

Short Communication

Procedural visualization of knitwear and woven cloth

Funda Durupınar, Uğur Güdükbay*

Department of Computer Engineering, Bilkent University, Bilkent 06800, Ankara, Turkey

Received 11 December 2006; received in revised form 22 May 2007; accepted 14 June 2007

Abstract

In this paper, a procedural method for the visualization of knitted and woven fabrics is presented. The proposed method is compatible with a mass-spring model and makes use of the regular warp–weft structure of the cloth. The visualization parameters for the loops and threads are easily mapped to the animated mass-spring model. The simulation idea underlying both knitted and woven fabrics is similar as we represent both structures in 3D. As the proposed method is simple and practical, we can achieve near real-time rendering performance with good visual quality.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Fabric rendering; Knitwear; Woven cloth; Physically-based modeling

1. Introduction

Modeling the visual appearance of cloth is as important as the simulation of its draping behavior for the sake of realism in computer graphics. There are various methods to produce fabrics from yarn, such as knitting, weaving, braiding or knotting. Particularly, knitting and weaving are the most common techniques and they have been studied in the literature.

Meissner and Eberhardt [1] introduce a system that simulates the physically correct appearance of knitted fabrics. Zhong et al. [2] propose a method for rendering knitwear on free-form surfaces. Another study that uses free-form surfaces is presented by Xu et al. [3], where photorealistic rendering of knitwear is handled by introducing an element called the *lumislice*. Later, Chen et al. [4] extend this study to include the realistic animation of knitwear. A recent study in textile research examines the dynamic draping behavior of woven and knitted fabrics [5]. In the case of woven cloth, we cannot do simple texture mapping because occlusion and self-shadowing terms come into play due to the structure of the individual threads and their interweaving pattern. In [6], woven cloth is simulated

by computing the lighting using a geometrical model of a stitch. Then, by sampling the stitch regularly within a plane, a view-dependent texture with per-pixel normals and material properties is generated. In [7], the weave of the texture is simulated by procedural displacements of the geometry and the loop is represented as a 2D curve. The 3D appearance is given by displacing the loop. A recent study that supports a wide variety of weave patterns and captures the difference in appearance of the front and back surfaces of woven cloth is presented by Adabala et al. [8]. The color texture, BRDF texture and the horizon maps are pre-computed and the rendering of the images using these textures is achieved in real time with a good visual quality.

In this paper, we propose a simple and efficient procedural method for the visualization of knitted and woven textiles. For this purpose, we use rectangular mass-spring meshes and cut out regular garment patterns composed of quadrilaterals from the meshes. Regularity both preserves the general cloth behavior such as shearing and bending, thus preserving physical accuracy and enables the definition of complex knit and weave patterns. In most of the existing systems, woven fabrics are not simulated physically; they are simplified as 2D structures. We exploit the regular structure of cloth models and parametrically define the repetitious structure of woven and knitted fabric. The weave and knit patterns can be as complex as desired and can support any color pattern. As the structure is 3D,

*Corresponding author. Tel.: +90 312 290 1386; fax: +90 312 266 4047.

E-mail addresses: fundad@cs.bilkent.edu.tr (F. Durupınar),
gudukbay@cs.bilkent.edu.tr (U. Güdükbay).

the resulting garments have different appearances for the front and back surfaces of the fabrics, depending on the weave or knit pattern as in [8]. However, our method does not require preprocessing for texture generation; all the calculations are done on-the-fly. We achieve near real-time rendering and simulation performance with good visual quality.

2. Garment visualization

In this section, the simulation of the two methods of fabric production, i.e., weaving and knitting, is introduced. The cloth model used is a mass-spring model [9]. This is a specific case of a particle system in which the particles are connected by spring forces. The type and behavior of the cloth is determined by the strength of the spring forces and the topology of the cloth, which in turn is determined by how the springs connect the particles. Three types of springs are used to reproduce the stretching, shearing and bending behavior of cloth. The mass-spring model is adopted due to its simplicity and efficiency [10]. The initial grid structure is a rectangular mesh of particles and springs connecting these particles in horizontal, vertical and diagonal directions. In addition, contrary to the irregular triangular meshes, the mesh structures of garment patterns are regular, which is compatible with the warp and weft directions of the woven cloth. Thus, instead of drawing the 2D garment pattern and discretizing it by triangulation, we use an initial rectangular mesh composed of quadrilaterals and cut out the desired shape by preserving the regularity.

2.1. Knitwear

The structure of knitwear is complicated compared to other techniques like weaving. This is due to the 3D geometry of a knit loop. In our system, we make use of the particle system and the mass-spring model of our cloth mesh in order to consider the interaction of neighboring loops. For this purpose, the cloth mesh must be a regular lattice consisting of quadrilaterals. There are two types of basic stitches when knitting: left and right loops. The knitwear pattern, which shows the order of the right and left loops, is read from an input file and can be changed interactively in the program. Each quadrilateral contains one type of loop. The structure of the loop in a quadrilateral is defined by the bonding points (BPs). The positions of the BPs can be determined parametrically using the vertices of the enclosing quadrilateral as follows:

$$BP_1 = \frac{6}{7} p_{3_{pos}} + \frac{1}{7} p_{4_{pos}},$$

$$BP_2 = \frac{5.5}{21} p_{1_{pos}} + \frac{1.5}{21} p_{2_{pos}} + \frac{11}{21} p_{3_{pos}} + \frac{3}{21} p_{4_{pos}},$$

$$BP_3 = \frac{1.5}{21} p_{1_{pos}} + \frac{5.5}{21} p_{2_{pos}} + \frac{3}{21} p_{3_{pos}} + \frac{11}{21} p_{4_{pos}},$$

$$BP_4 = \frac{1}{7} p_{3_{pos}} + \frac{6}{7} p_{4_{pos}},$$

$$BP_5 = p_{3_{pos}},$$

$$BP_6 = \frac{5}{7} p_{3_{pos}} + \frac{2}{7} p_{4_{pos}},$$

$$BP_7 = \frac{4.5}{28} p_{1_{pos}} + \frac{2.5}{28} p_{2_{pos}} + \frac{13.5}{28} p_{3_{pos}} + \frac{7.5}{28} p_{4_{pos}},$$

$$BP_8 = \frac{6}{7} p_{1_{pos}} + \frac{1}{7} p_{2_{pos}},$$

$$BP_9 = p_{4_{pos}},$$

$$BP_{10} = \frac{2}{7} p_{3_{pos}} + \frac{5}{7} p_{4_{pos}},$$

$$BP_{11} = \frac{2.5}{28} p_{1_{pos}} + \frac{4.5}{28} p_{2_{pos}} + \frac{7.5}{28} p_{3_{pos}} + \frac{13.5}{28} p_{4_{pos}},$$

$$BP_{12} = \frac{1}{7} p_{1_{pos}} + \frac{6}{7} p_{2_{pos}}, \tag{1}$$

where BP_i is the i th bonding point and $p_{j_{pos}}$ is the 3D position of vertex j .

Due to the thickness of the yarn, these BPs are moved slightly, taking the normal of that quadrilateral into consideration. Then, each bonding point BP_i is assigned the value $BP_i + c_i N$, where N is the surface normal and c_i is a constant that is defined according to the type of the loop and BP. Thus, the knitted fabric looks different when front and back views are considered. Fig. 1 shows the construction of the loops.

In order to maintain interactivity, complex volumetric models or time-consuming methods like [3,11] for rendering the yarns in a loop cannot be utilized. Three-dimensionality is achieved by drawing cylinders around each yarn and applying texture mapping on these cylinders (see Fig. 2 for a close-up view of a knitwear). Using the position information of consecutive BPs, a cylinder of height $|\vec{V}|$ lying along the positive z -axis is found. Then, the cylinder is transformed by a composite transformation

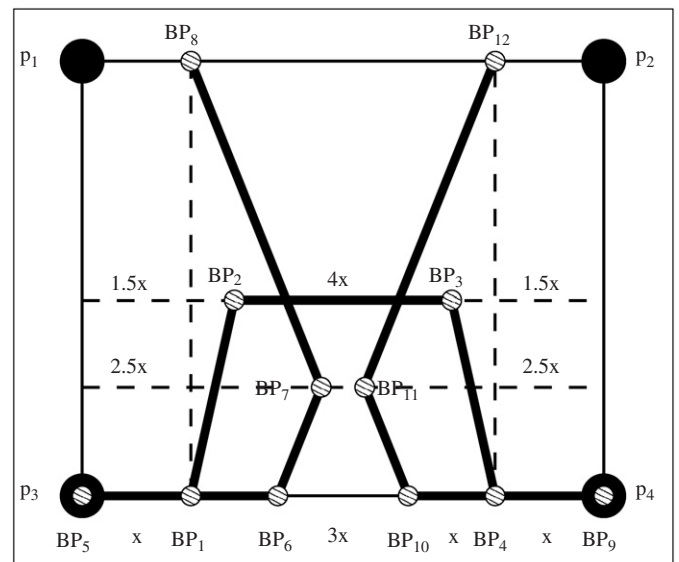


Fig. 1. Bonding points of a knit loop.

matrix so as to be oriented in the direction of the vector between the two consecutive BPs. The composite transformation matrix M includes a rotation around y -axis, a rotation around x -axis, and a translation:

$$M = T(BP_i) \cdot R_x(-\alpha) \cdot R_y(-\beta)$$

$$\ell = \sqrt{(BP_{i+1} \cdot y - BP_i \cdot y)^2 + (BP_{i+1} \cdot z - BP_i \cdot z)^2}$$

$$\alpha = \arcsin((BP_{i+1} \cdot y - BP_i \cdot y) / |\vec{V}|) / \ell$$

$$\beta = \arccos(\ell). \quad (2)$$

In order to optimize the performance, varying levels of detail from different viewing distances are shown. For distant views, since the yarn structure cannot be distinguished, instead of drawing cylinders by performing explicit transformations, we simply draw texture-mapped, antialiased line primitives between the BPs by hardware-supported line drawing. The line size, which is determined by the number of pixels, is proportional to the proximity of the viewer to the

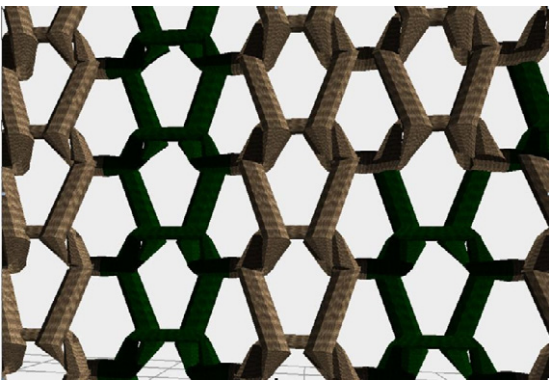


Fig. 2. The close-up view of a knitwear.

cloth. Another issue to consider is the popping artifacts for the transition between the two modes of rendering yarns. The transition should be gradual and devoid of sudden leaps. We achieve smooth transition by image blending. For intermediate distance ranges, the knitwear is generated with line primitives and with cylinders separately. These knitwear images are blended and the resulting image is rendered. The blending weights of the two images are determined according to the distance of the viewer to the cloth. Blending is handled using hardware support. The blending operation is illustrated in Fig. 3. In the most distant view, the cloth is rendered with line primitives. Then, at intermediate distance range, the blended image is rendered. The closest view is generated with cylinders. The images at the bottom row demonstrate magnified fragments from each image. The knitwear rendering algorithms are as follows:

Algorithm. RENDERKNITWEAR (*cloth*, *pattern*)

- (1) Compute viewing distance to cloth
- (2) **if** viewing distance > high
- (3) line width $\leftarrow c / \text{viewing distance}$
- (4) $\underline{I} \leftarrow \text{GENERATELINETHREADS}(\text{cloth}, \text{pattern}, \text{line width})$
- (5) **else if** viewing distance $\in [\text{low}, \text{high}]$
- (6) line width $\leftarrow c / \text{viewing distance}$
- (7) $\underline{I} \leftarrow \text{GENERATELINETHREADS}(\text{cloth}, \text{pattern}, \text{line width})$
- (8) $\underline{I}_2 \leftarrow \text{GENERATECYLINDRICALTHREADS}(\text{cloth}, \text{pattern})$
- (9) $\underline{I} \leftarrow (\underline{I}_2 * (\text{viewing distance} - \text{low}) + \underline{I}_1 * (\text{high} - \text{viewing distance})) / (\text{high} - \text{low})$
- (10) **else**
- (11) $\underline{I} \leftarrow \text{GENERATECYLINDRICALTHREADS}(\text{cloth}, \text{pattern})$
- (12) Render \underline{I}

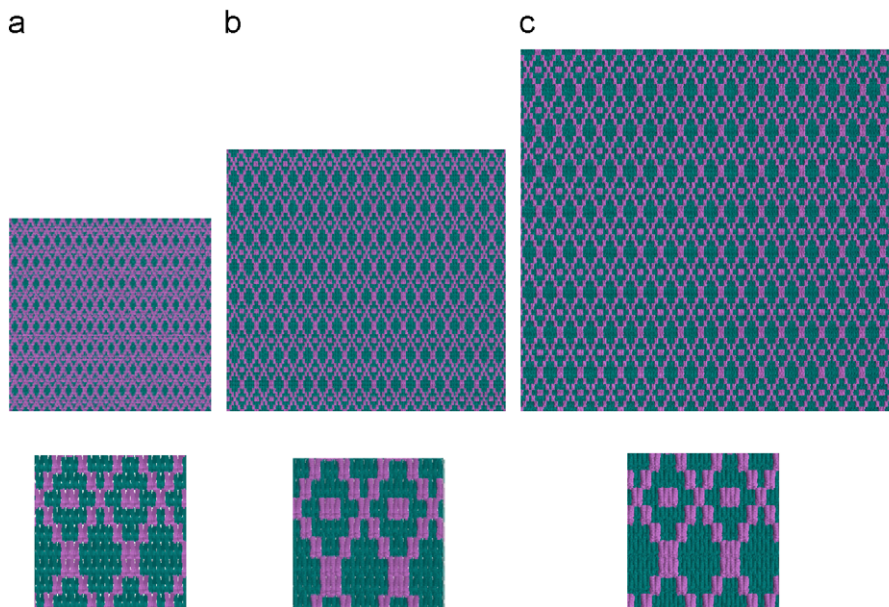


Fig. 3. Transition from line primitives to cylinders: (a) distant view with lines; (b) intermediate distance with blended image; and (c) close-up view with cylinders. Images at the bottom part show magnified views for each distance.

Algorithm. GENERATECYLINDRICALTHREADS (*cloth, pattern*)

- (1) **for each** *quadrilateral* in *cloth*
- (2) Compute bonding points *BP* using Equation 1 and according to *pattern*
- (3) **for each** $i \leftarrow 1, \dots, 11$
- (4) $\vec{V} \leftarrow BP_{i+1} - BP_i$
- (5) Find *cylinder* of height $|\vec{V}|$ lying along the positive z-axis
- (6) Align *cylinder* with the direction of \vec{V} by the transformation matrix in Equation 2
- (7) Draw *cylinder* in *image*
- (8) **return** *image*

The algorithm for GENERATELINETHREADS (*cloth, pattern, line width*) is similar to GENERATECYLINDRICALTHREADS (*cloth, pattern*) with the exception of drawing line primitives of size *line width* instead of cylinders.

Fig. 4 shows a knitted shirt from different distances, where the close-up views are rendered with cylinders and the distant view is rendered with lines. Fig. 5 shows examples of knitwear rendered with line primitives. Finally, Fig. 6 demonstrates the physical behavior of knit loops, whereby the correlation between the forces acting on the cloth and the distance between the threads can be observed.

2.2. Woven cloth

In real life, weaving can be performed by means of a loom. The idea of the loom is to interleave two sets of perpendicular threads [12]. These threads are the warp and the weft threads. Warp threads are the vertical ones, whereas weft threads are the horizontal ones. Weaving patterns can be obtained from two types of interleaving of threads, i.e., *warp on weft* and *weft on warp*. Various weaving patterns can be created by ordering these two



Fig. 4. Appearance of a knitted shirt from different distances. Close-up view is rendered with cylinders and distant view is rendered with lines.



Fig. 5. Knitwear garments rendered with line primitives.



Fig. 6. The physical behavior of knitwear.

thread interleaving types. Our system works as follows: (i) we read the patterns from a pattern description file, (ii) draw them in 3D as warp and weft threads by calculating their positions and (iii) do texture mapping on them to capture the detailed appearance of the component fibers. The texture is simply an image of a thread surface, which depicts the twists of the yarn clearly. Texture coordinates are calculated after the weave pattern is read from the file. The 3D structure of each weave unit is shown in Fig. 7.

We render the threads one by one at each iteration. In this way, our method for rendering woven cloth is similar to the rendering of knitwear since the 3D structure is not ignored. The indices of each of the fibers are calculated and they are moved in the direction normal to the quadrilateral they are on. Since the weaving structure is defined procedurally, there is no need to use alpha values for the gaps between the warp and the weft threads. When the weave pattern is read from the file, each quadrilateral in the cloth mesh is subdivided into smaller quadrilaterals in

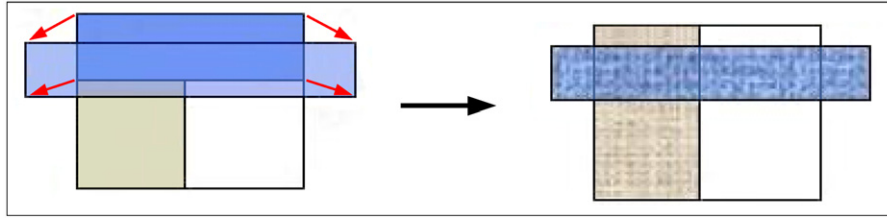


Fig. 7. 3D structure of a weave unit.

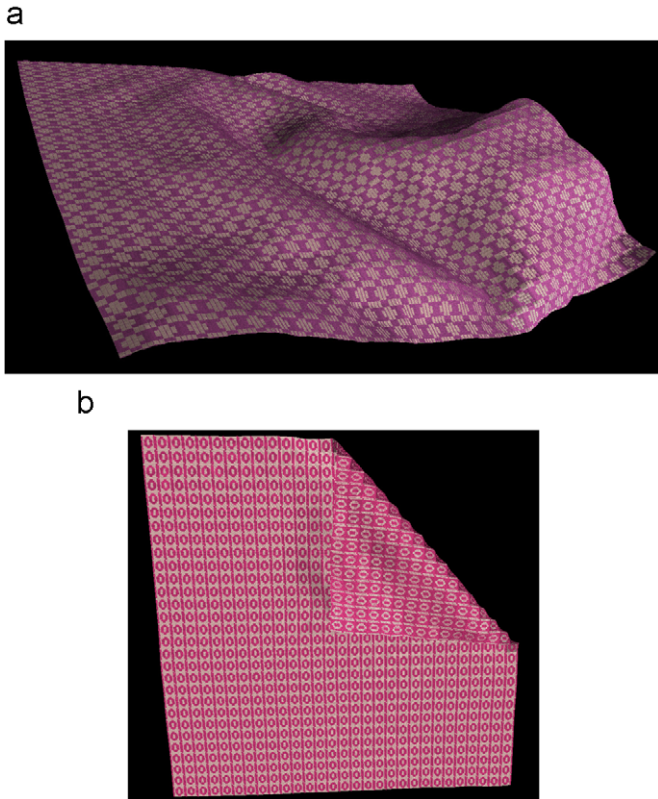


Fig. 8. Examples of woven cloth. The woven cloth in (b) shows the difference in the appearance of front and back surfaces.

order to construct the pattern. The input file consists of a matrix of binary numbers, where 0 denotes weft on warp and 1 denotes warp on weft. Fig. 8 exhibits examples of woven cloth. The woven cloth in Fig. 8(b) shows the difference in the appearance of front and back surfaces. Fig. 9 shows a woven shirt with different thread sizes and from different distances. Transparent cloth can be obtained by simply reducing the thickness of the threads.

3. Results and conclusion

The implementation is tested on a PC (T2500, 2.00 GHz) with 2 GB of RAM. The graphics card is NVIDIA GeForce Go 7400 with 256 MB memory size. The software platform is Microsoft Visual C++ 6.0 with OpenGL libraries. The average frame rates for rendering a cloth with

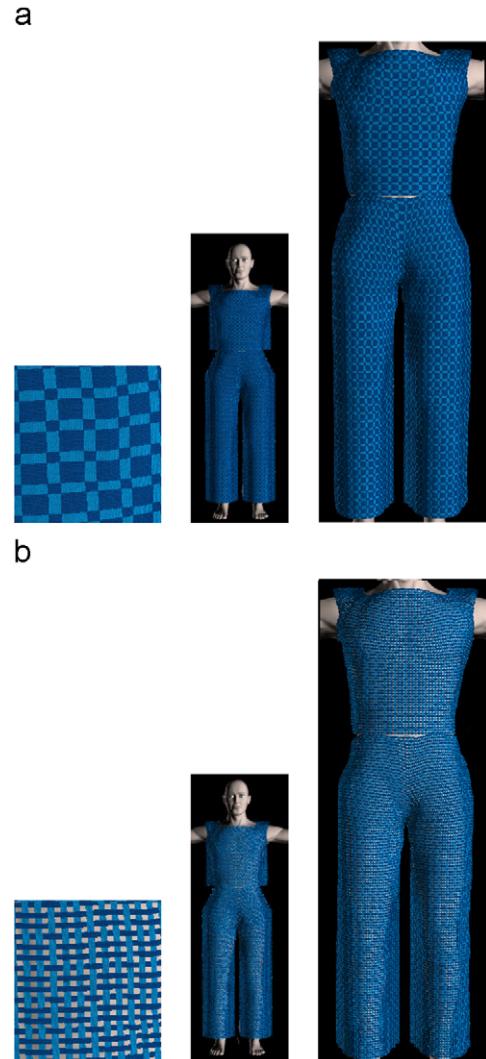


Fig. 9. (a) Woven outfit with (a) thick and (b) thin threads for magnified, distant and close-up views.

5000 loops (threads) are 625 frames per second (fps) for texture mapping, 42.73 fps for knitwear with line threads, 11.22 fps for knitwear with cylindrical yarns, 8.3 fps for blended knitwear, and 70.42 fps for woven cloth. When the same cloth patch is animated including the gravitational force and self-collisions, the average frame rates (including physical simulation and rendering) are 35.46 fps for texture mapping, 11.02 fps for simple knitwear, 2.05 fps for knitwear with cylindrical yarns, 1.72 fps for blended knitwear

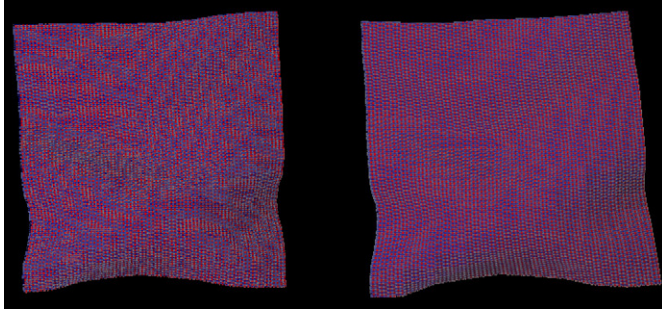


Fig. 10. Clothes before and after antialiasing.

and 12.84 fps for woven cloth. The higher frame rates for texture-mapped cloth is due to handling texture mapping on GPU. The difference between the rendering times of knitwear and woven cloth is due to the complexity of the knitwear's threading structure. The correspondence of the mass-spring system with the warp/weft structure of the cloth is more suitable for woven cloth rather than knitwear.

The frame rates for the blended images are lower compared to the simple and cylindrical knitwear rendering as two images are generated for blending. However, as blending is only applied for a short-distance range, this does not bring much performance overhead on the overall.

Due to the thin rendering primitives, some aliasing artifacts occur for distant views. This is prevented by applying scene antialiasing using the accumulation buffer mechanism. The images before and after antialiasing can be seen in Fig. 10.

Owing to the simplicity of the method, the rendering performance is mostly real time. As a future work, approaches for real-time animation will also be investigated. In addition, currently, knitting is only applicable to simple stitching structures, i.e., left and right loops; thus, further study on complex stitching techniques with a procedural method can be a future research direction.

Acknowledgments

This work is supported in part by EC 6th Framework Program under Grant no. FP6-511568 (3DTV Network of Excellence Project) and the Scientific and Technical Research Council of Turkey (TUBITAK) under Grant no. EEEAG-105E065.

References

- [1] Meissner M, Eberhardt B. The art of knitted fabrics: realistic and physically based modelling of knitted fabrics. *Computer Graphics Forum (Proceedings of Eurographics'98)* 1998;17(3):355–62.
- [2] Zhong H, Xu Y, Guo B, Shum H. Realistic and efficient rendering of free-form knitwear. *Journal of Visualization and Computer Animation* 2001;12(1):13–22.
- [3] Xu Y, Lin Y, Zhong CS, Wu E, Shum H. Photorealistic rendering of knitwear using the lumislice. *ACM Computer Graphics (Proceedings of SIGGRAPH'01)*, 2001. p. 391–8.
- [4] Chen Y, Lin S, Zhong H, Xu Y-Q, Guo B, Shum H-Y. Realistic rendering and animation of knitwear. *IEEE Transactions on Visualization and Computer Graphics* 2003;9(1):43–55.
- [5] Ji F, Li R, Qiu Y. Simulate the dynamic draping behavior of woven and knitted fabrics. *Journal of Industrial Textiles* 2006;35(3):201–15.
- [6] Daubert K, Lensch H, Heidrich W, Seidel HP. Efficient cloth modeling and rendering. In: *Eurographics rendering workshop'01*, 2000. p. 63–70.
- [7] Drago F, Chiba N. Procedural simulation of interwoven structures. In: Vince J, Earnshaw R, editors, *Advances in modelling, animation and rendering*. Berlin: Springer; 2002. p. 123–38.
- [8] Adabala N, Magnenat-Thalmann N, Fei G. Real-time rendering of woven clothes. In: *Proceedings of ACM virtual reality software and technology*, 2003. p. 41–7.
- [9] Provot X. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In: *Proceedings of the graphics interface (GI'95)*, 1995. p. 141–55.
- [10] Witkin A, Baraff D. Physically-based modeling. *ACM SIGGRAPH Course Notes*, no. #25, 2001.
- [11] Meissner M, Eberhardt B, Strasser W. Knit fabrics. In: House D, Breen D, editors, *Cloth modeling and animation*. A.K. Peters, 2000. p. 123–43.
- [12] Glassner A. Digital weaving, part 1. *IEEE Computer Graphics and Applications* 2002; 22(6):108–18.