Fast Energy-Based Surface Wrinkle Modeling

Yu Wang [a], Charlie C. L. Wang [b, *], Matthew M. F. Yuen [a]

[a] Department of Mechanical Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

[b] Department of Automation and Computer-Aided Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Abstract

This paper presents an energy-based approach that models the distinct wrinkle shapes to represent the different material properties of non-rigid objects at an interactive speed. Our approach is a curve driven technique, where the surface wrinkles are generated by deforming the given mesh surface according to the shape change of a governing curve on the surface. An energy function is defined on the governing curve to indicate flexure properties. By minimizing the energy function, our approach offers the ability to mimic desirable and pleasing wrinkle shapes corresponding to the given material properties. We then propagate the wrinkle shape of the governing curve on the given mesh surface in the influence region. The final surface wrinkles interpolate the governing curve and are attenuated while they gradually move close to the boundary of the influence region to achieve the smoothness. Consequently, this results in the fast manipulation for complex wrinkle shapes with material properties. The most common problem of physically based simulation, the speed bottleneck, is avoided in our approach. In one word, our approach provides an efficient and useful interactive tool to model realistic wrinkles on non-rigid objects.

Keywords: virtual reality; interaction techniques; computer-aided design; modeling packages; applications of computer graphics.

---

[*] Corresponding Author. Tel: (852) 2609 8052; Fax: (852) 2603 6002

E-mail address: cwang@acae.cuhk.edu.hk (C.C.L. Wang)

mewangyu@ust.hk (Yu Wang) meymf@ust.hk (M.M.F. Yuen).

1. Introduction

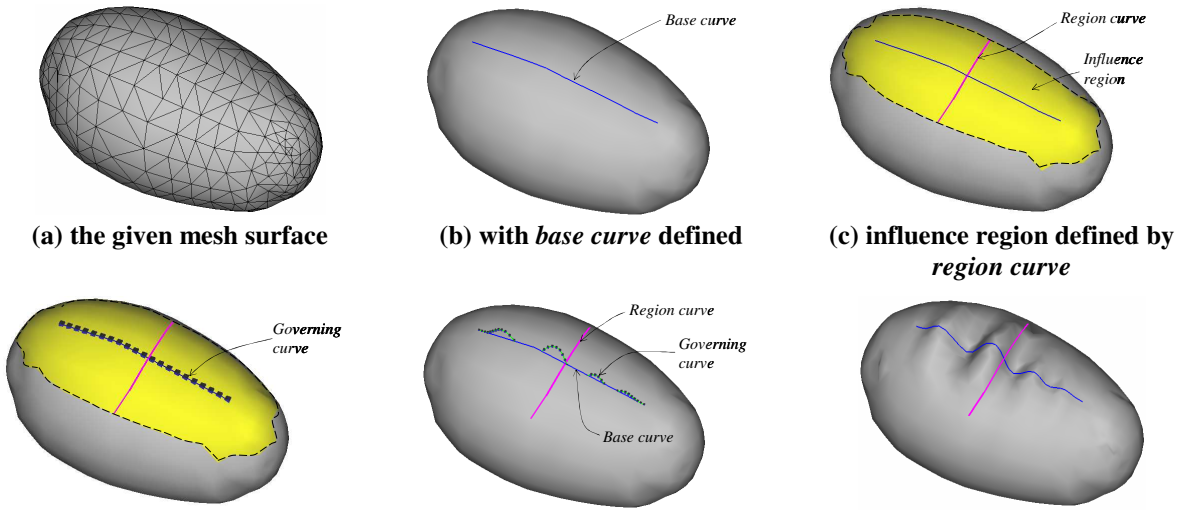
Wrinkle modeling techniques are indispensable in numerous applications, e.g. characters in animations, clothing simulation, freeform modeling in computer aided design, etc. These techniques are particularly necessary since the users always intend to greatly enhance the modeling results with finer detail. Often, a sequence of sculpting operations performing on the existing object would lead to the desired shapes, which are encountered in many real-life objects. In many problems, this set of operations is a critical modeling component. For example, in the animation industry, computer artists need to create skin wrinkles on digital characters to make them more realistic; in a digital fashion show, realistic simulation of clothing on the characters are represented via various wrinkled shapes; a toy designer constructing and editing his work on a 3D CAD system usually enhances its model by adding wrinkles.

In all the applications of surface wrinkle modeling, we encounter two major problems: 1) the realistic representation of the wrinkled shape; and 2) the speed of wrinkle modeling. For the previous problem, the physically based modeling techniques provide very good solutions. Many energy models were developed for research in physically based modeling. These models can be loosely classified into three categories: continuum models, mass-spring models, and particle models.

The full continuum model of a deformable object considers the equilibrium of a general body acted on by external forces, such as the model proposed in some pioneering work [1-3]. Moreover, for shape editing in CAD, many authors adopt FEM with various element types and interpolation functions to solve the continuum models with more robustness and accuracy, e.g. see [4, 5]. Easier to implement and faster than FEM, other works adopt mass-spring models which are extensively used for animation of dynamic behaviors, e.g. [6-8]. However, even by the fastest mass-spring models, it is still difficult to achieve an interactive speed for the modeling of surface wrinkles, especially when the number of mesh vertices is greatly increased for more realistic models. Therefore, the later problem appears. In this paper, we are going to present a fast energy-based surface wrinkle modeling approach which can achieve the interactive speed.

We create wrinkles on the mesh surface by a curve driven shape-editing approach. It could be more clearly described by examples. As illustrated in Fig. 1, with the help of sketching tools, users can rapidly define a *base curve* on the surface of the given mesh (the curve in Fig. 1b); after that, another stroke for a *region curve*, crossing the base curve, is applied to specify the influence region (see Fig. 1c, where the influence region is illustrated by dashed lines). A *governing curve* is then constructed according to the base curve, where the governing curve is actually a list of particles linked by polygonal edges. The underlying energy function which

represents the physical material properties of deformable objects is applied to the governing curve. To direct the wrinkle shape, users are request to input the expected length of the governing curve and a parameter indicating the material stiffness. By computing the minimum curve energy, we can modify the geometry of a governing curve and mimic the wrinkle shape with different material properties. Fig. 1d and 1e shows the shapes of the governing curve before and after energy releasing. The shape of the governing curve is next propagated on the base curve and the given mesh surface through geometry-oriented techniques. The final surface wrinkles interpolate the governing curve and are attenuated while they gradually move close to the boundary of the influence region to achieve the smoothness. During the propagation, adaptive mesh refinement is integrated to enhance the wrinkle effects. The surface wrinkle is actually constructed by a set of curves (a base curve, a region curve, and a governing curve) and two parameters (the target length of governing curve and the stiffness coefficient for wrinkle shape).



(a) the given mesh surface **(b) with *base curve* defined** **(c) influence region defined by *region curve***

(d) the *governing curve* with particles created before energy releasing **(e) the *governing curve* after energy releasing** **(f) the resultant surface wrinkle**

Fig. 1 The wrinkle modeling tool based on our technique

Our wrinkle modeling method achieves a realistic surface wrinkle shape representing the various materials of non-rigid objects without the shortcoming of the high cost in physically based modeling. This is achieved by using the energy-based simulation to obtain the target wrinkle shape on the governing curve and the propagation of deformation via a few desirable geometry-oriented techniques. Actually, the interactive wrinkle modeling tool provided here achieves a balance of the aforementioned two problems – realistic modeling and speed, which is the major contribution of our work.

The remainder of the paper is organized as follows: we first discuss previous related work. Then we introduce the overall algorithm of our wrinkling approach, which is followed by the detail descriptions of the

particle-based curve energy model and the surface wrinkling propagation techniques. Finally, we illustrate some experimental results and show the applications of our approach to process complex models.

2. Related Work

The pioneer work in computer graphics to animate deformable objects was introduced by Terzopoulos et al. [1, 3] using finite differences for the integration of energy-based Lagrange equations. Their energy functions were derived using a continuum formulation. Departing completely from continuum models, particle-based models formulate the energy function on discrete primitives. Our energy model for simulating the governing curve is essentially a particle-based curve model. The first particle model was introduced by Breen et al [8], where a piece of cloth is modeled as a 2D array of particles conceptually representing the crossing points of warp and weft yarns in a plain weave. The particles interact with each other in ways that are determined by the mechanics of the yarns between crossings. Their work was applied to simulate the cloth static drape, as opposed to animation. Speed is the second concern in their work. Later, an extension work [9] offered a new particle model for fast dynamic simulation by the use of higher-order explicit integration methods and Maple-optimized code. Also inspired by the work of Breen et al, Choi and Ko [10] have also presented a mass point model but with the different connectivities of interacting particles. The mechanical interactions that take place between the particles are compression, bending, stretching, and shearing resistance respectively associated with two types of interaction models. The integration scheme they adopted is similar to the scheme conducted in [11, 12], where the implicit solver is used to achieve larger computing steps. Recently, Bridson et al. [13] and Grinspun et al. [14] also studied the problem of wrinkling a given mesh surface. Their physically discrete models [10-14] are similar, where the bending forces are formulated on edge basis – between pairs of adjacent triangles. Nevertheless, even after the implicit integration with larger steps is adopted, the above approaches still cannot create surface wrinkles at an interactive speed. The approach presented in [15] simulated folds by the calculation of surface equilibrium. By means of locally eliminating the difference between current and reference mesh state, their modeling process adds materials through an internal surface growth. In this paper, we propose an energy-based surface wrinkling method, where the length change of a governing curve is somewhat similar to adding material on a curve. However, our method develops for interactive modeling and thus is much less expensive than previous physically based approaches.

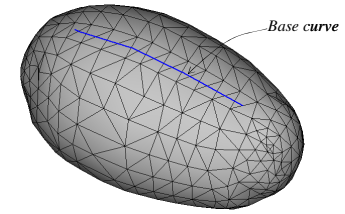
A judicious choice of design tool is imperative both to obtain the desired look and to achieve modeling results within a short time. Hence, geometry-oriented modeling techniques are also considered in our paper.

Free form deformation (FFD) [16] and their extensions [17, 18] are popular and provide a high level of geometric control over the deformation. FFD first gives the definition and deformation of lattice of control points. An object embedded within a lattice is then deformed subjecting to the lattice deformation. However, it is tedious and cumbersome to construct a very dense lattice, when these techniques are used for fine-scale deformations. To solve this problem, an interactive and geometric deformation technique, *wires*, has been developed by Singh and Fiume [19]. Their approach involves the definition of wire curves and domain curves. The wires give the description of an object and shape its deformable features. Domain curves specify the regions of deformation influence. Implicit functions are used to manipulate the deformation of the object through a set of wires which are bounded to the objects. In their paper, they demonstrated the wire which directly reflects object geometry to model the wrinkled surfaces. But as a pure geometric method, wires do not take material properties of objects into consideration. Our method also conducts a deformation technique by curve manipulation. However, instead of the given implicit functions, the underlying physical model is adopted and fine-scale wrinkle shapes are obtained by energy minimization.

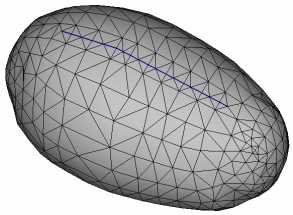
As the physically based wrinkling methods are computationally complex, several geometry-oriented [20, 21] wrinkling methods were developed. Aono proposed a 'wave' model in [20]. Recently, Larboulette and Cani [21] proposed an approach to craft the dynamic wrinkles for animation problems. While they edit wrinkles on an extra layer and establish the constraints between the underlying layers, the dynamic wrinkles are obtained without the need of further edit during production of cartoon animations. According to the observation that the wrinkle typically appears to absorb length changes, e.g. on the forehead when frowning or on clothes when a joint bends, they apply length preservation as the geometric constraint to the wrinkle propagation. Similar to the idea of Larboulette and Cani, our method adopts a curve to control the wrinkle shape in the region of influence, but we conduct the curve shape through a physical model and energy minimization.

Another distinct feature of our approach is that our wrinkle tool is built on a sketch-based modeling platform as proposed in [22]. A natural way to design is offered to complete the wrinkle specification on the surface. SKETCH system [23] rapidly constructs an approximate shape via direct mark base interaction. The Teddy system [24] constructs a rounded free-form mesh model by finding the chordal axis of user input 2D closed strokes to build a smooth surface around the axis, but only closed surfaces are involved. Like SKETCH and the Teddy system, GarSketch3D [25] is designed for rapid construction and modification of approximate models. GarSketch3D was developed for the apparel industry to create and edit apparel products in the 3D space. 2D strokes of both manifold and non-manifold objects can be manipulated in this system. Developing on the

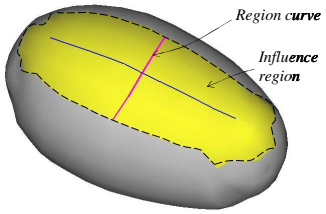
Garsketch3D, our method allows the user to intuitively and efficiently draw directly on the 3D surface to specify where the wrinkles are expected.



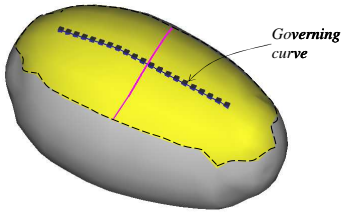
(a) given mesh M with stroked curves



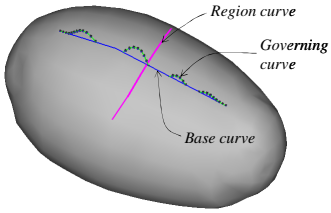
(b) re-triangulated mesh by the stroked curves – the *base curve* created

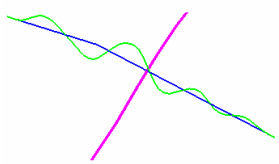


(c) influence region defined by *region curve*

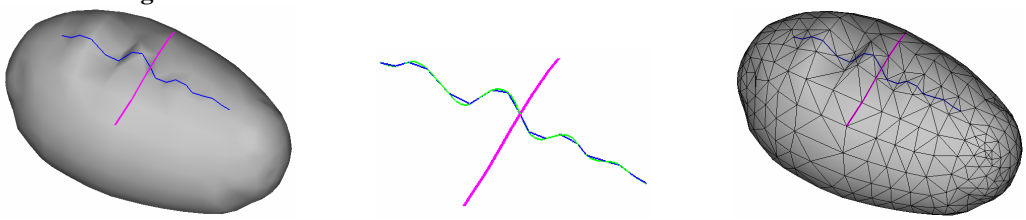


(d) the *governing curve* with particles created before energy releasing

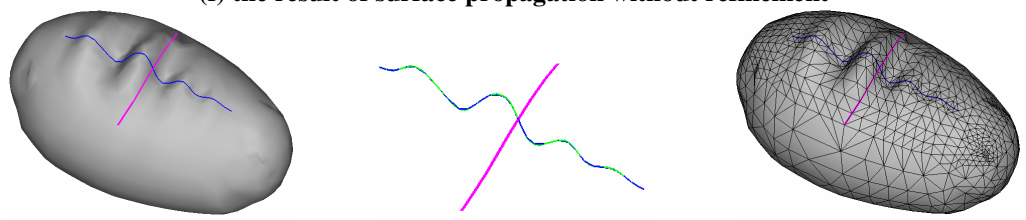




(e) the *governing curve* after energy releasing



(f) the result of surface propagation without refinement



(g) the result of surface propagation with refinement

Fig. 2 Wrinkling process

3. Wrinkling Algorithm

The wrinkling algorithm includes two stages to perform the wrinkling deformation of the given mesh surface: 1) wrinkle shape construction and 2) surface propagation. In this section, we take the example shown in Fig. 2 to help explain the overall algorithm.

In the first stage, which is typically computed once, a *base curves* c_b and a *region curve* c_r are specified on the given mesh surface M (see Fig.2a and Fig.2c). These curves can be specified by either the traditional 3D curve input methods in CAD systems or the sketched-input as shown in [25]. After the user has applied a stroke on the curves, M is re-triangulated by the *Constrained Delaunay Triangulation* (CDT) [26] to make the edges

and nodes of the triangles to conform to the input curves (see Fig.2b). This re-triangulation is necessary for the geodesic distance calculation in influence region search. After that, a governing curve c_p is established as an approximation curve for the base curve c_b. To do this, the particles along c_p are created to uniformly discrete c_b so that form a piecewise linear curve. As illustrated in Fig.2d and Fig.2e, the particles on a governing curve are represented by small dots. Then, based on a user specified flexure stiffness coefficient k_r and the proposed length of the governing curve, the energy defined on the particle system of c_p is released by moving each particle along its surface normal direction (the detail particle curve model will be presented in the following section). Consequently, the newly wrinkled governing curve c_p' is formed as shown in Fig. 2e.

Secondly, the surface in the influence region is deformed by propagating the deflections from the base curve. In above stage, when creating the governing curve c_p from c_b, each vertex v on the base curve c_b has a parameter $t \in [0,1]$ determined with $v = c_p(t)$. By the new governing curve c_p', all vertices of c_b are moved to match the target wrinkle shape given by setting their position to $v' = c_p'(t)$, hence a new base curve c_b' is created. This is illustrated in the middle figure of Fig. 2e. Moreover, we estimate the approximation error between c_p' and c_b' which is measured by the distances between c_p' and the middle-point of each edge on c_b'. This estimated error is used as the criterion for adaptive local mesh refinement. This error is compared against a preset threshold for local mesh refinement. If the maximum error is greater than the threshold ε, all triangles in the influence region would be subdivided into four triangles to repeat the propagation until the approximation error satisfies ε. The vertices on M in the influence region are moved along its original surface normal from its original position by a deflection. The value of deflection is given by an attenuating function in terms of the geodesic distance from a vertex to the base curve and the deflection of the closest vertex on c_b. Fig. 2f and 2g give the example results of surface propagation with and without mesh refinement, in which it is easy to find that the resultant surface with mesh refinement more tightly follows the wrinkled governing curve.

The pseudo-code of the entire wrinkling algorithm is expressed in Table 1. The detailed particle curve model and the surface wrinkle generation scheme are presented in the following two sections successively.

Table 1 The overall wrinkling algorithm

Algorithm WrinkleGeneration ($M, c_b, c_r, k_r, \tilde{L}, r, \varepsilon$)

Input: the given mesh M, the wrinkle control curve c_b, the region curve c_r, the elastic flexure stiffness k_r, the target length of the governing curve $\tilde{L}$, the region radius r, and the approximation criterion ε

Output: the mesh surface M with wrinkle modeled

1. Construct the governing curve c_p by c_b;
2. Minimize the energy function defined on c_p by k_r and $\tilde{L}$ - a wrinkled governing curve c_p' obtained;

repeat {

3. Search the influence region;
4. Compute the deflection of each vertex on c_b - so c_b' determined;
5. Evaluate the approximation error between c_b' and c_p';
6. **if** (the approximation error $\leq \varepsilon$), **then break**;
7. Refinement triangles in the influence region;

}

8. For each vertex in the influence region, determine its closest vertex[+] on c_b;
9. Propagate the deflection of each vertex in the influence region along its surface normal;

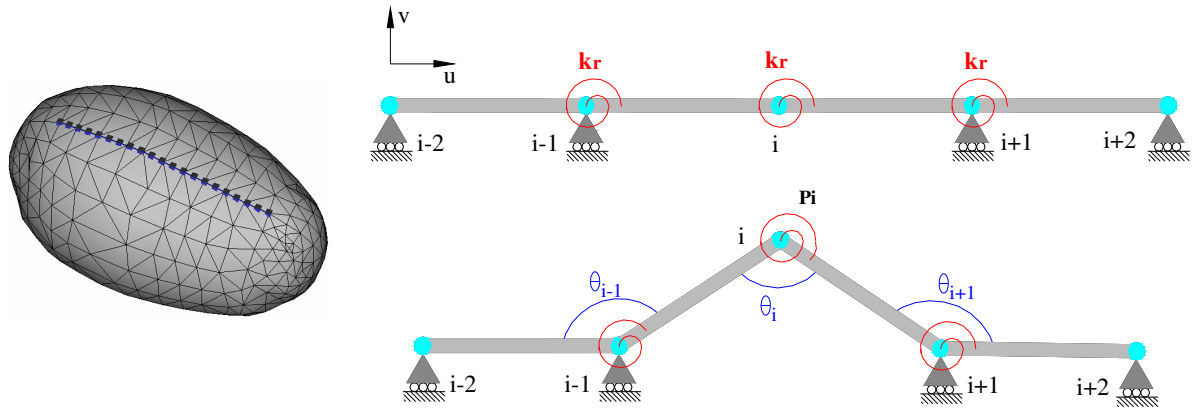


Fig. 3 Particle based model on the 3D mesh: (a) the governing curve c_p is defined on an object. As a piecewise linear curve c_p, a sequence of particles (dots) is regarded as the vertices; (b) The five-particle flexure model.

[+] The closest vertex here is determined by the geodesic distance.

4. Particle Curve Model

In this section, we introduce our discrete energy model for piecewise curves. A sequence of particles is associated with a piecewise linear polygonal curve defined as *governing curve* c_p as shown in Fig.3a. The particles are assumed to be sequentially connected by rotational springs, which simulate the elastic flexure properties of curve c_p and resist the deformation inspired by customized external energy. The model presented in this section is akin to the work of Breen et al [8] which introduces particle models of textile fabrics. In contrast, our model is much more efficient, since piecewise curve is merely considered. The number of particles is dramatically reduced comparing to the approximate model for the entire mesh [8-14]. Thus, it is possible to perform real-time modeling using our physical model. Consequently, a very different and easer way is used here to treat the interacting forces between these particles. First of all, a governing curve is defined to approximate the corresponding candidate curve on the mesh surface.

4.1 Energy model

A governing curve c_p is treated as a particle model with *n* particles and the position of the *i*th particle is $p_i \in \Re^3$. In our energy model, although it seems the governing curve c_p defined on the given mesh M, p_i is independent of the mesh vertices on M. Thus, our particle model is mesh independent. This distinct feature gives many benefits to the final shape of the surface wrinkle, e.g. the target wrinkle shape obtained from a governing curve does not rely on mesh resolution. This also offers the criterion if the mesh refinement is necessary and gives the refinement terminate condition. The material property of a governing curve is customarily described in terms of the potential energy function $E(c_p)$, whose parameters are the positions of particles on c_p.

Here, let us first consider the energy for an individual particle. With the help of configuration as depicted in Fig.3b, we define computing $E_i(p_i)$, flexural energy of particle p_i, only depends on its 4 immediate neighboring particles. For simplicity we start the explanation using a planer deformation. When considering p_i in Fig.3b, the flexure energy at this particle only relies on three angles, θ_j where $j = i-1, i, and\ i+1$. Rotation springs with stiffness k_r are defined at the particles to evaluate the flexure energy. The internal energy defined on each particle is reflected on the curve flexure which shows the curve rigidity resisting the wrinkling deformation after applying the external energy, where the flexural energy of particle p_i is written as

$$E_i(p_i) = \sum_j B(\theta_j) \text{ where } j = i-1, i, \text{and } i+1 \tag{1}$$

$$B(\theta_j) = k_r(e^{-\theta_j} - e^{-\pi}) \tag{2}$$

with θ_j as the angle formed by the segments connecting particles p_{j-1}, p_j, and p_{j+1}. This $E_i(p_i)$ is called the *five-particles flexure model* since five particles are involved to define the rotational energy on one particle. The elastic flexure stiffness k_r is a mesh independent material property. To define the flexure energy, the exponential function is used. This monotony function $B(\theta_j)$ is equal to zero at $\theta_j = \pi$. In that case, the internal energy of a flat curve is completely released. When the curve is sharply folded, the curve energy is increased to the maximum; i.e. the exponential function reaches its maximum when $\theta_j = 0$. Fig.4 illustrates the energy distribution with respective to angle θ_j, where the stiffness coefficient k_r is set to one. The flexure energy of p_i, if $3 \le i \le n-2$, is linearly expressed as the sum of three components – the energy for three angles (see equation 1).

The internal energy of an entire curve is estimated as the total amount of flexure energy for all the particles along the curve. But we found that, for the two endpoints and their neighbors, the five-particles flexure model is still not well defined. In order to ensure G^1 continuity at the two ends of c_p, we simply fixed the end-particles and their nearest neighbors on the particle curve during energy minimization and ignore their particle energies when computing the curve internal energy by the following equation,

$$E_{\text{int}} = \sum_{i=3}^{n-2} E_i(p_i). \tag{3}$$

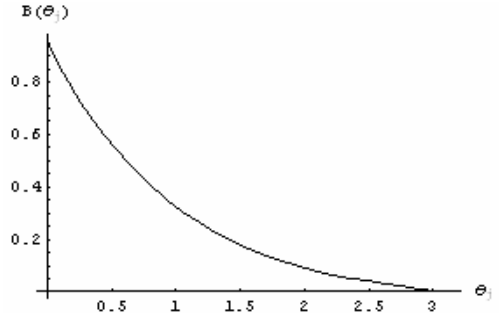


Fig. 4 Flexural energy for angle θ_j ($k_r = 1$)

In our approach, the external energy is user specified and independent of object property. We write the external energy as

$$E_{ext} = (\tilde{L} - L)^2, \tag{4}$$

where L corresponds to the piecewise curve length of un-wrinkled configuration and $\tilde{L}$ is the user expected curve length after wrinkling deformation. In addition, the square formulation is used here, since the quadratic function can be efficiently minimized. The final energy consists of the internal and external energy terms

$$E_{total} = E_{\text{int}} + E_{ext} . \tag{5}$$

These two terms both contribute to the energy minimizing for modifying the particle positions. As discussed above, our particle model measures the flexural energy, when the particles have planner deflection. For a spatial particle curve, the energy function defined on it can be measured in the same manner. The only difference is that the θ_j s in equations (1) and (2) are not coplanar.

4.2 Numerical scheme for minimization

To minimize the energy functional (equation 5) defined on c_p, we change the positions of particles p_i where $i \in [3, n-2]$. Let us first consider the planar particle curve. Without loss of general, we can define a local frame u_i–v_i at each particle p_i, where u_i is the unit tangent of c_p at p_i and v_i is the unit vector determined by rotating u_i anticlockwise with 90 degree. According to the observation that the motion u_i does not affect the elastic deformation of the rotational spring when p_{i-1}, p_i, and p_{i+1} are collinear, to simplify our model, we give only one DOF - v_i to each particle $i \in [3, n-2]$ when releasing E_{total}. In other words, the tangential DOF of particle - u_i is ignored. In our wrinkle modeling algorithm, we need to handle the three-dimensional particle curves which serves as governing curves. Following the simplification of particle movement in the planar case, we compute the surface normal of each particle on c_p, and only adjust particles $i \in [3, n-2]$ along its surface normal.

The new shape of c_p is determined through an energy minimization procedure. In detail, considering about a particle p_i, where the surface normal of M at it is n_i, if we give one DOF – Δ_i at this particle to change its position, its new position during energy minimization is $p_i + \Delta_i n_i$. Thus, the solution vector to minimize E_{total} is defined as $\mathrm{X} = [\Delta_3, \Delta_4, \cdots, \Delta_{n-3}, \Delta_{n-2}]$. The energy minimization problem is then formulated as:

$$\textbf{min}\ E_{total}(\mathrm{X}) .$$

This is an unconstrained optimization problem. As the initial guess of the solution vector is given as $\mathrm{X}^0 = [0,0,\cdots,0,0]$, the optimized configuration X^* which gives the minimal E_{total} could be determined

iteratively through a conjugate gradient scheme [27]. The dimension of solution vector depends on the number of particles conducted on a governing curve; in all our testing examples, we choose 50 particles for each c_p.

Our physical model is a particle-based curve model. It is easily implemented and dramatically reduces the computational complexity compared to the classical physical models of computer simulations. The material properties of modeling objects can be edited by the elastic stiffness of rotation spring, k_r. The external energy applied to the whole curve is intended to generate the deformation. Resisting this deformation, the internal energy expressed as flexure components represents the material properties of objects. In fact, the flexure components try to minimize the bending energy in c_p. Since the bending energy could also be used to measure the smoothness of curves and surfaces (ref. [8]), the smoothness of the governing curve is controlled by the flexure spring model. The bump deflections in the normal of particles are calculated from the underlying physical model. Consequently, the target wrinkle shape formed by the governing curve considers physical properties and can be achieved in real time. As shown in Fig. 5, with the increase in k_r and the fixed $\tilde{L}$, the particle curve behaves progressively stiff by the energy releasing. The experimental results follow our expectation – the greater k_r, the particle curve with less wrinkles is determined. In our current implementation, the flexure stiffness is generally set in the range $5 \le k_r \le 70$ and the target length $\tilde{L}$ is chosen as between 1.2 to 1.7 times of *L*.

(a) **(b)** **(c)** **(d)**

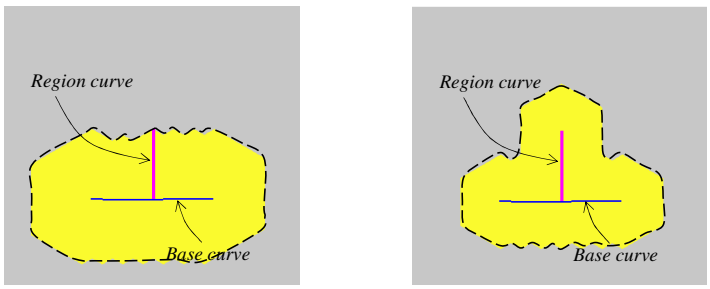
Fig. 5 Different k_rs leading to different physical properties of the particle curve

5. Surface Wrinkle Modeling

As mentioned in the section of overall algorithm, the surface wrinkle is constructed in a customized region by propagating the energy functional generated governing curve. Two options to define the influence region are developed for different applications. As a mesh processing technique for finer details, the adaptive refinement is critical and offered in our approach to achieve enhanced results. Our approach also allows the manipulation with multiple sets of wrinkling curves to obtain results with complex geometry. All the surface wrinkling operations are geometry-oriented, which allows the generation of a surface wrinkle at an interactive speed. Let's first think about the influence region defined on the given mesh surface M.

5.1 Influence region

We provide two ways to define the influence region of surface wrinkle propagation for different applications. In the following illustrations, all the influence regions are circled by dashed lines (e.g., Fig. 6). In the first mode of influence region definition, the region is centered on the base curve with a unified radius *r*, which is the maximum geodesic distance computed from the region curve (as shown in Fig. 6a). In the second mode of influence region definition, the region is defined on the vertices with its geodesic distance to the base curve and the region curve less than a user specified radius *r* on M (see Fig. 6b). To compute the geodesic distance W_{v_i} from each vertex v_i to the source vertices (in region definition mode I, the source vertices are vertices on the base curve; in region definition mode II, the source vertices are vertices on both the base curve and the region curve), an approximation geodesic computing method similar to [28] is conducted (the detail algorithm is given in Appendix).



(a) region definition mode I **(b) region definition mode II**

Fig. 6 Two options of influence region definition

5.2 Propagation of deflections

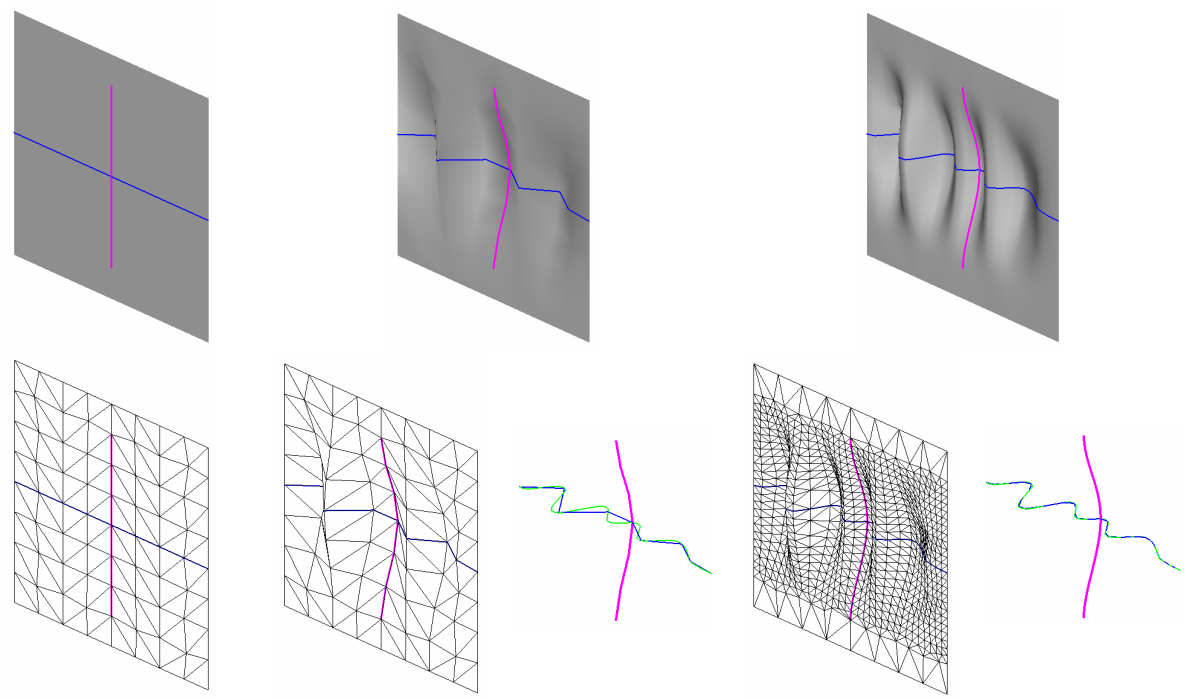
As mentioned in ***Algorithm*** WrinkleGeneration(…), the new position of each vertex in the influence region is determined by a deflection along the surface normal at this vertex. The deflection of a vertex v_i is defined by an attenuating function multiplied by the deflection of v_i's closest vertex on the base curve. The attenuating function f is selected to describe the shape of the influence region given that attenuating function is set to be one along the base curve, and zero at the boundary of the influence region. The order of the attenuating function, f, should be so chosen such that it preserves the continuity across the influence region boundary. Consequently, the attenuating function should be at least C^1 and monotonically decreasing with $f(x)=1,\ x\le 0$ and $f(x)=0$, for $x\ge 1$. Two attenuating functions can be considered: 1) a bell shaped one as adopted in [20] [21],

$$f(x)=\begin{cases}1+\dfrac{(-4x^6+17x^4-22x^2)}{9}, & x\in[0,1]\\ 0, & otherwise\end{cases}, \tag{6}$$

or 2) an implicit function as used in [18],

$$f(x)=\begin{cases}(x^2-1)^2, & x\in[0,1]\\ 0, & otherwise\end{cases}. \tag{7}$$

In our current implementation, we choose the one in equation (6) with $x=\frac{W_{v_i}}{r}$ as the input of an attenuating function. For all vertices in the influence region, the values of $\frac{W_{v_i}}{r}$ are always between zero and one.



(a) given surface with a base curve and a region curve

(b) the surface wrinkling result by our approach but without adaptive refinement

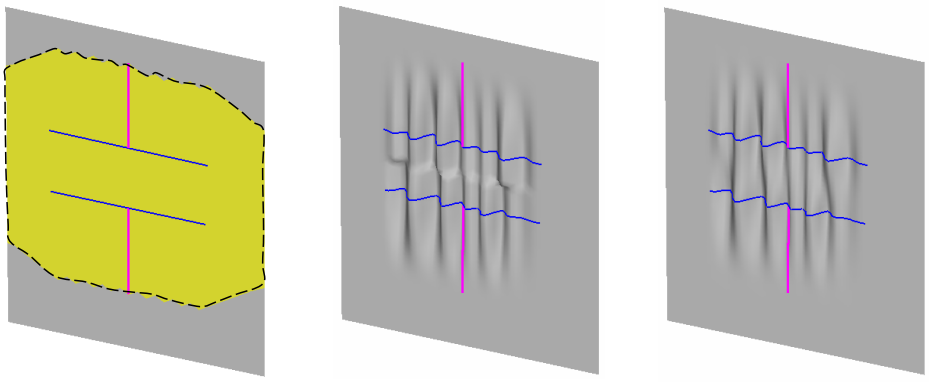
(c) the surface wrinkle result by our approach with adaptive refinement

Fig. 7 Adaptive refinement enhances the surface wrinkle propagation

5.3 Adaptive refinement

The steps 3-7 in ***Algorithm*** WrinkleGeneration (…) are actually an adaptive refinement process, where the distance error between the deflected base curve c_b' and the wrinkled governing curve c_p' is measured to detect whether the surface needs to be refined to get more accurate results. Since during the wrinkle modeling, only the vertices in the influence region are relocated, also only triangles in the influence regions are subdivided into 4 triangles to increase the degree-of-freedom for the surface wrinkle modeling. After subdivision, the 2nd order

umbrella operator in [29] is applied on the newly introduced vertices to adjust their positions so that the mesh after subdivision is smoothed. A comparison of the surface wrinkle propagation with and without mesh refinement has already been given in Fig.2f and Fig.2g, Fig.7 shows another example to illustrate the enhanced result by adaptive refinement. In Fig.7b, you can clearly see the difference between the unrefined base curve and the governing curve. Next in Fig. 7c, they are closely coincident.



(a) **(b)** **(c)**

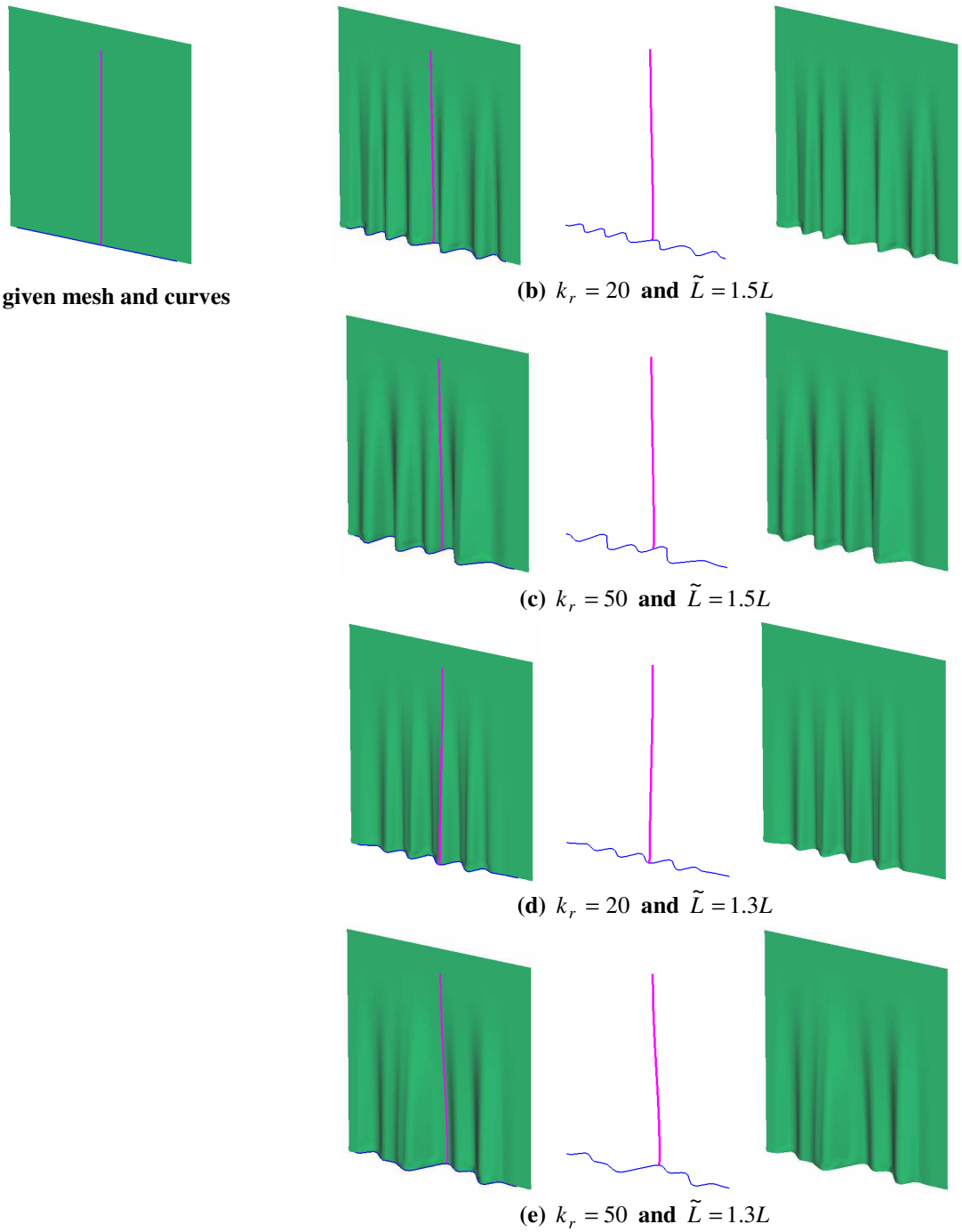
Fig. 8 Overlapping influence regions from two sets of wrinkling curves

5.4 Multiple sets of wrinkling curves

The surface propagation associated with multiple sets of wrinkling curves (where each set consists of a base curve, a region curve, and a governing curve) can also be provided by our approach, the major purpose of which is to manipulate more complex geometries. We also find that the usage of multiple sets of wrinkle curves can enhance the efficiency of shape editing operations. When the multiple sets of wrinkling curves are employed on one object, the result is distinct from summarizing the deformations of individual sets of wrinkling curves. Our result can be a blending of the propagation from multiple curves. For instance, in Fig.8a, the bright region (yellow) shows the union influence region for two couples of region and wrinkle base curves which are defined by the 1st region specification methods referred to above. There is an obvious overlapped influence region between these two sets of wrinkling curves. When simply using the above wrinkling algorithm on multiple c_b s, the deflection of a vertex in the overlapped influence region is determined by its closest vertex among the vertices on all c_w s. This leads to the serious discontinuity resultant surface wrinkle as shown in Fig.8b. Here, we offer a blended deflection for the vertices falling in the overlapped influence regions. For a vertex v in the influence regions from n sets of wrinkling curves, the geodesic distance from it to the ith wrinkle base curve is D_i, and the deflection of its closest vertex on the ith c_b is F_i. We first determined the minimum geodesic distance $D_{\min}$ among all D_i s. If $D_{\min} \neq 0$, the following equation would be adopted to compute the blended deflection of v.

$$F_v = \sum_{i=1}^{n} \frac{F_i}{D_i} \Big/ \sum_{i=1}^{n} \frac{1}{D_i} \tag{6}$$

Otherwise, when $D_{\min} = 0$, the deflection of the closest vertex on the c_b providing $D_{\min}$ would be given by F_v (see equation (6)). Based on this enhancement, the final resultant surface from multiple wrinkling curves is as shown in Fig.8c, where the surface change between the two sets of wrinkling curves is smooth.



(a) given mesh and curves

(b) $k_r = 20$ **and** $\tilde{L} = 1.5L$

(c) $k_r = 50$ **and** $\tilde{L} = 1.5L$

(d) $k_r = 20$ **and** $\tilde{L} = 1.3L$

(e) $k_r = 50$ **and** $\tilde{L} = 1.3L$

Fig. 9 Curtain wrinkles (in (b), (c), (d), and (e), the rows from left to right represent the curve driven surface propagation, the driven curves, and the resultant surface wrinkles

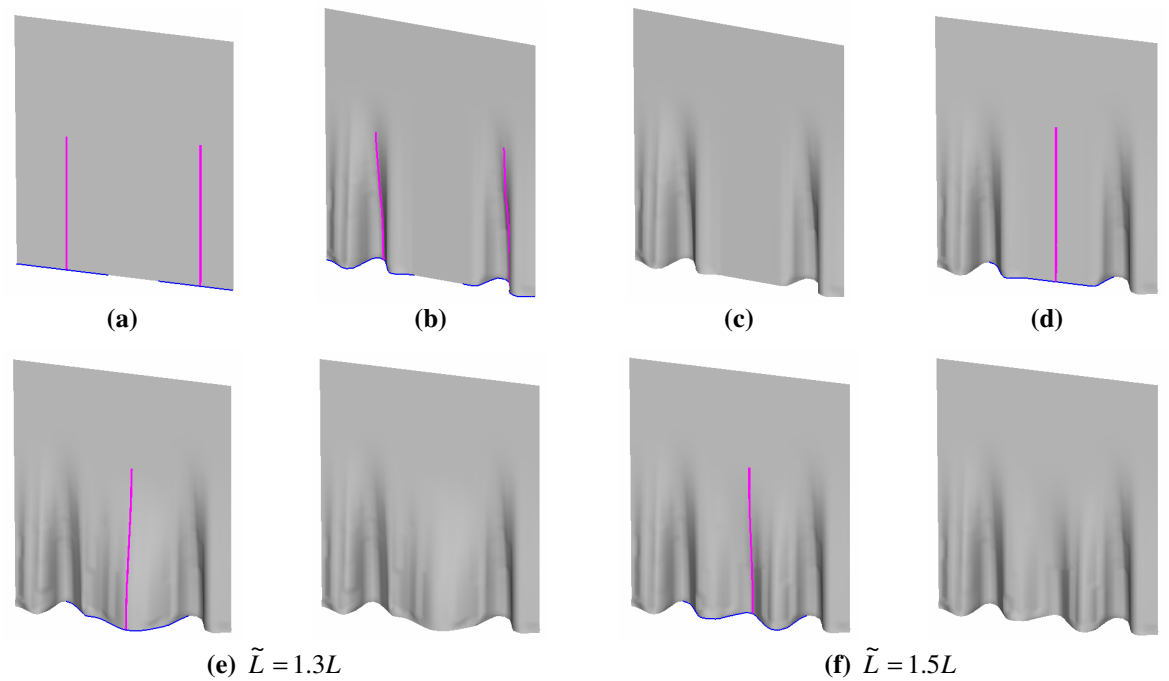


Fig. 10 Two curtain's wrinkle results from different length factors

6. Applications and Experimental Results

Wrinkles and creases can greatly enhance the realism of deformable objects. This section illustrates the versatility of our wrinkling tool with several experimental results that demonstrate different aspects of our wrinkle deformation technique. These results depict how our wrinkling curve is used to control the deformation, propagation on a surface, and localized to increase surface detail. Our approach is a hybrid of physically based and geometry-oriented modeling. All the operations of surface wrinkle modeling shown in this section are finished at an interactive speed on a PC with standard configuration by our prototype system written in Visual C++ and OpenGL.

6.1 Curtain wrinkles

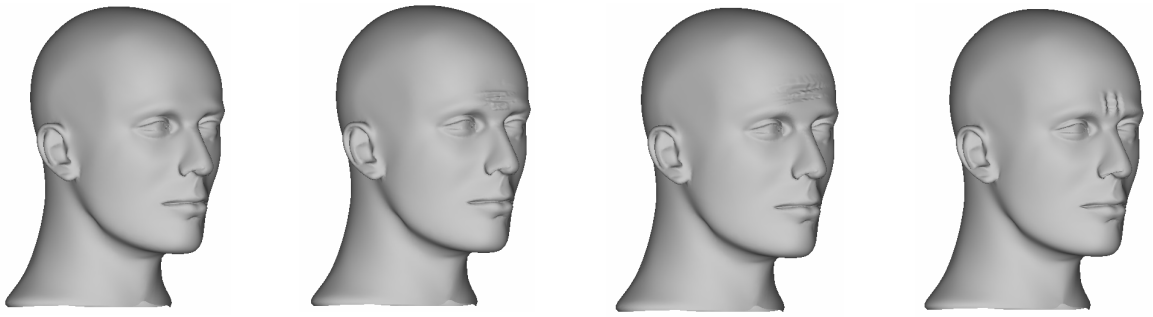
The first example is an application of our tool in modeling wrinkles on curtains. By the same input wrinkling curves, with the increase in flexure stiffness, our model mimics the object material property with more rigidity against the bending and its performance is more likely to resist the 'wave' shape. It illustrates how the resultant surface wrinkle shape depends on the physical model definition. Fig.9 shows the curtain-like wrinkle propagation on the given rectangular planar mesh with the association of one set of wrinkling curves. Beginning with the same wrinkling curves as shown in Fig.9a, a set of distinct curtain wrinkles can be obtained by tuning the stiffness coefficient k_r and the target length $\tilde{L}$ (see Fig. 9b-9e).

The second example of our paper demonstrates the curtain wrinkles formed by multiple sets of wrinkling curves. We procedurally creased the curtain within two operating stages. First, wrinkles formed at two end

portions of the curtain (see Fig 10a-10c). Then, we apply a new pair of control and region curves in the middle of the curtain (Fig. 10d). Two different curtain-like wrinkle shapes are created with two different target length, $\tilde{L} = 1.3L$ for the result in Fig. 10e, and $\tilde{L} = 1.5L$ for the case illustrated in Fig. 10f.

6.2 Forehead wrinkles

Our wrinkling technique is also adopted to simulate facial wrinkles and demonstrated in Fig. 11. Three results mimicking the forehead wrinkles are shown. To achieve such finer features like the facial wrinkles, it is not absolutely infeasible if without the local adaptive refinement. Furthermore, the gentle external energy and small stiffness coefficient ($k_r = 10$ and $\tilde{L} = 1.3L$) are assigned to simulate the soft property with respect to bending motion. Given the wrinkling curves, for each result in Fig.11, the wrinkling operation takes around 2 seconds to process on the PIV 1.8G + 512 MB RAM for the head model with 66737 polygons.



(a) given head model **(b) result 1** **(c) result 2** **(d) result 3**

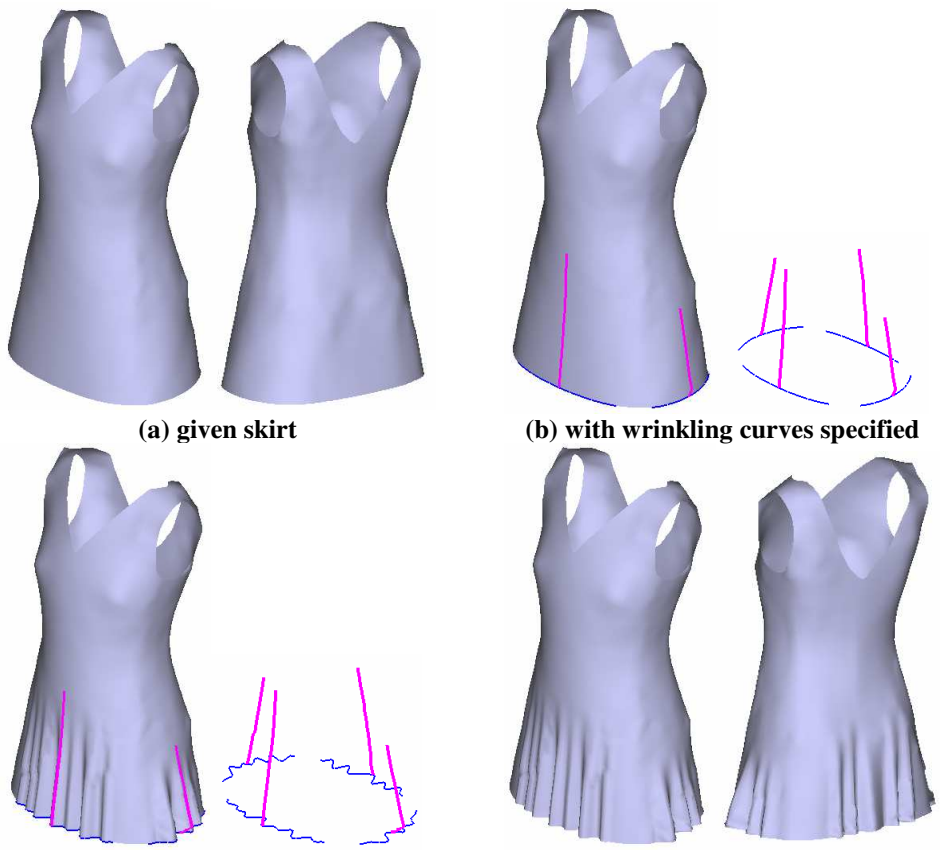
Fig. 11 Forehead wrinkles

6.3 Apparel wrinkles

Our fast surface wrinkling technique also provides a useful surface editor for apparel products. As illustrated in Fig. 12, we can see two views of the skirt initially from Fig. 12a and four sets of wrinkling curves are drawn on this smooth skirt (see Fig. 12b). The apparel wrinkles can be rapidly propagated over the skirt's lower portion, inspired by the base curves in the region limited by the region curves (see Fig. 12c). Fig. 12d shows the final resultant skirt after modeling the wrinkles on it, where the 2D patterns of this apparel are changed after the wrinkles are generated on the mesh.

6.4 Shoe wrinkles

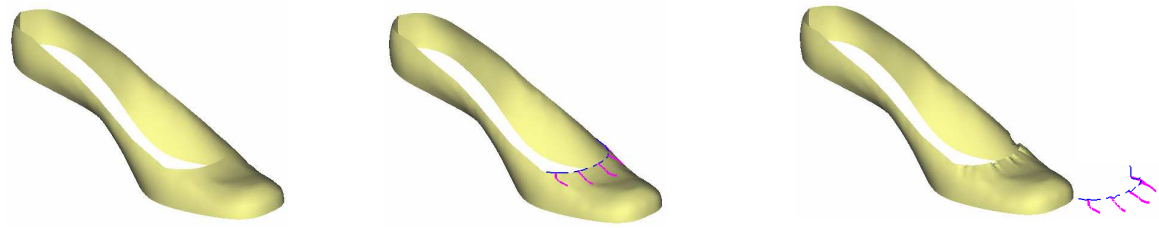
This example is an application in shoe wrinkle design. In order to obtain the fashionable design, the designer expects to edit some wrinkles on the upper surface of a ballroom shoe. To do this, we utilized four sets of wrinkling curves. Fig. 13a shows the shoe before wrinkling. Next, wrinkling curves are specified on the upper surface as illustrated in Fig. 13b. Finally, the wrinkling result and corresponding deformed wrinkling curves are given in Fig. 13c.



(a) given skirt (b) with wrinkling curves specified

(c) result with wrinkling curves (d) resultant surface

Fig. 12 Apparel wrinkles



(a) given surface (b) four sets of wrinkling curves specified (c) the wrinkling result

Fig. 13 Wrinkles on a ballroom shoe from four wrinkle control curves

7. Conclusion and Discussion

The paper presents an energy-based approach that generates the distinct wrinkle shapes to represent the different material properties of non-rigid objects at an interactive speed. This effective technique is based on the integration of physically based and geometric-oriented modeling to achieve visually realistic results. The surface wrinkle is generated by deforming the given mesh surface according to the shape change of a user-specified governing curve on the surface. An energy function is defined on the governing curve to indicate flexural properties. By minimizing the energy function, our approach offers the ability to mimic vivid wrinkle shapes corresponding to the given material properties. The curves on the mesh surface defined as base curves trace the

target wrinkle shapes provided by the governing curves. Meanwhile, the position errors of these tracing operations are offered as the criterion of local refinement. We then propagate the waved shape of the governing curve on the given mesh surface to form the final surface wrinkles, which interpolate the governing curve and are attenuated while they gradually approach the boundary of the influence region to achieve the smoothness. Checking the refinement criterion, the mesh surface is adaptively refined while the wrinkle shapes propagate in the influence region. The manners prevent the inveracious results caused by improper mesh resolution. Consequently, this results in a fast manipulation for complex and material based wrinkle shapes. Compared to other related work, our approach has the following advantages:

- The most common problem of physically based simulation, the speed bottleneck, is solved in our approach since physical simulation is only applied on the particle curves;
- The physical simulation is independent of the resolution of the given mesh surface. The simulation accuracy depends only on the customized particle density along each governing curve;
- Only several sketches are required to model surface wrinkles and the designers are not expected to have high-level computer and good artistic skills;
- Distinct wrinkle shapes can be generated to illustrate different physical properties of objects via tuning two energy parameters – the stiffness coefficient and the target curve length;
- The adaptive local subdivision is applied in our technique to adjust the resolution around the wrinkle mesh and it is a excellent way to achieve smooth and realistic wrinkles;
- Furthermore, our approach allows the multiple sets of wrinkling curves work together to achieve a complex modeling result and simplify the operating process at the same time.

In summary, our novel approach provides an efficient and useful interactive tool to model lifelike wrinkles on non-rigid objects.

Based on our current implementation, if two wrinkle base curves are crossing each other, the wrinkle results will lose the interpolation around the curve intersection point. This is because the two governing curves are deformed individually without consider if they have cross point. When two wrinkle base curves respectively interpolate the governing curves, the curve intersection point cannot be simultaneously subject to the deformations from two governing curves. This may even cause singularity deformation problem. One possible solution that will be investigated in our future work is to process the crossing governing curves in an integrated energy minimization to obtain the unified target wrinkle shape.

Acknowledgment

The authors would like to acknowledge the financial support from HKUST6234/02E project to graduate student Yu Wang during the course of this research, and thank the anonymous reviewer for the helpful comments.

Appendix A fast method to compute the geodesic distances to a set of vertices

A set of source vertices - Ψ is set as including all vertices of c_b for the region definition mode I. If in region definition mode II, Ψ contains the vertices of both c_b and c_r. The approximate geodesic distances from each vertex on M to the source vertices are determined by an advancing method, which progressively moves the event list L_v of nodes away from Ψ on the given surface M. Before starting to move the event list, the geodesic distance W_{v_i} of each vertex in Ψ is set to zero, other vertices are initialized to $+\infty$, and the length of every triangular edge e_j is calculated and stored. Our algorithm repeatedly moves L_v away from Ψ on M; during the movement, the approximate geodesic distance W_{v_i} and the passed flag fp_{v_i} of the nodes neighboring L_v are updated. The vertices in the influence region are the ones with $W_{v_i} \le r$. The pseudo-code of a fast method to compute the geodesic distance from each vertex on M to Ψ is given below.

Algorithm AdaptiveMapGeneration(M , Ψ)

Input: The given mesh surface M, and a set of source vertices Ψ.
Output: The updated geodesic distance W_{v_i} of every triangular node.

1. **for** every node $v_i \in G$ {
2. $W_{v_i} \leftarrow +\infty$;
3. Set the passed flag of $v_i - fp_{v_i}$ to **false**;
4. **if** ($v_i \in \Psi$), **then** add v_i to L_v, $W_{v_i} \leftarrow 0$, and set the passed flag of $v_i - fp_{v_i}$ to **true**;
5. }
6. Calculate the length l_{e_j} of every edge $e_j \in G$, and store the minimum edge length as $l_{\min}$;
7. $L_v' \leftarrow \phi$ and $\lambda \leftarrow l_{\min}$;
8. **do**{
9. **for** every node $v_k \in L_v$ {
10. **for** every node v_j adjacent to v_k {

11. **if** ((W_{v_k} + the length of edge $v_j v_k$) < W_{v_j}), **then** $W_{v_j} \leftarrow W_{v_k}$ + the length of edge $v_j v_k$;

12. **if** ((fp_{v_j} is **false**) and ($W_{v_j} < \lambda$)), **then** add v_j to L_v' and set fp_{v_j} to **true**;

13. }

14. }

15. **for** every node $v_k \in L_v$

16. **if** any fp_{v_i} of its adjacent node v_j is **false**, **then** add v_k to L_v';

17. Replace L_v by L_v' and make L_v' empty;

18. $\lambda \leftarrow \lambda + l_{\min}$;

19. }**while**($L_v = \phi$);

Reference:

[1] Terzopoulos D and Fleischer K, Deformable models. The Visual Computer, 1988; 4:306-331.

[2] Terzopoulos D and Fleischer K, Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. Computer Graphics (Proc. SIGGRAPH) 1988; 22: 269-278.

[3] Terzopoulos D, Platt JC, and Barr, AH, Elastically deformable models. Computer Graphics (Proc. SIGGRAPH) 1987; 21: 205-214.

[4] Collier J, Collier B, O'Toole, G, and Sargand S, Drape prediction by means of finite-element analysis. Journal of the Textile Institute 1991; 82(1): 96-107.

[5] Celniker G and Gossard D, Deformable curve and surface finite-elements for free-form shape design, Proceeding of SIGGRAPH'91 Las Vegas, Nevada 28-August2 July.1991 p. 257-266.

[6] Provot X, Deformation constraints in a mass-spring model to describe rigid cloth behavior. Proceeding of Graphics Interface '95, Canadian Human-Computer Communications Society, May 1995, p. 147-154.

[7] Haumann D R and Parent RE, The behavioral test-bed: obtaining complex behavior from simple rules, The Visual Computer 1988; 4:332-347.

[8] Breen DE, House DH and Wozny MJ, Predicting the drape of woven cloth using interacting particles, Proceeding of SIGGRAPH '94 Orlando, FL, July 1994; p. 365-372.

[9] Eberhardt B, Weber A, and Strasser,W. A fast, flexible, particle-system model for cloth draping. IEEE Computer Graphics and Applications 1996; 16:52-59.

[10] Choi KJ, and Ko HS, Stable but responsive cloth, Proceedings of SIGGRAPH 2002, San Antonio, Texas, USA July 2002; p.604-611.

[11] Baraff D and Witkin A, Large steps in cloth simulation, Computer Graphics (Proc. SIGGRAPH) 1998; p. 1-12.

[12] Kang YM, Choi JH, Cho HG, and Lee DH. An efficient animation of wrinkled cloth with approximate implicit integration. The Visual Computer 2001; 17(3): 147-157.

[13] Bridson R, Marino S, and Fedkiw R, Simulation of clothing with folds and wrinkles, Proceedings of Eurographics/SIGGRPH Symposium on Computer Animation, 2003.

[14] Grinspun E, Hirani A, Desbrun M, and Schroder P, Discrete shells, Proceedings of Eurographics/SIGGRPH Symposium on Computer Animation, 2003.

[15] Combaz J and Neyret F, Painting folds using expansion textures, Proceeding of Pacific Graphics 2002, p.176--183.

[16] Sederberg T and Parry S, Free-form deformations of solid geometric models, Computer Graphics (Proc. SIGGRAPH) 1986; p. 151-160.

[17] MacCracken R and Joy K, Free-form deformation with lattices of arbitrary topology. Computer Graphics 1996; p.181-189.

[18] Faloutsos P, Van De Panne M, and Terzopoulos D, Dynamic free-form deformations for animation synthesis, IEEE Trans. on Visual and Computer Graphics 1997; 3:201-214.

[19] Singh K and Fiume E, Wires: a geometric deformation technique, Proceeding of SIGGRAPH'98, Orlando, Florida, USA, July 1998; p. 405-414.

[20] Aono M, A wrinkle propagation model for cloth, Proc. 8th International Conf. of the Computer Graphics Society on CG International '90 1990; p.95-115.

[21] Larboulette C and Cani MP, Real-Time Dynamic Wrinkles, Proceeding of Computer Graphics International, Greece 2004.

[22] Wang CCL, Sketch based 3D freeform object modeling with non-manifold data structure, Ph.D. Thesis, Hong Kong University of Science and Technology, June, 2002.

[23] Zeleznik RC, Herndon KP, and Hughes JF, SKETCH: an interface for sketching 3D scenes, Proceeding of SIGGRAPH, 1996; p.163-170.

[24] Igarashi T, Tanaka H, and Matsuoka S., Teddy: a sketching interface for 3d freeform design, Proceeding of SIGGRAPH, 1999; p. 409-416.

[25] Wang C.C.L., Wang Y., and Yuen M.M.F., Feature based 3D garment design through 2D sketches, Computer-Aided Design 2003, 35(7), p.659-672.

[26] Ganapathy S., Dennehy T.G., A new general triangulation method for planar contours, Computer Graphics, vol.16, no.3, pp.69-75, 1982.

[27] William H., *Numerical recipes in C: the art of scientific computing* (2nd Ed.), Cambridge: Cambridge University Press, 1995.

[28] Wang C.C.L., Wang Y., Tang K., and Yuen M.M.F., Reduce the stretch in surface flattening by finding cutting paths to the surface boundary, Computer-Aided Design, vol.36, no.8, pp.665-677, 2004.

[29] Kobbelt L., Campagna S., Vorsatz J., and Seidel H.-P., Interactive multi-resolution modeling on arbitrary meshes, Proceeding of SIGGRAPH 1998.