A Multi-Resolution Cloth Construction Method Based on Low-Resolution Cloth Sampling (Postprint)

Authors: Zhong Litao, Hou Jin, Gong Sui, Zhang Juan, Tang Yuanhao

Date: 2018-05-20T00:00:00+00:00

Abstract

To balance the realism and computational efficiency of cloth simulation, we propose a multi-resolution cloth construction method based on low-resolution cloth sampling. First, by performing multiple samplings on low-resolution cloth motion simulation instances, we obtain the average deformation degree of each cloth region during the simulation process, which is represented using vertex average deformation degree and edge collision markers. Then, based on the average deformation degree, the low-resolution mesh regions are divided into high-deformation zones, medium-deformation zones, and low-deformation zones. Subsequently, an improved adaptive subdivision algorithm is employed to subdivide the three types of deformation zones to varying degrees, thereby constructing a multi-resolution cloth geometric model corresponding to the low-resolution cloth. Finally, based on the multi-resolution geometric model, the cloth particle masses and spring coefficients are defined to obtain a multi-resolution physical model. Experimental results demonstrate that compared with high-resolution cloth, multi-resolution cloth reduces the number of meshes and improves computational efficiency; compared with low-resolution cloth, it enhances simulation realism.

Full Text

Preamble

Title: Multi-resolution Cloth Construction Based on Low-resolution Cloth Sampling

Authors: Zhong Litao, Hou Jin†, Gong Sui, Zhang Juan, Tang Yuanhao
Affiliation: School of Information Science & Technology, Southwest Jiaotong University, Chengdu 611756, China

Abstract: To balance the fidelity and computational efficiency of cloth simulation, this paper presents a multi-resolution cloth construction method based on low-resolution cloth sampling. First, by sampling a low-resolution cloth motion simulation instance multiple times, we obtain the average deformation degree of each cloth region during the simulation process, represented by the average vertex deformation degree and edge collision markers. The low-resolution mesh regions are then classified into high, medium, and low deformation zones based on the average deformation degree. Next, an improved adaptive subdivision algorithm is applied to subdivide the three deformation zones to varying degrees, thereby constructing a multi-resolution geometric model corresponding to the low-resolution cloth. Finally, based on the multi-resolution geometric model, we define the cloth particle masses and spring coefficients to obtain the multi-resolution physical model. Experimental results demonstrate that compared with high-resolution cloth, multi-resolution cloth reduces mesh count and improves computational efficiency; compared with low-resolution cloth, it enhances simulation fidelity.

Keywords: cloth simulation; multi-resolution mesh; region division; adaptive subdivision; low-resolution cloth sampling

0 Introduction

Cloth simulation technology has become a research hotspot in virtual reality, a rapidly emerging industry in the 21st century. As a soft and flexible material, cloth undergoes complex deformations under external forces, making the realistic and efficient simulation of wrinkles a significant challenge. An ideal approach would involve high-precision modeling using dense meshes, but such meshes often contain tens of thousands of geometric elements, imposing heavy computational burdens and memory requirements. To achieve realistic wrinkle effects without increasing computational cost, various wrinkle enhancement methods have been proposed.

Cutler et al. [1] marked wrinkle baselines on clothing surfaces and generated three-dimensional wrinkles based on these baselines, custom wrinkle shapes, and radii. Rohmer et al. [2] also simulated wrinkles based on wrinkle baselines derived from coarse mesh simulation analysis, offering greater objectivity. Zurdo et al. [3] combined low-resolution and high-resolution cloth simulation examples to propose a method for quickly simulating realistic wrinkles. Jing et al. [4] proposed a wrinkle enhancement algorithm based on coarse meshes. While these methods simulate cloth details more realistically, they are only applicable to static cloth.

To authentically simulate dynamic wrinkle effects, Li et al. [5] proposed a cloth simulation method based on triangular mesh adaptive subdivision and simplification, using curvature criteria for mesh operations combined with the Baraff model for garment animation. Simnett et al. [6] approached the problem from an edge perspective, proposing an edge-based adaptive mesh subdivision and simplification method for cloth simulation. A more novel approach is the reverse simplification adaptive mesh method proposed by Lee et al. [7], which first establishes a high-resolution cloth model through Loop surface subdivision [8] and then reduces mesh count by simplifying dynamically flat regions. Numerous other studies [9–12] have also simulated dynamic cloth using adaptive mesh methods.

Another common approach for dynamic cloth simulation is the example-based method [13–15], which reconstructs dynamic garment effects from video data and actual motion data. To improve real-time performance, Aguiar et al. [16] proposed an augmented linear dynamic system for fast and stable synthesis of garment animation. Wang et al. [17] simulated garment wrinkles using extensive example data. To predict cloth motion trends, Shi et al. [18] proposed a multi-resolution mesh cloth animation method based on example data analysis, which achieves good detail deformation features under similar motion drivers and effectively improves computational efficiency, though extracting bending deformation modes is time-consuming.

Adaptive mesh methods require continuous updates to mesh topology and force relationships during simulation, resulting in high computational complexity and poor continuity. Example-based methods consume considerable time in the animation synthesis process. To address these issues, this paper proposes a multi-resolution cloth construction method based on low-resolution cloth sampling. The main idea is to sample and analyze the deformation degrees of cloth mesh regions from a low-resolution simulation instance, classify the cloth model into high, medium, and low deformation zones, and then apply an improved adaptive subdivision algorithm to subdivide high-deformation zones once or twice, medium-deformation zones once, and leave low-deformation zones unchanged. This constructs a multi-resolution cloth model suitable for similar motions. The method features short preprocessing time, low computational complexity, and can preserve wrinkle details well in simulations driven by similar motions while allowing control over simulation fidelity and computational efficiency through adjustable deformation zone classification thresholds.

1 Average Deformation Degree Acquisition

Although cloth deformation under external forces is complex, the distribution of deformation regions in the same motion remains consistent. For example, when a clothed human walks, wrinkles mainly appear in the waist area and regions colliding with the legs. Moreover, under stable external forces (i.e., without instantaneous large forces), wrinkle generation and changes remain relatively stable [18]. Based on this observation, to quickly and effectively obtain the deformation degree of cloth mesh regions under specified motions, we sample and analyze the deformation degrees of low-resolution cloth during a specified motion pattern to obtain the average deformation degree throughout the simulation process. We then use this sampling data to predict and classify the deformation conditions of each cloth region.

1.1.1 Vertex Weight Deformation Degree

The wrinkled regions during cloth motion correspond to deformed regions in the triangular mesh. While Gaussian curvature is a common metric for deformation degree, its computation is complex and lacks robustness [8]. Another common metric is the dihedral angle [19] (the angle between two triangular faces), which, although less computationally intensive than Gaussian curvature, suffers from lower accuracy. To more accurately measure mesh region deformation, this paper proposes a vertex weight deformation degree that combines the advantages of Gaussian curvature and dihedral angle while fully considering the influence of triangular face area on deformation.

First, we define the unit normal vector of mesh vertex v_i:

$$\mathbf{n}_i = \frac{\sum_{j=1}^{N_i} \mathbf{N}_{ij} \cdot S_{ij}}{\left\| \sum_{j=1}^{N_i} \mathbf{N}_{ij} \cdot S_{ij} \right\|}$$

where N_i is the number of faces adjacent to vertex v_i, $\mathbf{N}_{ij}$ is the normal vector of the j-th adjacent face, and S_{ij} is the area of the j-th adjacent face.

The vertex weight deformation degree for a triangular mesh is defined as:

$$D_i = \frac{\sum_{j=1}^{N_i} (\mathbf{n}_i \cdot \mathbf{N}_{ij}) \cdot S_{ij}}{N_i \cdot \sum_{j=1}^{N_i} S_{ij}}$$

where N_i is the number of faces adjacent to vertex v_i, $\sum_{j=1}^{N_i} S_{ij}$ is the total area of adjacent faces, $\mathbf{n}_i$ is the unit normal vector of vertex v_i, $\mathbf{N}_{ij}$ is the normal vector of the j-th adjacent face, and S_{ij} is the area of the j-th adjacent face. The 1-neighborhood geometry information of a vertex is shown in [Figure 1: see original paper].

According to this definition, if the deformation degree of a vertex' s 1-neighborhood region is low, the vertex' s unit normal vector and its adjacent faces' unit normal vectors are nearly parallel, making the vertex weight deformation degree approach 1. Conversely, if the deformation degree is high, the vertex weight deformation degree approaches 0.

1.1.2 Edge Collision Marking

In addition to using vertex weight deformation degree to represent mesh deformation, this paper also marks edges that experience collisions, as shown in Figure 2: see original paper. Since mesh regions where edges collide with objects should also deform, these collision edges require subdivision to more realistically display simulation effects, as shown in Figure 2: see original paper. During multi-resolution geometric model construction, marked edges participate in the subdivision process.

1.2 Deformation Degree Sampling and Processing

To reduce sampling and processing time and improve computational efficiency, we sample the vertex weight deformation degree of each vertex in the low-resolution cloth dynamic process. The sampling count K is determined based on the complexity of deformation in the low-resolution cloth simulation. For cloth with diverse and complex deformation forms, a larger K is selected; otherwise, a smaller value is used. The chosen K is far less than the total frame count of the dynamic simulation. Since cloth doesn' t undergo large deformations between every two frames and each deformation state persists for some time, sampling once before the state changes can capture the deformation features.

Let the vertex deformation degree obtained for the i-th vertex during the k-th sampling be D_i^k. After K samplings, the average deformation degree is:

$$\bar{D}_i = \frac{1}{K}\sum_{k=1}^{K} D_i^k$$

We use the processed vertex average deformation degree as the standard for measuring the deformation magnitude of the low-resolution cloth vertex' s 1-neighborhood region throughout the simulation, and then determine how many times the vertex will participate in subdivision based on its relationship with subdivision thresholds.

2 Multi-resolution Geometric Model Construction

This paper employs an improved adaptive subdivision algorithm based on vertex average deformation degree and edge collision marking, using the Loop subdivision pattern. The following definitions are necessary for the algorithm. To classify mesh regions into three deformation zones, we first set two threshold parameters β_1 and β_2 (where $0 \leq \beta_1 \leq \beta_2 \leq 1$). When a vertex' s average deformation degree $\bar{D}_i$ is less than β_1, its activity is set to 2; when greater than β_2, activity is set to 0; when between β_1 and β_2, activity is set to 1.

If the sum of activities of a triangle' s three vertices exceeds 3, the region is classified as high-deformation; if less than or equal to 3 but greater than 0, it' s medium-deformation; if equal to 0, it' s low-deformation. We subdivide high-deformation zones once or twice, medium-deformation zones once, and leave low-deformation zones unchanged.

A vertex with activity 0 is called a fixed vertex. An edge is called a fixed edge when the sum of its two endpoint activities equals 0; otherwise, it' s a live edge. When a mesh edge is marked as a collision edge, it is also considered a live edge. A triangular face is fixed when all three vertices are fixed; otherwise, it' s a live face. During subdivision, fixed vertices, fixed edges, and fixed faces are not subdivided. Thus, vertex average deformation degree, collision edge marking, and threshold parameters control the entire mesh' s adaptive subdivision.

2.1 Multi-resolution Geometric Model Construction

Algorithm: Improved Adaptive Subdivision Algorithm

a) Label all vertices, edges, and faces in the mesh as fixed or live when created.

b) Traverse triangular mesh vertices, assign the processed average deformation degree from low-resolution mesh sampling to corresponding vertices, and update vertex activity based on the relationship between average deformation degree and threshold parameters.

c) Traverse all triangular faces, update edge and face activity based on the number of fixed vertices in each face and collision edge marking.

d) Update all vertex positions using the Loop subdivision mode' s new vertex calculation formula.

e) Calculate new edge points. For fixed edges, no new edge points are calculated; for live edges, calculate new edge points using Loop subdivision formulas and set their activity as the average of endpoint activities.

f) Generate new faces. Fixed faces are not subdivided; live faces are subdivided according to the number of live edges they contain, as shown in [Figure 3: see original paper].

g) Traverse all vertices, decrement their activity by 1, and set to 0 if activity becomes negative.

h) If high-deformation zones are only subdivided once, end subdivision; if high-deformation zones undergo multiple subdivisions and vertices with activity > 0 remain after decrementing, return to step d).

Figure 3: see original paper shows the original mesh before subdivision. (b) and (c) show subdivision cases when a face contains two live edges. When the vertex average deformation degree exceeds the threshold, subdivision follows Figure 3: see original paper; otherwise, it follows Figure 3: see original paper. Figure 3: see original paper shows the standard Loop subdivision case with three live edges.

2.2 Multi-resolution Cloth Model Parameters

2.2.1 Particle Mass This paper uses the mass-spring model as the physical model. Given a fixed cloth model size, the total mass should be identical for both low-resolution and multi-resolution models. However, multi-resolution models have non-uniform particle distributions. If all particles have equal mass, the multi-resolution mesh model' s total mass would exceed that of the low-resolution mesh, violating mass conservation and causing deformation distortion in simulation. To maintain realistic mass properties, we determine particle mass based on its influence domain.

Let cloth density be constant ρ. The influence domain A_i of particle i is defined as one-third of the total area of its adjacent triangular faces:

$$A_i = \frac{1}{3}\sum_{j=1}^{k_i} A_{ij}$$

where k_i is the number of faces adjacent to the particle and A_{ij} is the area of the j-th adjacent face. After calculating the influence domain, the particle mass m_i can be approximated as:

$$m_i = \rho \cdot \tau \cdot A_i$$

where τ is the cloth model thickness (set to 1 in this paper).

2.2.2 Spring Coefficients In multi-resolution mesh models, different regions have varying mesh sizes, requiring different elastic coefficients for springs of different lengths. Generally, spring elastic coefficients are inversely proportional to spring length. To prevent "super-elasticity" caused by excessive stretching, we employ a piecewise linear elastic force calculation:

$$\mathbf{F} = \begin{cases} k_1 \cdot l & \text{if } l < \tau \\ k_1 \cdot \tau + k_2 \cdot (l - \tau) & \text{if } l \geq \tau \end{cases}$$

where l is spring deformation, τ is a threshold, and k_1 and k_2 are different elastic coefficients. This method simulates the nonlinear relationship between spring force and deformation in real environments, as shown in [Figure 4: see original paper].

3 Collision Detection and Response

Collision processing in this paper involves two scenarios: cloth-sphere collision and garment-human model collision. Cloth-sphere collision detection and response are relatively simple and not detailed here; the key challenge is handling edge penetration with spheres. We address this by inflating the sphere radius and using the inflated sphere for collision detection and response calculations.

For garment-human model collisions, to improve simulation real-time performance, we improve upon the algorithm in [20] by using ellipsoidal bounding boxes for adaptive fitting of the human model. During dynamic simulation, these ellipsoidal bounding boxes completely replace the human model for motion and collision processing with garments. The adaptive construction of ellipsoidal bounding boxes and motion schematic are shown in [Figure 5: see original paper].

4 Algorithm Framework

The overall algorithm framework is shown in [Figure 6: see original paper], consisting of two main phases: preprocessing and dynamic simulation. In the preprocessing phase, we first sample mesh region deformation degrees from a low-resolution cloth simulation instance while marking collision edges. The sampled data is processed to obtain average deformation degrees throughout the simulation. Based on these degrees, mesh regions are classified into high, medium, and low deformation zones. The improved adaptive subdivision algorithm then subdivides these zones to varying degrees, yielding a multi-resolution geometric model. In the dynamic simulation phase, we define particle masses and spring coefficients for the multi-resolution mesh, apply gravity or wind forces, process collisions between cloth particles and bounding boxes, and obtain the final multi-resolution cloth simulation results.

5 Experimental Results and Analysis

Experiments were conducted using cloth and two skirt models under three conditions: (a) sphere-cloth collision with four fixed corners; (b) wind force applied to skirt hems with upper portions fixed; (c) clothed human motion. To balance real-time performance, high and medium deformation zones underwent only one subdivision during adaptive subdivision. For comparison with other methods, high-deformation zones were subdivided twice in later experiments. Algorithm comparisons used two additional scenarios detailed in Section 5.3. We analyzed our method by comparing simulation effects and efficiency across different mesh resolutions and against methods in [9] and [18]. Numerical calculations used the method from [21]. The experimental environment was an Intel(R) Core(TM) i3-2350M CPU@2.30 GHz with VS 2010 and C++.

5.1 Multi-resolution Mesh Model Construction

Experiments used a cloth model and two skirt models (skirt 1 and skirt 2). We first performed specified sampling counts on mesh deformation degrees during low-resolution simulation, processed the results, and classified mesh regions into low, medium, and high deformation zones based on thresholds. provides data on face counts, sampling times, and simulation costs for different low-resolution models, with corresponding results shown in [Figure 7: see original paper]. Thresholds were set sequentially as 0.99/0.98, 0.97/0.96, and 0.96/0.95 for the three models.

Based on the classification results, the improved adaptive subdivision algorithm subdivided different zones to varying degrees: no subdivision for low-deformation zones, one subdivision for high and medium-deformation zones. This yielded multi-resolution geometric models for each experimental model, as shown in [Figure 8: see original paper]. Adaptive subdivision thresholds were set to 0.99/0.97 for cloth and skirt 1, while skirt 2 used two threshold sets (0.96 and 0.95) for comparison.

Comparing Figure 8: see original paper and (d) shows that different subdivision thresholds control the proportion of high-resolution mesh within the multi-resolution mesh, meeting varying application requirements for effect and efficiency. The results demonstrate that multi-resolution mesh density distribution aligns with actual regional deformation degrees, validating that vertex weight deformation degree effectively measures mesh region deformation and that average deformation degree serves as a reliable metric for overall simulation deformation. Thus, the proposed method effectively classifies dynamic deformation regions and correctly constructs multi-resolution geometric models.

5.2 Simulation Effect and Efficiency Comparison

Dynamic simulation comparisons were performed using low-resolution, multi-resolution, and high-resolution meshes for cloth and two skirts. Comparison metrics included mesh face count, frame rate, step size, average per-frame simulation time, and total simulation time. presents the simulation data, with effects shown in [Figure 9: see original paper]-[12].

Low-resolution mesh simulations exhibited poor visual quality, appearing rigid and losing substantial wrinkle detail. Multi-resolution mesh simulations closely approximated high-resolution results, showing realistic wrinkles in frontal cloth regions ([Figure 9: see original paper]) and wind-driven skirt deformation ([Figure 10: see original paper]). For skirt 2, multi-resolution and high-resolution models produced similar overall effects with wrinkles in waist and leg regions ([Figure 11: see original paper], [Figure 12: see original paper]), confirming that the multi-resolution construction algorithm preserves detail information in high-deformation zones and improves simulation fidelity.

Comparisons of mesh face counts and per-frame simulation times are shown in [Figure 13: see original paper] and [Figure 14: see original paper]. Multi-resolution meshes significantly reduced triangle counts and per-frame simulation time compared to high-resolution meshes while maintaining similar step sizes. Specifically, multi-resolution cloth reduced face count by 31.7%, doubled frame update frequency, and improved per-frame efficiency by ~50%; skirt 1 reduced face count by 23.1%, increased frame rate by ~0.5×, and improved per-frame efficiency by ~40%; skirt 2 reduced face count by 45.8%, increased frame rate by 4×, and improved per-frame efficiency by ~75%. Overall, multi-resolution meshes outperform high-resolution meshes in both computational and update efficiency while preserving more detail than low-resolution meshes.

5.3 Algorithm Analysis and Comparison

To verify superiority, we compared our method with [9] and [18]. compares simulation time and frame rate with the open-source ArcSim system from [9] for two scenarios: single sphere collision (scene 1) and dual sphere collision (scene 2). [Figure 15: see original paper] shows corresponding simulation effects. compares results with [18] using skirt 1 and cloth models. Remesh time comparisons with

[9] are shown in [Figure 16: see original paper].

As shown in , our method spends considerable time in low-resolution sampling and multi-resolution model construction during preprocessing, but these operations occur only once, minimally impacting overall performance. Method [9] performs dynamic remeshing each frame based on deformation and edge collisions, affecting both continuity and efficiency. and [Figure 16: see original paper] show our method achieves higher frame rates and better continuity. Preprocessing time is constant across scenarios, while dynamic remeshing in [9] becomes increasingly costly. Our method demonstrates superiority after 50 frames in scene 1 and 35 frames in scene 2, with advantages becoming more pronounced in longer simulations. Simulation effects ([Figure 15: see original paper]) show comparable realism between methods, both effectively displaying wrinkle details.

As shown in , our method samples low-resolution meshes (320 and 606 faces) 15–20 times, while [18] samples high-resolution meshes (2,424 and 1,280 faces) 1,560 and 1,108 times respectively—nearly 100× more sampling operations. Our data processing uses low-complexity averaging, reducing computational cost. Thus, our multi-resolution mesh construction algorithm is more efficient than [18].

6 Conclusion

This paper proposes a multi-resolution cloth construction method based on low-resolution cloth sampling to address fidelity and efficiency in dynamic cloth simulation. Compared to high-resolution simulation, our multi-resolution cloth improves computational efficiency while maintaining detail; compared to low-resolution cloth, it significantly enhances fidelity. The method allows flexible control over regional mesh precision through adjustable subdivision thresholds to meet varying fidelity and efficiency requirements.

However, limitations remain compared to classic dynamic adaptive algorithms [10]: (a) it is less suitable for highly complex motion patterns where computational efficiency decreases; (b) it requires manual participation in preprocessing, specifically command input during adaptive subdivision, limiting flexibility; (c) it only applies to similar motion-driven simulations—effectiveness is lost if the motion pattern during sampling differs significantly from that during simulation. Future work will focus on improving algorithm adaptability and flexibility.

References

[1] Cutler L D, Gershbein R, Wang X C, et al. An art-directed wrinkle system for CG character clothing [C]//Proc of ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New York: ACM Press, 2005: 117-125

[2] Rohmer D, Popa T, Cani M P, et al. Animation wrinkling: augmenting coarse cloth simulations with realistic-looking wrinkles [J]. ACM Transactions on Graphics, 2010, 29(6): article No. 157

[3] Zurdo J S, Brito J P, Otaduy M A. Animating wrinkles by example on non-skinned cloth [J]. IEEE Trans on Visualization and Computer Graphics, 2013, 19(1): 149-158

[4] Jing M, He B, Lvy Y. Cloth wrinkle enhancement based on the coarse mesh simulation [C]//Proc of 14th International Conference on Smart Homes and Health Telematics. Switzerland: Springer International Press, 2016: 293-301

[5] Li L, Volkov V. Cloth animation with adaptively refined meshes [C]//Proc of the 28th Australasian Computer Science Conference. Darlinghurst: Australian Computer Society Press, 2005: 107-114

[6] Simnett T J R, Laycock S D, Day A M. An edge-based approach to adaptively refining a mesh for cloth deformation [C]//Proc of the 7th Theory and Practice of Computer Graphics Conference. Geneve: Eurographics Association Press, 2009: 77-84

[7] Lee Y, Yoon S, Oh S, et al. Multi-resolution cloth simulation [J]. Computer Graphics Forum, 2010, 29(7): 2225-2232

[8] Wu J H S, Liu W J, Wang T R. Adaptive refinement scheme for subdivision surfaces based on triangular meshes [C]//Proc of the 9th International Conference on Computer Aided Design and Computer Graphics. Los Alamitos: IEEE Computer Society Press, 2005: 119-124

[9] Narain R, Samii A, O' Brien J F. Adaptive anisotropic remeshing for cloth simulation [J]. ACM Trans on Graphics, 2012, 31(6): 1-10

[10] De Oliveira S M F, Vidal C A, Cavalcante-Neto J B. Cloth simulation with triangular mesh adaptivity [C]//Proc of the 27th SIBGRAPI Conference on Graphics, Patterns and Images. Los Alamitos: IEEE Computer Society Press, 2014: 235-242

[11] Narain R, Pfaff T, O' Brien J F. Folding and crumpling adaptive sheets [J]. ACM Trans on Graphics, 2013, 32(4): 1-8

[12] Koh W, Narain R, O' Brien J F. View-dependent adaptive cloth simulation with buckling compensation [J]. IEEE Trans on Visualization and Computer Graphics, 2015, 21(10): 1138-1145

[13] White R, Crane K, Forsyth D A. Capturing and animating occluded cloth [J]. ACM Trans on Graphics, 2007, 26(3): 34

[14] Sumner R W, Popovic' J. Deformation transfer for triangle meshes [J]. ACM Trans on Graphics, 2004, 23(3): 399-405

[15] Bradley D, Popa T, Sheffer A, et al. Markerless garment capture [J]. ACM Trans on Graphics, 2008, 27(3): Article No. 99

[16] De Aguiar E, Sigal L, Treuille A, et al. Stable spaces for real-time clothing [J]. ACM Trans on Graphics, 2010, 29(4): 1-9

[17] Wang H M, Hecht F, Ramamoorthi R, et al. Example-based wrinkle synthesis for clothing animation [J]. ACM Trans on Graphics, 2010, 29(4): 1-9

[18] Shi M, Mao T L, Wang Z Q. Multi-resolution mesh cloth animation based on example data analysis [J]. Chinese Journal of Computers, 2015, 38(6): 1296-1306

[19] Li G Q, Wu Z Z, Ma W Y. Research progress in adaptive subdivision technology [J]. Journal of Computer-Aided Design & Computer Graphics, 2006, 18(12): 1789-1798

[20] Tang Y, Yang S S, Lv M Y. Improved cloth collision detection method using adaptive ellipsoidal bounding boxes [J]. Journal of Computer-Aided Design & Computer Graphics, 2013, 25(10): 1589-1596

[21] Wang W, Zhou C, Yang Y, et al. Flexible rope simulation based on improved spring-mass model [J]. Journal of Computer-Aided Design & Computer Graphics, 2015, 27(11): 2230-2236

Note: Figure translations are in progress. See original paper for figures.