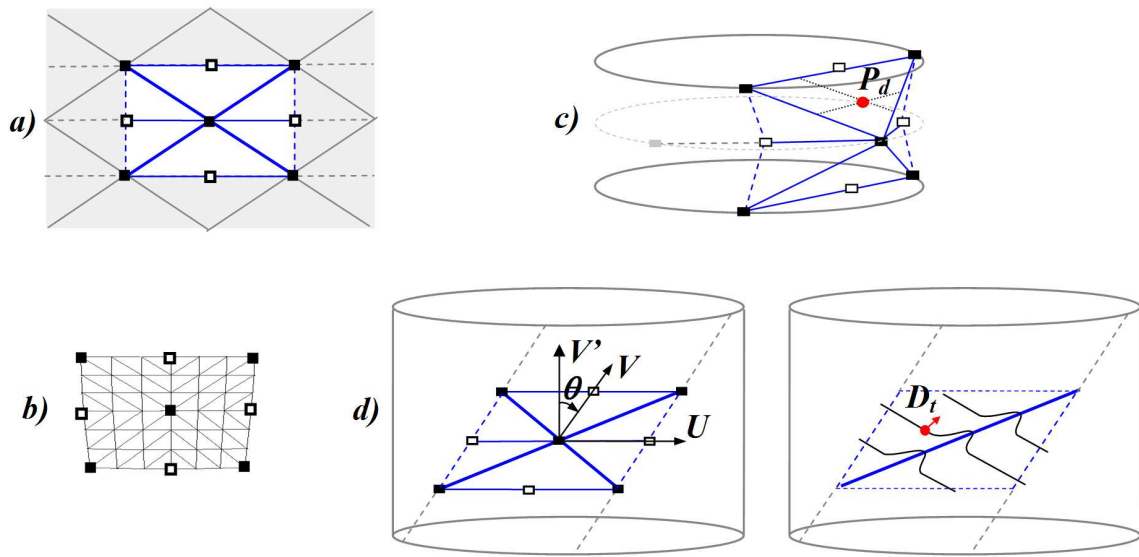
Figure 3: *a) Buckling primitive made of 9 control points that guide the procedural deformation. b) Refined mesh interpolating the control points. c) Moved triangles of a diamond buckling pattern (point P_d). d) Fold across the selected diagonal of a twist buckling pattern.*
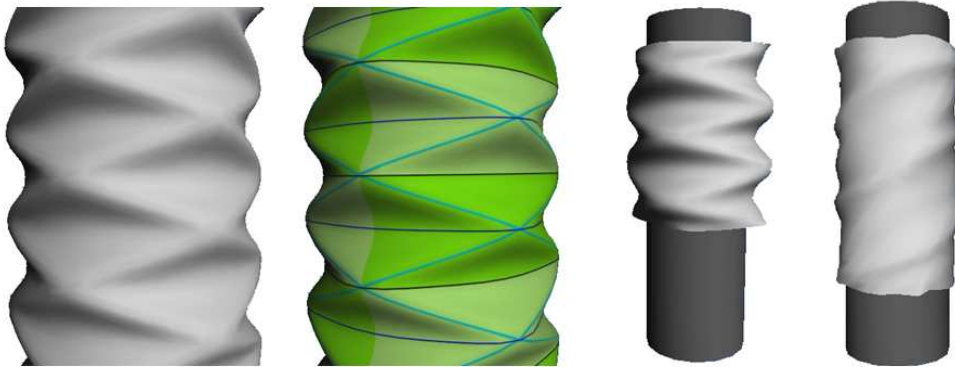

Figure 4: *Diamond and twist buckling on a cylinder with our model.*

To do so, 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 9 control points, depicted in black and white small squares in Figure 3a. Each primitive represents 4 quarters of neighbouring diamond patterns. This scheme allows us to use a quite regular grid of points to control the set of buckling primitives. This is useful for the subsequent physically-based animation of the control points (see section 4), and it eases the creation of a smooth (G^1) interpolating surface over these points, which is needed to create the buckling mesh (as described in the next paragraph).

The specific refined mesh shown in Figure 3 provides node points and edges exactly where needed for having a good approximation of the necessary folds with few nodes (the edges are aligned with the possible wrinkles). The 3D shape of the mesh, blending wrinkles and folds due to diamond and twist buckling modes, is then defined as follows:

1. Cardinal spline interpolation is used to create a smooth patch from the positions of the 9 control points. The tension parameter is not constant: we allow it to vary according to the distance to neighbouring control points, in order to avoid the creation of regions that are too flat or too curved when the control points move. Let P_s denote the position we get on this smooth surface for a given mesh-point P of parameters (u, v) of the buckling primitive.

2. We compute the position P_d the point would have if diamond buckling was the only activated mode: the diamond buckling pattern is formed of 6 triangular parts (see Figure 3c). During compression, these triangles move according to the points of the control mesh. To assure correct folding of the diamond, only 5 of the control points are used

---

[4]If the cylinder of fabric is exposed to axial tension while twisted, the direction of twist lines may be different from those of the diamond mode, but we assume that this situation does not occur for garments moving due to standard motion of a character.

(black dots in Figure 3c) while the remaining 4 points (white dots in Figure 3c) are recomputed from the former. According to (u,v) we know the triangle P belongs to and its relative position inside it. From this we obtain its position P_d on the moved 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 1 if the primitive is totally compressed to 0 if it is not compressed at all.

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 primitive. The diagonal to choose is determined by the sign of the angle θ between direction V and a direction V' orthogonal to U, where U and V are defined by the control points (see Figure 3d, left). A sinusoidal fold across the chosen diagonal is created on the patch (see see Figure 3d, right) so that the length of a thread remains unchanged (compared to its length mesured on the 2D panel). Thus, $D_t = \beta N_s$, where N_s is the normal to the interpolating smooth patch at point P_s, and β varies smoothly to follow the fold (β is equal to the height of the fold on the chosen diagonal and decreases to zero elsewhere).

4. We combine both by defining the final 3D position of the point as:

$$P' = P_s + D_d + D_t \tag{6}$$

5. We then smooth the mesh with a standard square filter: discontinuities on the final surface are introduced through P_d during step 2 because of the sharp edges between the triangular parts. To avoid this and to give the folds generated by the diamond buckling a smooth shape, we apply a square filter of coefficients $(\frac{1}{4}, \frac{1}{8}, \frac{1}{16})$ to the buckling mesh.

Note that if the black points of a buckling primitive were in the same plane (i.e. if the local configuration was planar instead of cylindrical), the white points would stay in this plane as well, yielding a zero value for D_d and suppressing the diamond pattern. This feature of the model will be used in section 4.

3   Garment design using piecewise, approximate unfolding

Our general approach for garment modelling is inspired by [MS04]: we approximate an arbitrary surface by a piecewise developable one. This enables us to start from an arbitrary mesh for the garment. The latter could come from a scan, be created from scratch or through a sketch-based interface. We approximate this surface by another mesh, which makes it look like fabric by incorporating the necessary folds and wrinkles.

It is worth noting that Mitani's solution, which generates strip-shaped patterns, is far from being applicable in our case; fashion designers typically use large fabric panels that limit the total length of seams. Another observation is that anybody, even if not a clothing design specialist, knows approximately where the seams are located on standard garments. So rather than trying to infer this information, we ask the user to provide it.

This gives us a different problem to solve: knowing the border lines and seams of fabric panels in 3D, can we approximate each of them by a developable buckling mesh, as defined in the previous section?

3.1   Approximate unfolding

As in Mitani's work, the user first segments the model into parts roughly corresponding to the limbs of the model, i.e. generalized cylinders, such as the parts of the mesh that cover the arms, the legs and the body. A rough axis and seam lines are defined for each part. In our implementation, seams are just defined as the intersection between the mesh and a plane that includes the cylinder axis. The user can choose between having a single seam such as under a sleeve or two seams, such as on the two sides of the chest. In the latter case, two 2D fabric panels instead of one will be computed to approximate this part of the garment. See Figure 5.

Approximately unfolding a cloth panel is done by computing a one to one mapping between 3D and 2D. [WTY05, MHS99, WSY02] propose good methods to do this in the general case. Here, we rather use our hypothesis that the 3D cloth panel is wrapped around a cylinder. It can then be unfolded by starting along a central line parallel to the axis of the associated limb and progressively flattening it perpendicularly to this direction. This is done in the following way (see Figure6):

- In the 3D space, define a set of regularly spaced planes, called *cross-section planes* normal to the axis of the cylinder.
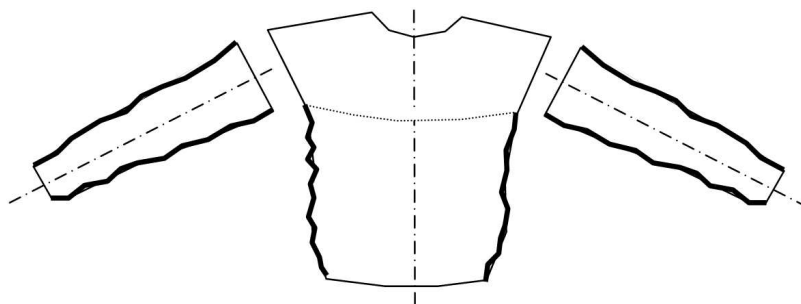
Figure 5: *Separation of a garment into generalized cylinders.*

- Unfold along the direction of the axis by computing the distance, in the space of the fabric, between the cross-section planes: for a set of *axial planes* containing the axis, of different angles with the front plane, cut the geometry with the axial plane and measure the length of the intersection curve between each pair of cross-section planes. The 2D distance, in the space of the fabric, between two successive cross-section planes is set to the average of the length values computed with the different axial planes. The total height of the cloth panel in 2D, along its axis, is the sum of the distances between the cross-section planes.

- Unfold perpendicularly to the axis: for each cross section plane in 3D, on each side with respect to the central axial plane, measure the length of the cross section of the cloth panel (intersect the geometry with the plane and sum the lengths of the edges). This gives the width of one side of the 2D panel for this specific cross-section, with respect to its central, axial line.

In practice, the result is stored as a table giving the 3D coordinates, l and h values for each vertex of each cross section of the initial geometry. This table enables us to retrieve the 3D coordinates of a point (h, l) of the 2D fabric panel using bi-linear interpolation.
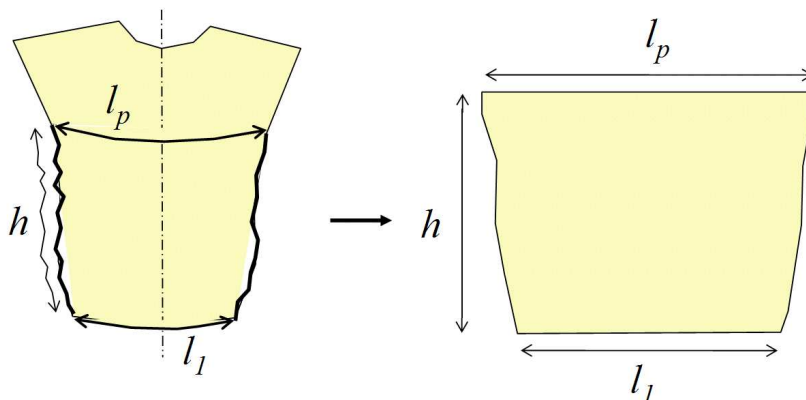

Figure 6: *Unfolding a cylindrical piece of a garment to get the 2D fabric panel.*

Of course, since this process is an approximation based on the hypothesis of a cylindrical panel, it does not flatten the geometry evenly as usual methods do, but rather pushes the 3D bumps to the lateral seam lines of the cloth panel. See for instance the result in the extreme case of the head example in Figure 8 – an obviously not developable shape. The fabric panel for the face is flat but wider in the region of the nose, so approximating the head with it is possible, but will create some folds.

3.2 Buckling mesh generation in 2D

The next step of the process takes place in the 2D space in which the cloth panel has been unfolded. We now have to mesh the 2D panel according to the main folds orientation and direction, as was done for ideal cylinders in section 2.

We first choose the fabric by selecting a value for the parameter λ among those measured on real samples, as explained in Section 2.1. We get the number n of buckling patterns for this piece of garment from equation (5), using an approximate,

global value for the radius R of the pseudo-cylinder (evaluated using a bounding box of the geometry aligned with the axis).
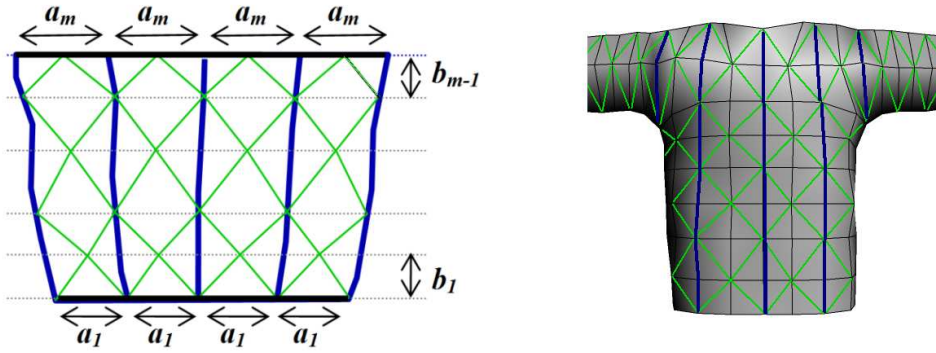

Figure 7: *Left: 2D computation of the control structure of the buckling mesh (positions of buckling primitives and rest distances between control points). Right: 3D view of the resulting control mesh.*

We then compute the set of buckling primitives as follows: starting at the bottom of the 2D panel, we compute the local radius R_1 of the pseudo-cylinder along a horizontal line from the local width l_1 of the 2D panel (using $2\pi R_1 = l_1$ if there is a single panel for this limb, and $\pi R_1 = l_1$ if there are a front and a back panel). We set $a_1 = l_1/n$ and compute b_1 from a_1 and R_1 using equation (1). This gives us the position of a second horizontal line, at the top of a first row of buckling primitives. We continue this way, computing different values a_i and b_i for each successive line. Finally, a grid of buckling primitives is set up from all the a_i and b_i parameters, as depicted in Figure 7, left.

The coarse mesh we obtain in 2D gives us the position and distances between the control points of the buckling mesh in the flat configuration. These distances will serve for activating or not the wrinkle patterns when the refined buckling mesh will be computed in the 3D space. We also use the 2D panel space to define texture mapping if needed.
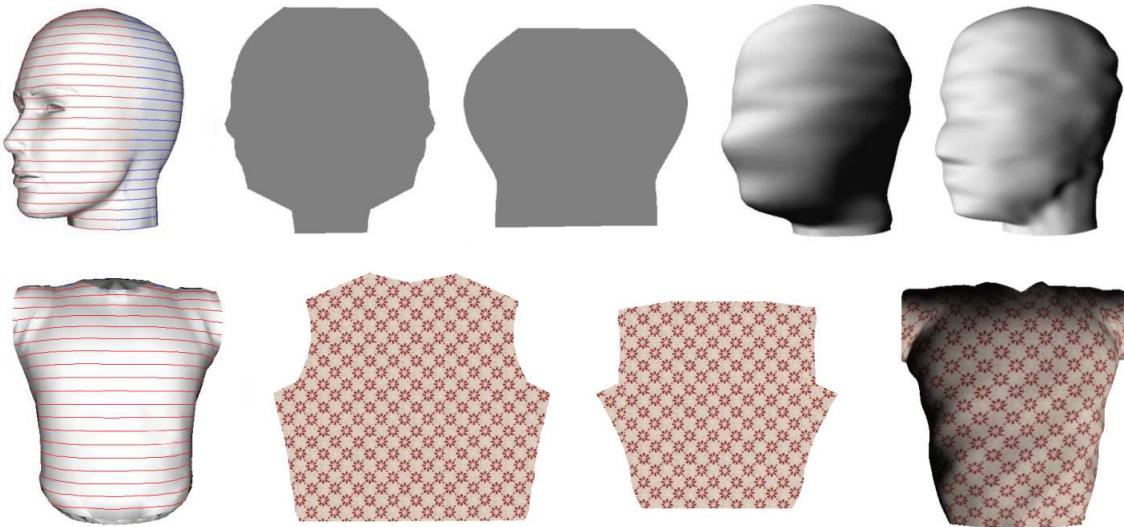
3.3 Final 3D garment

Figure 8: *Validation of our piecewise, approximate unfolding method on an arbitrary shape (top), versus on a model created with an existing cloth design software (bottom). Top: the surface we get looks as a hood, so our approach has achieved giving it cloth appearence (the two results correspond to a thick and then a thinner fabric). Bottom: note the quality of the approximation and the convincing shape of the 2D panels, although we used arbitrary seam lines.*

For each 2D panel, we compute the position of the control points of the buckling mesh in 3D using the correspondence table between 2D and 3D set up in section 3.1. Note that this process automatically maps the seam lines of two neighbouring pattern pieces to the same position, even if these lines did not have exactly the same length in the two 2D panels (due

to the approximations in the unfolding). In this case, the seam will be correct, but the cloth will wrinkle on one side of the seam line. 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 and we manually moved a few control points to fit the chest and sleeves meshes in the sweater case (see Figure 7, right). This is mandatory to subsequently generate a continuous, refined buckling mesh.

Once we have the 3D *control mesh* of the buckling primitives for the whole garment, we apply the procedural generation of the refined buckling mesh, as detailed in section 2, to get the final geometry. Note that the parameters that control the displacement of mesh points such as the rest distances and angles between control points are now local to each buckling primitive. The final result is similar, but thus less symmetric and regular than the ones we obtained on ideal cylinders. We finally get an approximation of the initial, arbitrary surface with a developable surface that behaves like cloth, as demonstrated by the introduction of local folds and wrinkles in Figure 8 (top), where a head model is given the appearance of a hood by our approximation process. A result for an initial surface that looked like a garment is given at the bottom of the same figure, to show the quality of our approximation in this case.

4 Animating Garments

Being able to animate our loose garments is of course essential for most of the applications we have in mind. Our first approach was to animate the control points of the buckling mesh through skinning, and then use the procedural buckling method to add local folds. Although this produces acceptable results for still images, the lack of dynamic effects made the cloth look unnatural during motion. We thus decided on using a physically-based simulation for the control mesh of a garment. This can be done at low cost since this mesh is coarse (see Figure 7) and cloth is very soft at such a large scale: the (desired) compression of the control mesh is used to tailor the wrinkling of the buckling mesh at a finer scale. Meanwhile, the latter will keep the final surface piecewise isometric to the flat fabric panels.

After briefly presenting the physically-based model we are using for the control mesh, the remainder of this section discusses our animation results in terms of both visual quality and performance.

4.1 Modelling the dynamics of loose clothes

The physically-based model for the control mesh is set up as follows. After associating mass values to the control points, we connect them by three kinds of springs, which we call tension, shear and flexion springs, respectively (see Figure 9): the first ones connect adjacent control points on the same line or column; the second type is positioned along diagonals; the third ones connect a point with it's second neighbour on a line or a column. Since the control points are relatively far away from each other on the fabric (see Figure 7, right), there is no need to set up very stiff springs between them (we are currently using stiffness values of 20 to 30 Newton per meter, for masses around 0.0025 kg each). However, the elongation should not exceed the distance between the points in the 2D, unfolded panel. We thus use a modified spring model that combines the spring force with a maximal elongation constraint treated as an impulse [BFA02]. Integration is performed using the Verlet scheme [KANB03]. It allows the use of a large time step (0.01 seconds) in all the examples we computed.

Detecting and processing collisions with such a coarse mesh and large time steps is somewhat intricate: in our running example (see Figure 10) the arm of the character would move outside of the sleeve in a few frames without collision processing and may even pass between two neighbouring lines of control points. We thus detect both the penetration of control points into the character's model (by sending a ray from the point in the direction of its normal) and collisions between the edges of the control mesh and the character's model. This is done by using the library Opcode [Ter01]. In both cases, we use an impulse to re-position the control point (respectively the two control points at the extremities of the edge) in order to avoid the penetration.

In addition to processing collisions with the control mesh, we have to avoid that the procedural buckling process produces intersections between the newly generated folds and wrinkles and the character's body. This can be done very efficiently since the buckling mesh edges are aligned with the possible folds and since we know exactly which of the control points define the bottom of these folds. In practice, we just have to keep the procedural displacement of a few points per buckling primitive (the four white points in Figure 3a.) smaller than their current distances to the body. Moreover, since our geometric process does not create local self-intersections of folds (it would only put neighbouring folds into contact if a cylindrical cloth panel was fully compressed), we do not need to process cloth self-collisions, granted that the character's motion does not put cloth coating different limbs into contact.

The standard simulation method we just described could be replaced by a more sophisticated one to improve the accuracy and/or the efficiency of the simulation. As it is, our implementation was sufficient to validate the combination of our geometric folds and wrinkles with dynamic effects computed at a coarser scale and the application of both in real time on a standard PC.
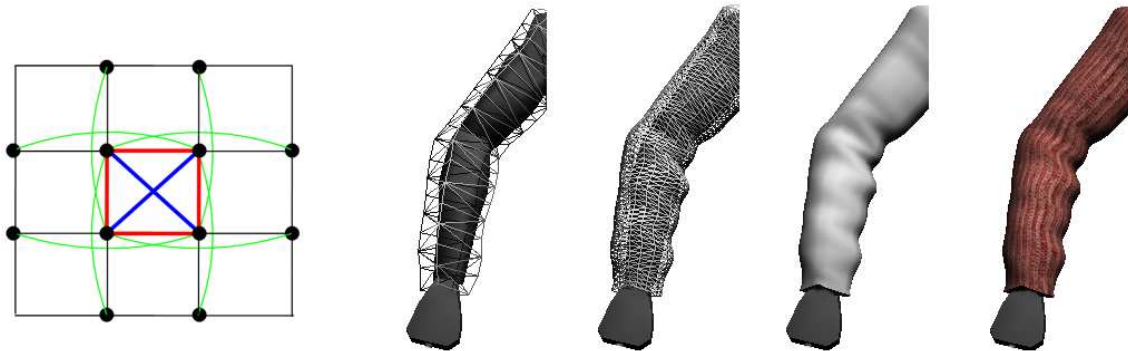

Figure 9: *Left: physically-based model for the control mesh. Red: tension springs, blue: shear springs, green: flexion springs. Right: control mesh and buckling mesh in wireframe for a sleeve model. Resulting shaded surface without and with texture.*

4.2 Qualitative Results

Figure 10: *3D models of garments (top-left) approximated and animated with our developable cloth model (middle), to be compared with the real garments sewed from the 2D patterns we output (bottom-right). Note the wrinkles due to the bending of the arms and to the twist of the body.*

Figures 10 and 11 display some of our results. See also the acompanying video. Our first example is a running character wearing thick trousers and a medium-thick sweater. The second one is a walking, female character wearing a relatively

thick shirt and a very thin skirt (using a better proportioned geometric model for the character would yet greatly improve the quality of the results in the last case).

As they are, these results are sufficient for the qualitative validation of our approach: although diamond and twist fold patterns can be identified (for instance on the sleeves and body parts of the sweater), they are most of the time combined in a complex and beautiful way that makes them look much more natural than in the procedural examples of Figures 4.
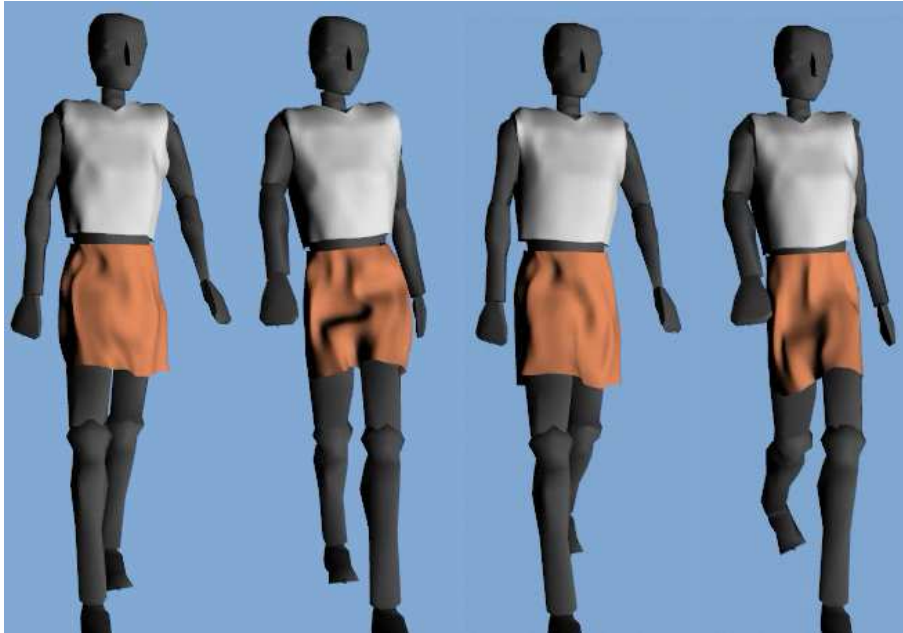

Figure 11: *A walking character wearing a T-shirt and a skirt.*

As a side effect, our method also generates the expected standard sinusoidal folds on the roughly flat parts of the garments (see the skirt model at rest in the video). These extra folds appear when the control mesh is compressed while being in a flat configuration: then, the buckling process does not generate diamond buckling patterns (see the end of section 2.3); since the fine mesh already incorporates vertical and horizontal edges, the smoothing step results in the standard, expected sinusoidal wrinkles, without the need of any further processing.

4.3 Performances

The efficiency of our method is due to the very coarse scale at which the physical simulation is performed: for instance, in the sweater case, we only animate 270 mass points, instead of about 7000 if the fine buckling mesh was used, or even more if an arbitrary mesh not aligned with the possible wrinkles was animated. Thanks to the procedural buckling process, the fine details are added at low cost.

All our animations were computed using a time step of 0.01 sec for the physically-based simulation of the control mesh. The buckling mesh, which includes 128 faces per buckling primitive, is generated every 4 simulation frames, in order to get 25 images per second for display. The benchmarks in the table below were computed on a P4-3GHz (+ GeForceFX 5900). Note that each individual garment can be computed faster than the required 25 images per second, which yields real time performances for a fully dressed character.

	c. pts	prim.	faces	simulation	buckling	fps
Sweater	270	60	7680	52 %	48 %	34.1
Trousers	134	32	4096	51 %	49 %	70.4
Skirt	168	36	4608	50 %	50 %	56

5   Limitations of the method

We have exploited the buckling properties of cloth wrapped around cylindrical limbs. We believe that this is a very common case, but all garments cannot be modelled this way. In particular, our method would fail to capture the behaviour of floating clothes such as very large skirts. In this case, any kind of cloth folds, including the "three-folds" we did not model can occur anywhere on the cloth, so our approach would not be applicable. In addition, we only handle worn garments. Of course, trousers in a washing machine could not be animated with our approach anymore, since the cylindrical shape of the legs would be lost.

Our method uses a fixed buckling mesh. It thus only captures the deformations of ordinary cloth such as cotton fabric (e.g. blue jeans) that keeps a memory of its deformation, resulting in folds and wrinkles that tend to re-appear at the same location each time the same motion is performed. We do not claim that this is true for all fabrics: a polo-shirt made of knit cotton material seems to buckle in a new place for each motion. Modelling this behaviour with our approach is still an open problem.

Lastly, the buckling mesh we generate is developable, by construction, in the initial position, but our method does not insure that the 3D garment will remain perfectly developable throughout the animation. It would obviously not be the case if the control points were allowed to move anywhere. With our animation technique, they are constrained to stay at a maximal distance to their neighbours, which insures that an isolated buckling primitive stays developable with its initial size and shape. Moreover, the shear springs prevent large distortions in the direction of neighbouring primitives, so animation results look correct: for instance, the texture of the sweater does not get visually distorted during motion. However, we cannot claim that the garments we generate stay piecewise developable at anytime (by the way, this is not the case with standard physically based models either, and many real cloth are slightly extensible).

6   Conclusion

We have proposed a novel solution for the animation of loose garments, which cannot be animated both convincingly and efficiently using conventional techniques. A garment is modelled by a constrained mesh that follows the well studied buckling properties of cloth. As a result, adequate folds that make the surface piecewise developable are automatically added to the user-defined initial geometry. Since this process computes the 2D cloth patterns needed to form the garment, distortion-free texture mapping of the cloth patterns becomes straight-forward. Since the cloth mesh already stores the main fold directions, it uses fewer triangles than standard meshes used in cloth simulation and eases collision detection. During animation, the constrained mesh automatically folds and unfolds to follow the underlying character's motion. Dynamic effects such as floating motion are simulated at a low resolution, without requiring the use of a complex, stiff or costly physically-based model. As a result, the whole process is robust and stable, and truely runs in real time.

References

[AH86]   J. Amirbayat and J. W. S. Hearle. The complex buckling of flexible sheet materials, part i. theoretical approach. *International Journal of Mechanical Science*, 28(6):339–358, 1986.

[AH89]   J. Amirbayat and J. W. S. Hearle. The anatomy of buckling of textile fabrics : Drape and conformability. *Journal of Textile Institute*, 80(1):51–70, 1989.

[AP04]   K. Athiannan and R. Palaninathan. Experimental investigations on buckling of cylindrical shells under axial compression and transverse shear. *Sadhana*, pages 93–115, February 2004.

[BA04]   E. Boxerman and U. Ascher. Decomposing cloth. In *ACM-EG Symposium of Computer Animation (SCA)*, September 2004.

[BFA02]   R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics (ACM SIGGRAPH 2002)*, 21(3):594–603, July 2002.

[BMF03]   R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pages 28–36, 2003.

[BWK03]   D. Baraff, A. Witkin, and M. Kass. Untangling cloth. *ACM Transactions on Graphics*, 22(3):862–870, 2003. Proceedings of ACM SIGGRAPH'03.

[CK02]   K-J. Choi and H-S. Ko. Stable but responsive cloth. *ACM Transactions on Graphics*, 21(3):604–611, July 2002.

[Den76]   E. F. Denby. The deformation of fabrics during wrinkling - a theoritical approach. *Textile Research Journal*, 46:667–670, 1976.

[GS66]   P. Grosberg and N. M. Swani. The mechanical properties of woven fabrics, part III: The buckling of woven fabrics. *Textile Research Journal*, 36:332–338, 1966.

[HB00]   D. H. House and D. E. Breen, editors. *Cloth Modelling and Animation*. A. K. Peters Ltd, USA, 2000.

[HBVMT99] S. Hadap, R. Bangarter, P. Volino, and N. Magnenat-Thalmann. Animating wrinkles on clothes. In *IEEE Visualization '99*, pages 175–182, San Francisco, USA, October 1999.

[HLP03]   G. W. Hunt, G. J. Lord, and M. A. Peletier. Cylindrical shell buckling: a characterization of localization and periodicity. *Discrete and continuous dynamical systems*, pages 505–518, November 2003.

[JSAH77]   W. Johnson, P. D. Soden, and S. T. S. Al-Hassani. Inextensional collapse of thin-walled tubes under axial compression. *Journal of Strain Analysis*, 12(4):317–330, 1977.

[KANB03]   Z. Kacic-Alesic, M. Nordenstam, and D. Bullock. A practical dynamics system. In *SCA '03: ACM SIG-GRAPH/Eurographics Symposium on Computer animation*, pages 7–16. Eurographics Association, 2003.

[KG90]   T. L. Kunii and H. Gotoba. Modeling and animation of garment wrinkle formation processes. In *Proc. Computer Animation '90*, pages 131–147, 1990.

[KJL04]   T. J. Kang, K. H. Joo, and K. W. Lee. Analysis of fabric buckling based on nonlinear bending properties. *Textile Research Journal*, 2004.

[Kre91]   E. Kreyszig. *Differential Geometry*. Dover Publications, 1991.

[KWH04]   S. Kimmerle, M. Wacker, and C. Holzer. Multilayered wrinkle textures from strain. In *Proceedings VMV*, 2004.

[LC04]   C. Larboulette and M-P. Cani. Real-time dynamic wrinkles. In *Computer Graphics International*. IEEE Computer Society Press, 2004. Greece.

[May04]   Maya cloth. In *The user manual for designing clothes with Maya*, 2004. http://caad.arch.ethz.ch/info/maya/manual/MayaCloth.

[MHS99]   J. McCartney, B. K. Hinds, and B. L. Seow. The flattening of triangulated surfaces incorporating darts and gussets. *Computer Aided Design*, 31(4):249–260, 1999.

[MS04]   J. Mitani and H. Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph.*, 23(3):259–263, 2004.

[MTCK+04] N. Magnenat-Thalmann, F. Cordier, M. Keckeisen, S. Kimmerle, R. Klein, and J. Meseth. Simulation of clothes for real-time applications. In *EUROGRAPHICS'2004, Tutorial Notes*, sept 2004.

[SyF04]   Syflex. In *The cloth simulator behind many 3D productions*, 2004. http://www.syflex.biz/index.html.

[TCH04]   E. Turquin, M-P. Cani, and J. Hughes. Sketching garments for virtual characters. In John F. Hughes and Joaquim A. Jorge, editors, *Eurographics Workshop on Sketch-Based Interfaces and Modeling*. Eurographics, august 2004.

[Ter01]   P. Terdiman. Opcode: collision detection library. 2001. http://www.codercorner.com/Opcode.htm.

[Tso91]   N. Tsopelas. Animating the crumpling behaviour of garments. In *Eurographics Workshop on Animation and Simulation*, pages 11–23, 1991.

[VMT01]   P. Volino and N. Magnenat-Thalmann. Comparing efficiency of integration methods for cloth animation. In *Computer Graphics International (CGI)*, pages 265–274. IEEE Press, July 2001.

[WSY02]   C. C.L. Wang, S. S.F. Smith, and M. M.F. Yuen. Surface flattening based on energy model. *Computer-Aided Design*, 34(11):823–833, 2002.

[WTY05]   C. C.L. Wang, K. Tang, and B. M.L. Yeung. Freeform surface flattening by fitting a woven mesh model. *Computer-Aided Design*, 37(8):799–814, 2005.