A Virtual Garment Design and Simulation System

Funda Durupınar, Uğur Güdükbay
Department of Computer Engineering, Bilkent University, Ankara, Turkey
fundad@cs.bilkent.edu.tr, gudukbay@cs.bilkent.edu.tr

Abstract

In this paper, a 3D graphics environment for virtual garment design and simulation is presented. The proposed system enables the three dimensional construction of a garment from its cloth panels, for which the underlying structure is a mass-spring model. The garment construction process is performed through automatic pattern generation, posterior correction, and seaming. Afterwards, it is possible to do fitting on virtual mannequins as if in a real life tailor's workshop. The system provides the users with the flexibility to design their own garment patterns and make changes on the garment even after the dressing of the model. Furthermore, rendering alternatives for the visualization of knitted and woven fabric are presented.

Keywords— **garment design; garment simulation; physically-based modeling**

1 Introduction

Cloth simulation has been a challenging issue in computer graphics for a long time. Today, there is an increasing demand for the involvement of computer graphics in the textile and entertainment industries. A further step in cloth simulation is the simulation of garments, which are basically assemblies of different cloth parts. Recently, there has been a great deal of research on 3D garment design [1, 2, 3, 4, 5, 6, 7, 8, 9].

In this paper, we present an integrated framework that provides all aspects of a 3D garment design and simulation system, including human motion and animation, garment design, cutting, sewing and realistic visualization of textiles. We propose an automatic 2D garment pattern construction algorithm that uses only integer arithmetic. The algorithm is inspired by Bresenham's line drawing algorithm [10]. The method considers the measurements of the human model and automatically generates the garment patterns that are specific to a garment type. The algorithm finds the corner vertices of the pattern boundary and computes the shape of the pattern from these vertices. In most of the developed software tools, the 2D garment patterns are obtained from existing textile design tools and they are then triangulated and converted into physically-based models. In contrast to these systems, we create garment patterns from 3D cloth meshes by cutting the initial rectangular mesh. Our main motivation underlying this preference is to create regular mesh structures, which preserve general cloth behavior such as shearing and bending. This technique both simplifies pattern construction and preserves regularity of the cloth, thus complying with the mechanical properties and the anisotropic behavior of fabric.

Furthermore, we introduce a procedural method for the visualization of knitted and woven textiles that makes use of the regular structure of the fabric. With the proposed technique, it is possible to get physical accuracy in simulation, to define complex knit and weave patterns, and to obtain different appearances for the front and back surfaces of woven or knitted fabrics. In most of the existing systems, woven fabrics are not simulated physically; they are simplified as 2D structures. Moreover, knitwear is usually represented as free-form surfaces, ignoring the physical properties of the fabric. In our system, we exploit the regular structure of the cloth models and parametrically define the repetitious structure of woven and knitted fabric.

Our system is built on a human modeling and animation tool [11]. The human body motion module allows the user to define various human motion behaviors by adjusting the position, distance and rotation curves of the joint points. The garment design module is an effective tool for creating 3D garments from their cloth panels through cutting and seaming. It enables the designer to position the 3D garment around the virtual character and sew the garment panels. In addition, the system provides different options for rendering garments. The simulation module allows for human motion, cloth deformation and collision handling to be considered simultaneously. The system architecture is given in Figure 1.

The organization of the paper is as follows: Section 2 explains the garment design process, Section 3 details the garment simulation operations and Section 4 overviews the garment visualization methods used in the system. Finally, Section 5 presents performance analysis and future work.

11th International Conference Information Visualization (IV'07)
0-7695-2900-3/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

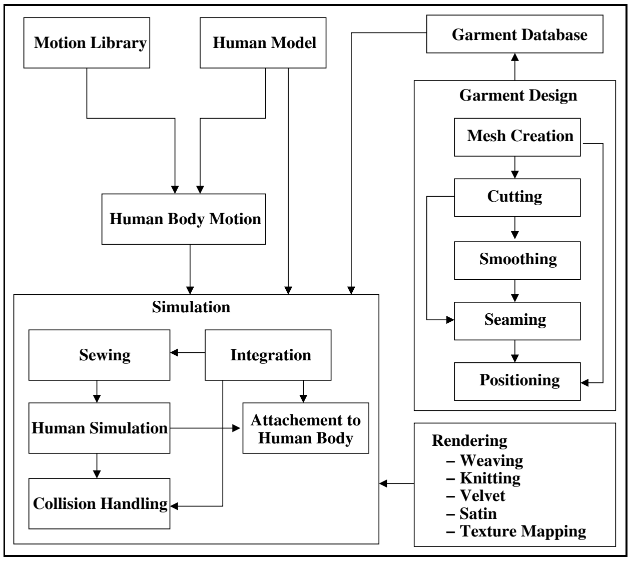


Figure 1: The system architecture.

2 Garment Design

Garment design consists of defining the cloth model and the garment construction process, in turn of the construction of garment patterns and the definition of the seaming points between each panel.

The cloth model used in our system is a mass-spring model [12]. 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. Structural springs connect the vertices adjacent along the row or column of the grid. Shear springs connect the diagonal vertices. Bend springs connect every other particle along the two directions of the rectangular grid. The mass-spring model is adopted due to its simplicity, efficiency and capacity to simulate the physical behavior of cloth. The initial grid structure is a rectangular mesh of particles at the vertices and springs connecting these particles in horizontal, vertical and diagonal directions. The regular mesh structure in our system is compatible with the warp and weft directions of woven fabrics.

Garment pattern construction is a tedious and time-consuming task. In order to reduce the effort spent in this task, we propose an automated process that considers the measurements of the human model. The process also involves some minor human intervention to make some final touches. Our model is compliant with the H-anim 1.1 specification, which enables the classification of the articulated human figure into several body segments such as knee, shoulder or elbow. Garment creation in real life requires the calculation of some body measurements such as shoulder length, arm length and bust girth. We compute these values considering the body segments of the model. We store information related to different garment types in our database. For instance, creation of a shirt requires shoulder length, bust, waist and arm girths, and bust height. Thus, the corner vertices of the boundary of the garment panel are defined according to the body measurements. From these vertices, the desired shape is obtained on the cloth mesh. The algorithm for achieving the final pattern is as follows:

1. Let $V = v_1,\ v_2,\ \ldots,\ v_n$ be the corner vertices of the boundary of the cloth panel and let the initial boundary set be $B = v_1,\ v_2,\ \ldots,\ v_n$.
2. Find the centroid c of the set V as: $c = \frac{1}{n}\sum_{i=1}^{n} v_i$.
3. Display c and if c is not inside the desired pattern, let the user select another point c' inside the panel
4. Sort V around c in counter-clockwise order. Let $W = w_1,\ w_2,\ \ldots,\ w_n$ be the new sorted set of vertices.
5. For i = 1 to n, apply Bresenham's line drawing algorithm to find the vertices comprising the edge $(w_i, w_{(i+1)\ mod\ n})$. Add each found vertex w_j that belongs to the edge $(w_i, w_{(i+1)\ mod\ n})$ to the set B. Thus, the boundary set B of the pattern is obtained.
6. Apply the 4-neighbor boundary fill algorithm in the region defined by B.
7. If necessary, refine the result manually:
   (a) To prevent jagged borders and obtain a smooth appearance, the user can select the vertices that need to be corrected. The selected vertices are defined as control points and the corresponding Bézier spline is generated. The vertices of the generated curve replace the previous
   (b) To make corrections such as creating holes, the user can select some parts of the cloth that are then extracted.

If the garment panel has a complicated shape, which is very rare, the points on the boundary of the panel can be taken from the user in sorted order. Figure 2 shows the step-by-step construction of a shirt pattern.

The panels are placed around the virtual mannequin by keyboard and mouse interaction. 3D interaction of the user to move, resize, and reshape the garment panels and the garment itself around the mannequin is possible. In addition, local parts of a garment panel can be moved or scaled

11th International Conference Information Visualization (IV'07)
0-7695-2900-3/07 $20.00 © 2007 IEEE
IEEE COMPUTER SOCIETY

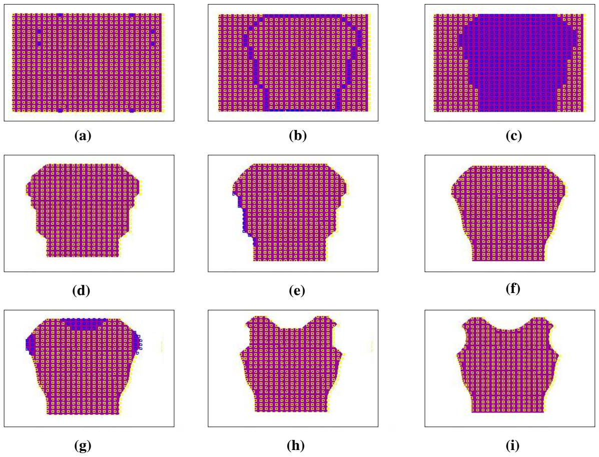


Figure 2: Step-by-step construction of a shirt pattern: (a) defining corner vertices; (b) the application of the Bresenham's line drawing algorithm to fill the boundary vertices; (c) the application of the boundary-fill algorithm; (d) cutting the boundary; (e) defining the smoothing boundary; (f) smoothing (g) defining extraction boundaries for the collar and arms (h) extracting the collar and arms; (i) smoothing.

individually. This is achieved by changing the position of the selected particles. In this way, the garment panels and the garment itself attain the desired shape.

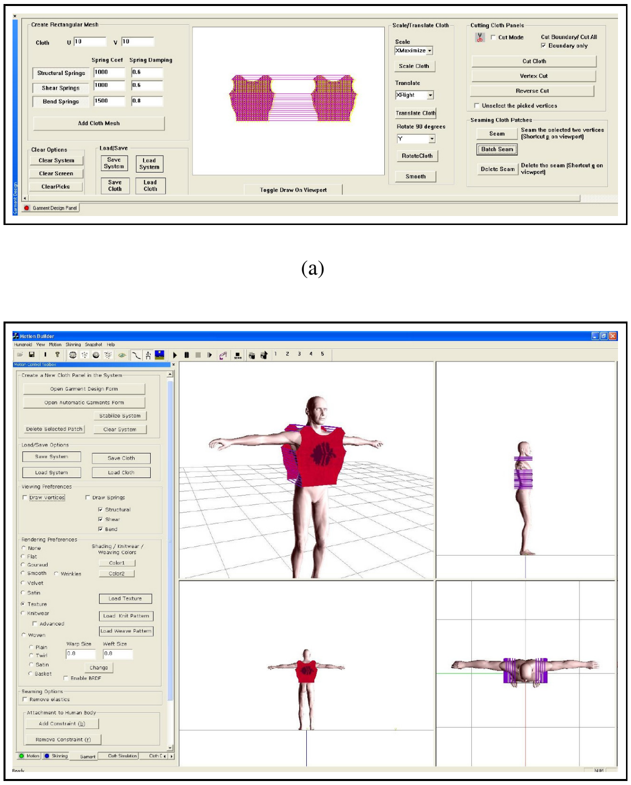
Seaming is performed by defining the seam points between the pairs of vertices on cloth panels. In our method, the line of seaming on one cloth panel is first selected manually. The algorithm then finds the closest vertices on the second cloth panel for each seam vertex on the first panel and adds seams between them. The seaming process can be seen in Figure 3.

3 Garment Simulation

To simulate the garment, the system must be integrated numerically. The positions of the particles making up the cloth are calculated over time considering the forces acting on the cloth. Several integrator types, such as Explicit Euler or adaptive 5th order Runge-Kutta, or Implicit Euler are offered to the user. The forces that act on cloth parts can be classified into internal and external forces. Internal forces are the spring forces. External forces, which determine the interaction of cloth with the environment include the gravity and collision forces.

3.1 Sewing

After the garment panels are in their accurate positions around the body, sewing is invoked by applying forces between the seams of garment parts. Seams can be regarded as forces attracting two particles to each other. In that sense, they can be considered elastic forces. However, simulating the exact behavior of elastic forces is expensive and a much simpler heuristic can solve the problem more efficiently. The heuristic approach involves applying symmetrical forces on the two particles so that they pull each other as in Equation 1.



Cloth U 10 V 10

	Spring Coef	Spring Damping
Structural Springs	1000	0.6
Shear Springs	1000	0.6
Bend Springs	1500	0.8

Scale XMaximize

Translate XRight

Rotate 90 degrees Y

□ Cut Mode

Cut Boundary/ Cut All ☑ Boundary only

□ Unselect the picked vertices

Seam the selected two vertices (Shortcut p on viewport)

Delete the seam (Shortcut g on viewport)

(a)

(b)

Figure 3: Seaming garment panels: (a) 2D view; (b) 3D view on a human model.

$$p1_{vel} = c_{attraction} \frac{|p1_{pos} - p2_{pos}|}{\| p1_{pos} - p2_{pos} \|}, p2_{vel} = -p1_{vel} \quad (1)$$

The two particles attract each other until they are constrained by collision forces. During the sewing process, no other forces such as gravity are applied on the clothes. After two particles p_1 and p_2 are closer than a threshold, the sewing process is finalized and these particles are combined into one. This is performed by adding spring forces between p_2 and the neighbors of p_1 and between p_1 and neighbors of p_2. The neighbor of a particle p means the particle q such that there exists a spring between p and q. By this approach, the garment can be a complex assembly of different textile materials.

3.2 Attachment Constraints

In order to keep the garment on the virtual model without losing efficiency, some parts of the clothes can be attached to the human body. This approach is followed depending on the type of the garment. For instance, tight clothes can be bound to the human body with attachment constraints. After the virtual human is dressed in the garment, the selected particles are attached to the closest polygon on the virtual human. In this way, the selected parts of the garment move with the human.

3.3 Collision Handling

We need to perform collision handling between garments and the skin layer of the human body in order to obtain realistic simulation results. Collision handling consists of two phases: *collision detection* and *collision response*.

The collision detection algorithm tests all the intersections between the bounding volumes of the human model against the Axis Aligned Bounding Box (AABB) hierarchy of the cloth model [13]. The AABB hierarchy is traversed until the leaf nodes are reached. If an intersection between the two bounding volumes is found (the bounding volume of the human model and the bounding volume in the leaf node of cloth model's hierarchy), then geometrical collision detection methods are applied for testing collisions between triangles. The same method is used for detecting self-collisions of clothes. Collision detection between two polygons is done by checking the point-triangle intersection and proximity, similar to the cloth-human body collision detection. In order to avoid edge-edge collision detection, we enlarge the bounding volumes of the cloth model. In addition, adjacent polygons are tested for self-collision detection by using the polygonal adjacency information.

We use velocity and position correction after detecting a collision. Velocity correction is similar to [14]. The velocity of the particle after collision response is modified as in Equation 2:

$$V_{res} = C_{fric}V_t - C_{refl}V_n + V_{human}, \tag{2}$$

where C_{fric} and C_{refl} are friction and reflection coefficients, V_{human} is the velocity of the human body, V_t and V_n are the tangent and normal components of the relative velocity of the cloth particle with respect to the human body. The relative velocity of the cloth particle is found by subtracting V_{human} from the velocity of the particle.

Position correction is applied if the point has passed beneath the polygon or if the point is at a distance from the surface closer than a threshold. In the first case, the point is brought to the surface as in Equation 3:

$$P = \Pi + N_{polygon}, \tag{3}$$

where P is the final position, Π is the projection of the particle on the triangle and $N_{polygon}$ is the normal vector of the collision plane. In the second case, the point's position is corrected by Equation 4:

$$P = P + N_{polygon}. \tag{4}$$

Implementation details of collision handling within the system are given in [15].

4 Garment Visualization

Realistic rendering of clothes is as important as the simulation of their draping behavior since important information about the material the fabric is made of can be obtained via its visual appearance. In addition to the general rendering techniques such as Gouraud shading, there are some shading techniques specifically related to textiles. There are various methods of producing garments from yarn, such as knitting, weaving, braiding or knotting. The most important ones among these are knitting and weaving. Thus we have simulated these two methods in our system. Moreover standard methods and material-specific BRDFs are also implemented.

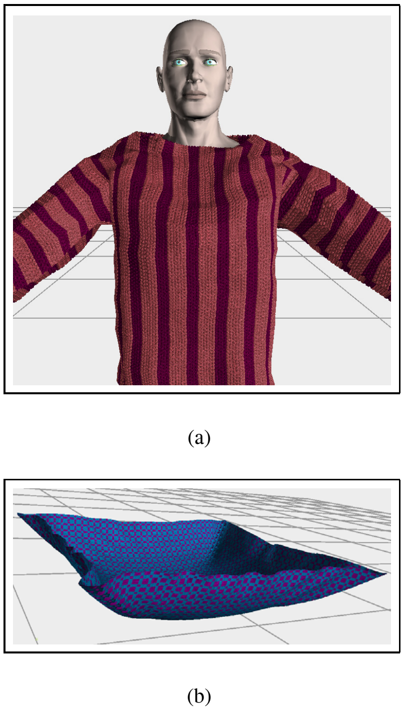
4.1 Knitwear

The structure of knitwear is complicated compared to other techniques like weaving. This is due to the three dimensional geometry of a knit loop. In our system, we make use of the particle system and the mass-spring model of the cloth mesh in order to consider the interaction of neighboring loops. For this purpose, the cloth mesh must consist of quadrilaterals and must be regular. 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 system.

Each quadrilateral of the cloth mesh contains one type of loop. The structure of the loop in a quadrilateral is defined by the bonding points (BPs). The position of the bonding points can be determined parametrically by the vertices of the enclosing quadrilateral. Due to the thickness of the yarn, these bonding points are moved slightly taking the normal of that quadrilateral into consideration. Then, each bonding point BP_i is assigned the value $BP_i + N$, where N is the surface normal. Thus the knitted fabric looks different when front and back views are considered. In order to render the 3D structure of yarns, texture-mapped cylinders are drawn between successive bonding points. Figure 4 (a) shows an example of knitwear.

4.2 Woven Cloth

In actual life, weaving can be performed by means of a loom. The idea of the loom is to interleave two sets of per-


(a)

(b)

Figure 4: (a) A knitted shirt; (b) A woven cloth.

pendicular threads, i.e. warp and weft threads [16]. 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 thread interleaving types. Our system works as follows: (i) we read the patterns from a pattern description file, (ii) draw them in three dimensions as warp and weft threads by calculating their positions and (iii) do texture-mapping on them in order to capture the detailed appearance of the component fibers. Texture coordinates are calculated once the weave pattern is read.

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 three-dimensional structure is not ignored. The indices of each of the fibers are calculated and they are moved in the direction of the normal of 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 weft threads. Transparent cloth can be obtained by simply reducing the thickness of the threads. When the weave pattern is read, each quadrilateral in the cloth mesh is subdivided into smaller quadrilaterals to construct the pattern. Figure 4 (b) shows a woven cloth with different front and back views.

4.3 Velvet and Satin

The material type for the fabric and the anisotropic behavior of cloth can be revealed by an explicit sophisticated illumination model. Anisotropic BRDF (Bidirectional Reflectance Distribution Function) is used to model the scattering of light from individual fibers in the threads. Milliscale illumination is obtained from microscale BRDF that is calculated for each of the warp and weft threads.

To simulate different fabric types, such as velvet and satin, Ashikmin's BRDF Model [17] is utilized in the system. Satin is represented as fibers running in one direction with about 70% of the fiber length being relatively flat while the other 30% at the ends is bent. Thus, the distribution of microfacets is modeled as a linear combination of these two terms of the cylindrical fiber. The main character of the distribution of microfacets in velvet is represented by an "inverse Gaussian" probability distribution.

5 Performance Analysis and Future Work

The system is run on a PC (Pentium IV, 3.01 GHz) with 1 GB of RAM. The graphics card is NVIDIA GeForce PCX 5750 with 256 MB memory size. The software platform is Microsoft Visual C++ 6.0 with OpenGL libraries. The timestep for each iteration is taken as 0.05 seconds. The human model has 17,953 vertices and 33,234 faces. The whole body is composed of 53 parts like hip, abdomen, head, right leg, left leg, etc. The frame rates per second for an animation sequence of a naked model, a model with a texture mapped outfit of 1700 vertices, a model with a woven outfit, and a model with a knitted outfit are 18.020, 1.574, 1.050, 1.570, respectively. The animation of a walking dressed human is given in Figure 5. Figure 6 shows the sewing times of a skirt with 1,400 vertices and 61 seams and the sewing times for trousers with 2,774 vertices and 140 seams. The snapshots of the sewing processes can be seen in Figure 7. As can be seen from these tables, sewing time per iteration increases as the clothes approach the human body; this is due to the calculations for collisions.

Currently the garment panel construction process requires some minor final corrections. As a future work, the whole garment design process can be automated by automatically defining such things as seaming points, positions of the panels in 3D space.

Acknowledgement

This work is supported by European Union 6th Framework Program under Grant No. FP6-511568 (3DTV NoE Project) and The Scientific and Technical Research Council of Turkey (TÜBİTAK) under Grant No. EEEAG-105E065.

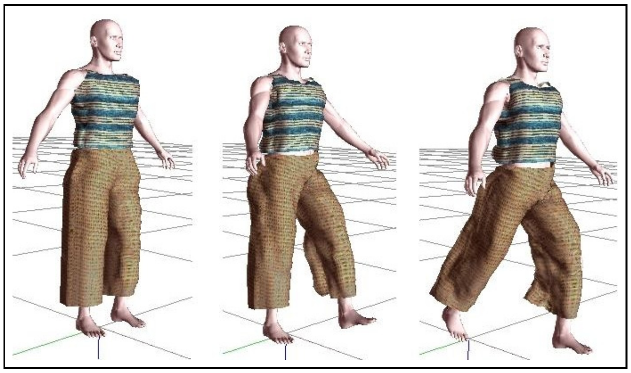


Figure 5: A dressed walking human model.

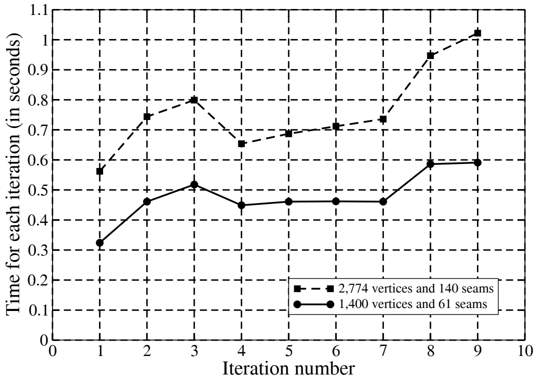


Figure 6: Sewing times. The total sewing time for a skirt with 1,400 vertices and 61 seams is 4.313 seconds and for trousers with 2,774 vertices and 140 seams, it is 6.863 seconds.

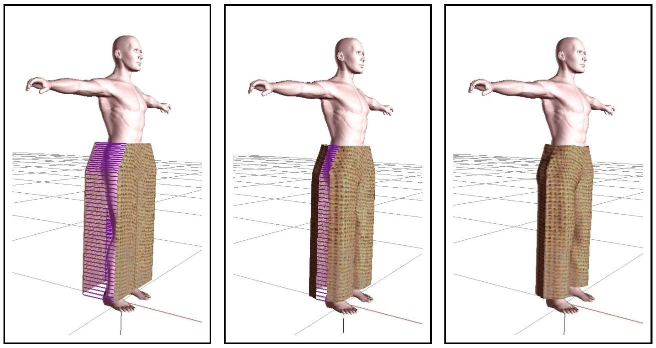


Figure 7: Sewing trousers of 2,774 vertices and 140 seams.

References

[1] P. Volino and N. Magnenat-Thalmann, *Virtual Clothing: Theory and Practice*. Springer-Verlag, 2000.

[2] Y. Yang, N. Magnenat-Thalmann, and D. Thalmann, "3D garment design and animation – a new design tool for the garment industry," *Computers in Industry*, vol. 19, pp. 185–191, 1992.

[3] F. Cordier, H. Seo, and N. Magnenat-Thalmann, "Made-to-measure technologies for an online clothing store," *IEEE Computer Graph. and App.*, vol. 23, no. 1, pp. 38–48, 2003.

[4] C. Gross, A. Fuhrmann, V. Luckas, and J. Encarnação, "Virtual try-on: Topics in realistic, individualized dressing in virtual reality," in *Proc. of the Virtual and Augmented Reality Status Conf.*, 2004.

[5] L. Chittaro and D. Corvaglia, "3D virtual clothing: from garment design to Web3D visualization and simulation," in *Proc. of the 8th Int. Conf. on 3D Web Technology*, 2003, pp. 73–85.

[6] A. Fuhrmann, C. Gross, V. Luckas, and A. Weber, "Interaction-free dressing of virtual humans," *Computers & Graphics*, vol. 27, pp. 71–82, 2003.

[7] C. Wang, Y. Wang, and M. Yuen, "Feature based 3D garment design through 2D sketches," *Computer-Aided Design*, vol. 35, no. 7, pp. 659–672, 2003.

[8] P. Decaudin, D. Julius, J. Wither, L. Boissieux, A. Sheffer, and M.-P. Cani, "Virtual garments: A fully geometric approach for clothing design," *Computer Graphics Forum (Proc. of Eurographics'06)*, vol. 25, no. 3, sep 2006.

[9] E. Turquin, J. Wither, L. Boissieux, M.-P. Cani, and J. Hughes, "Made-to-measure technologies for an online clothing store," *IEEE Computer Graphics and Applications*, vol. 27, no. 1, pp. 72–81, 2007.

[10] J. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems J.*, vol. 4, no. 1, pp. 25–30, 1965.

[11] U. Güdükbay, B. Özgüç, A. Memisoğlu, and M. Yeşil, "Realistic modeling, animation, and rendering of humans," in *Three-Dimensional Television: Capture, Transmission, Display*. Springer Verlag, to appear.

[12] X. Provot, "Deformation constraints in a mass-spring model to describe rigid cloth behaviour," in *Proc. of Graphics Interface (GI'95)*, 1995, pp. 141–155.

[13] G. Van Den Bergen, "Efficient collision detection of complex deformable models using AABB trees," *Journal of Graphics Tools*, vol. 2, no. 4, pp. 1–13, 1997.

[14] T. Vassilev and B. Spanlang, "Fast cloth animation on walking avatars," *Computer Graphics Forum (Proc. of Eurographics'01)*, vol. 20, no. 3, pp. 260–267, 2001.

[15] I. Kaynar, "Animating dressed virtual humans," Master's thesis, Dept. of Comp. Eng., Bilkent Univ., 2004.

[16] A. Glassner, "Digital weaving, part 1," *IEEE Computer Grap. and App.*, vol. 22, no. 6, pp. 108–118, 2002.

[17] M. Ashikmin, S. Premoze, and P. Shirley, "A microfacet-based BRDF generator," *ACM Computer Graphics (Proc. of SIGGRAPH'00)*, pp. 65–74, 2000.