Three-dimensional Garment Dressing Simulation

Abstract   The garment dressing simulation is a process which transforms multiple two-dimensional garment patterns into a three-dimensional configuration that follows the surface of the human body and enables visualization of the draping effect of the garment. The dimensional stability of a garment pattern is an important constraint that needs to be imposed to ensure the created garment has a size complying with the original design of the pattern. In this paper, we used a mass-spring system to simulate the physical properties of the fabric and adaptive sewing forces to wrap the pattern around the human model to form a virtual garment. Combined with the collision detection and response routine, the strain control for maintaining the pattern size was implemented with velocity adjustments.

Key words   collision detection, dressing simulation, initial positioning, strain control

Yueqi Zhong and Bugao Xu[1]
Department of Human Ecology, The University of Texas at Austin, Austin, TX, USA

The three-dimensional (3D) garment dressing simulation is a process to create a virtual garment onto a digital human model to verify the style and fit of a designed pattern before the garment is actually made. A robust simulation system should include the key functions that perform fabric modeling, sewing scheme, accurate and fast collision detection/response, and size stability maintenance.

Of the various methods of fabric modeling [1–9], the mass-spring model seems to be the widely adopted way to simulate the draping properties of a fabric [7]. In such a system, a sheet of fabric is discretized into mass particles networked by springs, and the deformation of the fabric is visualized through the movement of each particle that is governed by various external and internal forces. If the mass of the particle and the stiffness and viscosity of the spring are selected properly, this model can demonstrate cloth-like deformations for different types of fabrics with specific fiber contents and structures.

However, a mass-spring system is likely to produce the "super-elastic" effect [7]. Since the fit assessment of a virtual garment on a particular model is one of the important applications of the virtual dressing, the strain (elongation/ compression rate) of the meshes should not exceed a given tolerance. Caramana et al. [10] specified that any edge of a triangular mesh should not change by more than 10 % in a single time step of the integration. To rectify the super-elastic behavior, two algorithms were developed based on the position adjustment [7] or velocity adjustment of the end-points of the spring that suffers the over elongation/ compression [11]. Though the position adjustment algorithm can effectively restrain the strain within the limit, its possibility to introduce extra intersections between cloth mesh and subject surface and self-intersection among cloth meshes makes it unsuitable for strain control in the virtual dressing procedure, where unpredicted penetration may introduce unacceptable visual effects. Since the velocity adjustment scheme can be integrated into the framework of collision processing routine, it is more reliable in maintaining the real-time size stability.

[1] Corresponding author: Department of Human Ecology, The University of Texas at Austin, Austin TX, USA. Tel: (512) 471-7226; fax: (512) 471-5630; e-mail: bxu@mail.utexas.edu

The sewing scheme is another important issue in the virtual dressing simulation. Volino and Magnenat-Thalmann [12] demonstrated that both elastics and sewing forces could bring objects together. In their later work, an acceleration correction scheme was introduced to prevent two matching nodes from swinging around each other when their motions do not converge to the same point.

In this paper, we report the progress made in developing practical and efficient dressing simulation techniques. After the mass-spring mesh models of the pattern pieces for a garment are created [13], the dressing simulation can be divided into three stages: initial positioning, wrapping, and draping. In the first stage, an optimal initial position for each "to be sewn" panel is determined based on the body shape and posture. The appropriate settings of the panels around the body can greatly reduce the computing time in the following stage. In the wrapping stage, the gravity force on the particles of the panel mesh is temporarily removed, and only the attractive force on each particle is imposed to wrap the panels towards the body surface. The force is adaptive to the distance between the particle and the body at each time step of the integration. The particles on the seamlines of two matching panels (front and back) are pulled together to form a 3D garment. In the last stage, the gravity and surface frictions are added so that the draping effect of a virtual garment can be generated. Since the dressing simulation is an intricate process, this paper focuses on only a few important issues, such as initial position, adaptive sewing force, collision detection and response, and strain control.

Initial Positioning and Rolling of Panels

To wrap pattern panels onto a digital human model, the panel should be positioned in accordance with the pose of the model, which can be defined by the skeleton structure of the human model. The skeleton structure is constructed through the connections of several joints that follow the principle of anatomical landmarking, as shown in Figure 1. The detailed information for acquiring a skeleton from a human model has been reported in a previous paper [14].

Once a panel is placed in front of the relevant body part, a bounding box is formed and the axis of the bounding box is regarded as the axis of the panel. To position the panel around a body part of the model, an affine transformation matrix is computed to transform the main axis (either in the transversal direction or in the longitudinal direction) of the bounding box into the orientation of the two joint connections that the panel is laid along. Suppose that the position vector of a mass particle before the transformation is $\mathbf{P}_0$, and the affine transformation matrix of the main axis is $\mathbf{M}$. Hence, the corresponding position $\mathbf{P}$ of the particle on the transformed panel will be:

$$\mathbf{P} = \mathbf{M} \cdot \mathbf{P}_0.$$

Sometimes, a panel faces several skeleton joints, and the joint connections may not be a straight line. In this case, the main axis of the bound box will target the average orientation of the multiple joint connections when computing the affine transformation matrix.

The initial configuration of a panel in the 3D space is a flat panel. Before the wrapping, some patterns can be tube-shaped by a self-rolling process, which greatly shortens the time of wrapping these pieces. The self-rolling pieces normally have two seam lines that need to be sewn together. Figure 2 displays an example of a self-rolled sleeve piece. The key requirement of the self-rolling process is to keep the dimensions of the panel unchanged. This can be fulfilled through a cylindrical mapping method. Let x_{min} and x_{max} be the minimum and maximum x coordinates,
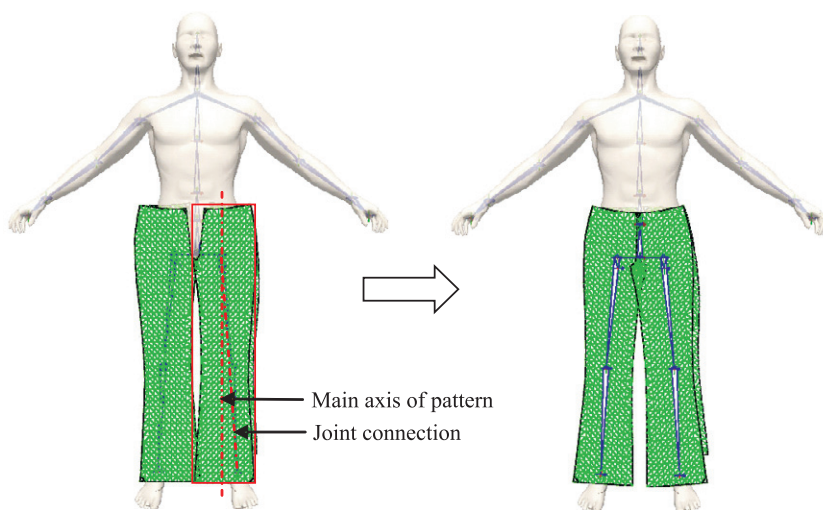
Figure 1 Body skeleton and initial positioning of panels.

Main axis of pattern
Joint connection
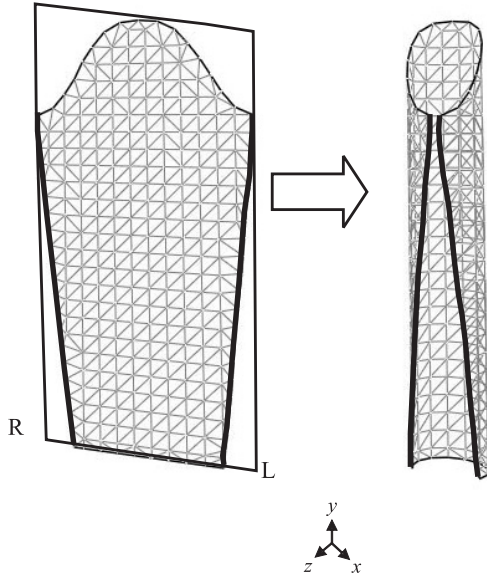
Figure 2 Self-rolling through cylindrical mapping.

Figure 3 Initial positioning of multi-panels.

and z_{min} be the minimum z coordinate of the panel. The radius of the cylinder is

$$r = \frac{x_{max} - x_{min}}{2\pi} + \zeta$$

where ζ is the gap between the left side (L) and the right side (R) on the cylindrical surface (Figure 2). The x and z coordinates of the cylinder axis are given by

$$O_x = \frac{x_{max} + x_{min}}{2} \quad \text{and} \quad O_y = z_{min} - r.$$

If the rolling of the flat panel occurs in the $+z$ direction, the new coordinates of particle P_0 can be calculated by

$$P_x = \alpha \cdot r \cdot \sin\theta + O_x$$
$$P_z = r \cdot \cos\theta + O_z$$
$$P_y = P_{0y}$$

where $\theta = \dfrac{\sqrt{(P_{0x} - O_x)^2 + (P_{0z} - z_{min})^2}}{r}$

and $\alpha = \begin{cases} 1 & P_{0x} > O_x \\ -1 & P_{0x} \le O_x \end{cases}$.

The self-rolling moves the seamlines of the two panels close to each other and, therefore, reduces the sewing time

in the wrapping stage. Figure 3 gives one example for initial positioning of panels around a digital model before the wrapping.

Adaptive Sewing Force

A robust sewing procedure often requires three basic tasks: (1) associating the corresponding seamlines of panels that need to be sewn together, (2) preventing oscillations of the particles on the seamlines when they approach each other, and (3) keeping the sewn panels untorn (surface penetration) during the entire simulation. These tasks can be achieved by using an adaptive sewing force with acceleration corrections [12] and a particle-merging scheme to secure the seamline when being pulled into a distance tolerance.

As shown in Figure 4, the sewing forces $\mathbf{f}_{sew}$ are imposed on the particles on the seamlines of two panels, and the directions are determined by the directional vectors between two corresponding nodes, P_a and P_b. The sewing force applied on P_a is in the direction of vector $\mathbf{x}_{ab} = \mathbf{x}_b - \mathbf{x}_a$, where $\mathbf{x}_a$ and $\mathbf{x}_b$ are the position vectors of P_a and P_b, respectively. Similarly, the sewing force on P_b is in the direction of $\mathbf{x}_{ba}$. To bring P_a and P_b together smoothly, $\mathbf{f}_{sew}$ on P_a is constructed as a function of the distance ($\varepsilon \mathbf{x}_{ab}$), air resistance ($-\tau\mathbf{v}_a$), and an oscillation stabilizer ($\Delta\mathbf{A}_a$)

Table 1 Coefficients for computing sewing force in our practice.

ε	τ	c	λ_f	λ_{damp_n}	λ_{damp_t}	λ_{atten_n}	λ_{atten_t}
1.0	0.2	0.8	0.66	0.17	0.83	0.03	0.0001
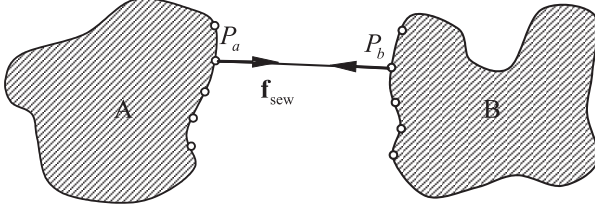
Figure 4 Illustration of sewing force.
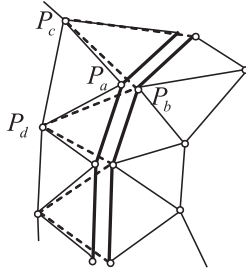
Figure 5 Merging the nodes/particles on the seamlines.

$$\mathbf{f}_{\text{sew}} = \begin{cases} \varepsilon \mathbf{x}_{ab} - \tau \mathbf{v}_a + \Delta \mathbf{A}_a & |\mathbf{x}_{ab}| > \delta \\[2mm] c \dfrac{\mathbf{x}_{ab}}{|\mathbf{x}_{ab}|} & |\mathbf{x}_{ab}| \le \delta \end{cases}$$

where ε, τ, and c are the coefficients, δ is the distance tolerance, and $\Delta\mathbf{A}_a$ is the acceleration correction at P_a, which can be derived from the total acceleration correction $\Delta\mathbf{A}_{ab}$ [12]. $\Delta\mathbf{A}_{ab}$ is defined as follows:

$$\Delta\mathbf{A}_{ab} = (\lambda_f + \lambda_{damp\_n})\mathbf{x}_{ab} + (\lambda_f + \lambda_{damp\_n})\mathbf{v}_{ab}^N$$

$$+ \lambda_{damp\_t}\mathbf{v}_{ab}^T + \lambda_{atten\_n}\mathbf{a}_{ab}^N + \lambda_{atten\_t}\mathbf{a}_{ab}^T$$

$$\Delta\mathbf{A}_a = \frac{m_a^{-1}}{m_a^{-1} + m_b^{-1}}\Delta\mathbf{A}_{ab}, \quad \text{and} \quad \Delta\mathbf{A}_b = -\frac{m_b^{-1}}{m_a^{-1} + m_b^{-1}}\Delta\mathbf{A}_{ab}$$

where $\mathbf{v}_{ab}^N$, $\mathbf{v}_{ab}^T$, $\mathbf{a}_{ab}^N$, and $\mathbf{a}_{ab}^T$ are the relative velocities and relative accelerations of P_a to P_b in $\mathbf{x}_{ab}$ and its transversal direction, respectively. The coefficients λ_{damp_n}, λ_{damp_t},

λ_{atten_n}, and λ_{atten_t} represent the damping effects on the sewing force, the relative velocity and acceleration. m_a and m_b are the masses of P_a and P_b. Different combinations of the coefficients in computing $\mathbf{f}_{\text{sew}}$ can provide different sewing results, and it needs trial tests to find out an optimal combination. The values of these coefficients employed in our practice are given in Table 1.

When $|\mathbf{x}_{ab}|$ is greater than δ, the magnitude of the sewing force $\mathbf{f}_{\text{sew}}$ changes with $|\mathbf{x}_{ab}|$. When $|\mathbf{x}_{ab}|$ is smaller than δ, the two nodes are about to be sewn together, and the magnitude of $\mathbf{f}_{\text{sew}}$ should be kept at a lower level ($c < 1$), facilitating the final sewing procedure. We chose $c = 0.8$ after several tests.

To keep the sewn panels together, the next action is to merge P_a and P_b geometrically and change the spring connections accordingly, as shown in Figure 5. When P_a and P_b get close enough ($< 10^{-5}$ m), it is believed that these two nodes are sewn together, the sewing force is removed, and the spring connections to node P_a are replaced by the new spring connections to node P_b, that is, springs P_cP_a and P_dP_a are replaced by springs P_cP_b and P_dP_b. This change causes two nodes (P_a and P_b) to be glued together all the time, so that the merged panels can be treated as one panel in the subsequent procedures. The masses of the merged particles are combined, and the mass matrix of the mass-spring system needs to be changed accordingly.

Collision Detection and Response

The virtual dressing may involve wrapping multi-layer panels, which is conducted layer-by-layer. The surface of a human model is always regarded as the first layer, and the garment layers are regarded as the second layer, the third layer, etc. The geometrical surface of a previous layer is named as a subject surface, which is deemed to be motionless. The final 3D configuration of a virtual garment is dictated by the interactions of the particles of the garment pattern with those of the subject surface (cloth/object collisions) and among themselves (self-collisions). These two basic collisions can be further divided into point-triangle collisions and edge-edge collisions. The point-triangle collision occurs when a panel node collides with a triangle facet on the subject surface or a subject surface vertex collides with a panel triangle. Both need the same ray/triangle collision detection procedure, but different collision response mechanisms. In edge-edge collision detections, one edge of a panel triangle collides with an edge of a subject surface triangle.

To accelerate the collision detection between a garment surface and a subject surface, an axis-aligned bounding box (AABB) hierarchy tree is pre-calculated by recursive subdivisions of the subject surface [15–17]. At each time step, the proximity status between the garment surface and the AABB tree is checked. The checking starts with the root node of the AABB tree. If a collision occurs (i.e. the proximity condition is met), the collisions with its subsequent nodes. The checking is performed recursively until all the branching nodes of the AABB tree are checked. The found branching nodes lead to the actual triangles of the subject surface where collision detections need to be further performed.

Point-triangle collision detection and response

The first type of point-triangle collision is the collision between a panel particle and a triangle facet of the subject surface. At one instant of time, if the position of a panel particle P is $\mathbf{x}$ and the instant velocity vector is $\mathbf{v}$, an incident ray $(\mathbf{x}, \mathbf{v})$ can be constructed to test whether particle P has a tendency to intersect a triangle facet on the subject surface. A collision is announced when the distance between P and the intersection point is within a given tolerance ξ. Physically, particle P cannot pass through the subject surface. P needs to be bounced back when the collision is detected. This is so called the collision response. Let v_{rel}^{n-} and v_{rel}^{n+} be the magnitude of the relative velocities in the normal direction of the triangle before and after the colliding, respectively. A coefficient of restitution λ ($0 \le \lambda \le 1$) is employed to denote the energy loss for the frictionless collision model i.e. $v_{rel}^{n+} = -\lambda v_{rel}^{n-}$. Since the subject surface is motionless, the new relative velocity of P equals to its absolute velocity in the 3D space.

The second type of point-triangle collision is the collision between a panel triangle and a vertex of the subject surface, which is depicted in Figure 6. To detect if a panel triangle $P_1P_2P_3$ will collide with a vertex Q on the subject surface, the velocity $\mathbf{v}_G$ of the triangle centroid C_G is calculated, and then an incident ray $(Q, -\mathbf{v}_G)$ is draw at Q. This is based on the assumption that at one instant of time, the moving direction of triangle $P_1P_2P_3$ can be represented by the moving direction of its centroid C_G. If $(Q, -\mathbf{v}_G)$ intersects with triangle $P_1P_2P_3$ at an interior point P_4, and the distance between P_4 and Q is smaller than ξ, a collision is announced.

After calculating the barycentric coordinates of P_4 in triangle $P_1P_2P_3$ as w_1, w_2, and w_3, the velocity of P_4 relative to Q before the collision can be computed as $\mathbf{v}_{4\_rel}^{-} = w_1\mathbf{v}_1^{-} + w_2\mathbf{v}_2^{-} + w_3\mathbf{v}_3^{-}$, and its magnitude in the collision normal direction $\mathbf{n}_Q$ is $v_{4\_rel}^{-} = \mathbf{v}_{4\_rel}^{-} \cdot \mathbf{n}_Q$. Here, the collision normal $\mathbf{n}_Q$ equals to the normal of the vertex Q in the subject surface, and is set to be the average normal of all the facets that share vertex Q. According to the frictionless
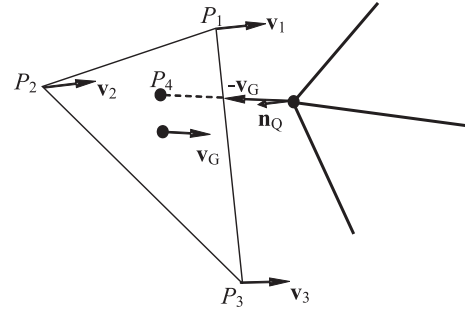
Figure 6 Collision between a subject surface vertex and a panel triangle (point-triangle).

collision model, the magnitude of the relative velocity of P_4 in the collision normal direction after the collision is $v_{4rel}^{n+} = -\lambda v_{4rel}^{n-}$. Therefore, the velocity of P_4 after collision can be synthesized in terms of the collision normal $\mathbf{n}_Q$. Since Q is motionless, the relative velocity difference of P_4 equals to its absolute velocity difference.

Governed by the momentum conservation law, the impulse $\mathbf{I}$ in direction $\mathbf{n}_Q$ equals to the momentum change during the collision:

$$\mathbf{I} = m_4\mathbf{v}_4^{+} - m_4\mathbf{v}_4^{-} = (w_1 m_1 + w_2 m_2 + w_3 m_3)(\mathbf{v}_4^{+} - \mathbf{v}_4^{-}).$$

Assume that all the particles have the same mass m, and $w_1 + w_2 + w_3 = 1$. We have

$$\mathbf{I} = m(\mathbf{v}_4^{+} - \mathbf{v}_4^{-}) = m\Delta\mathbf{v}_4.$$

Due to the nature of the mass-spring system, the velocity is specified only for the triangle vertices. If one point inside the triangle or on the edge collides with other objects, the collision impulse can be distributed to the triangle vertices in terms of the affine deformation rule. According to Bridson et al. [11], the adjusted impulse $\tilde{\mathbf{I}}$ applied on the triangle vertices P_1, P_2, and P_3 is given by

$$\tilde{\mathbf{I}} = \frac{2\mathbf{I}}{1 + w_1^2 + w_2^2 + w_3^2} = \frac{2m\Delta\mathbf{v}_4}{1 + w_1^2 + w_2^2 + w_3^2}$$

and the new velocity of each vertex is

$$\mathbf{v}_i^{+} = \mathbf{v}_i^{-} + w_i\left(\frac{\tilde{\mathbf{I}}}{m}\right) = \mathbf{v}_i^{-} + \frac{2w_i}{1 + w_1^2 + w_2^2 + w_3^2}(\Delta\mathbf{v}_4);$$
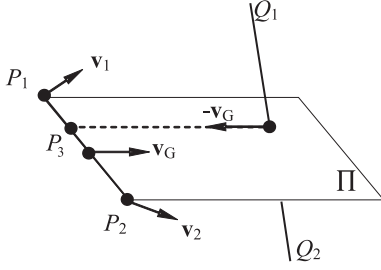
$$i = 1, 2, \text{ and } 3.$$

Figure 7 Edge-edge collision.

Edge-edge collision detection and response

In the edge-edge collision (see Figure 7), the instant velocity of centroid C_G of edge P_1P_2 of a panel triangle is employed as the moving direction towards the triangle edge Q_1Q_2 in the subject surface. The possible colliding point between P_1P_2 and Q_1Q_2 can be found on a plane Π that is formed by using P_1P_2 and $\mathbf{v}_G$. $\mathbf{v}_G$ stands for the velocity vector of C_G that can be determined by the masses and velocities of P_1 and P_2 as follows:

$$\mathbf{v}_G = (m_1\mathbf{v}_1 + m_2\mathbf{v}_2)/(m_1 + m_2).$$

When $m_1 = m_2$, $\mathbf{v}_G = (\mathbf{v}_1 + \mathbf{v}_2)/2$. Assume that Q_1Q_2 intersects with the moving plane Π at point Q_3. The incident ray $(Q_3, -\mathbf{v}_G)$ will intersect with P_1P_2 at P_3, which is the potential collision point on edge P_1P_2. If the distance between Q_3 and P_3 is smaller than the tolerant distance ξ, a collision is announced.

To evaluate the new velocities of P_1 and P_2 after the collision, the velocity change at P_3 needs to be calculated. Before the collision, the velocity at P_3 is calculated as

$$\mathbf{v}_3 = (1 - w_1)\mathbf{v}_1 + w_1\mathbf{v}_2$$

where $w_1 = |\mathbf{x}_1 - \mathbf{x}_3|/|\mathbf{x}_1 - \mathbf{x}_2|$, and $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ stand for the position vectors of P_1, P_2, and P_3, respectively. Here the collision normal at P_3 is defined by the cross product of the two colliding edges. As the edge orientations of P_1P_2 and Q_1Q_2 are not pre-defined, it is necessary to test the dot product of the normal and the moving direction $\mathbf{v}_G$. If it is positive, it then needs to invert the normal before making a correct collision response. If the collision normal is $\mathbf{n}$, the magnitude of P_3's relative velocity in the collision normal direction before the collision is $v_{3\_rel}^{n-} = \mathbf{v}_{3\_rel}^{n-} \cdot \mathbf{n}$. The magnitude of P_3's relative velocity in the collision normal direction after the collision is $v_{3\_rel}^{n+} = -\lambda v_{3\_rel}^{n-}$, from which the velocity difference of P_3 caused by the collision can be obtained.

When distributing the impulse onto P_1 and P_2, we have

$$\tilde{\mathbf{I}} = \frac{2\mathbf{I}}{w_1^2 + (1 - w_1^2) + w_2^2 + (1 - w_2^2)}$$

$$= \frac{2m\Delta\mathbf{v}_4}{w_1^2 + (1 - w_1^2) + w_2^2 + (1 - w_2^2)}$$

where $w_2 = |\mathbf{y}_1 - \mathbf{y}_3|/|\mathbf{y}_1 - \mathbf{y}_2|$, and $\mathbf{y}_1$, $\mathbf{y}_2$, and $\mathbf{y}_3$ stand for the position vectors of Q_1, Q_2, and Q_3, respectively. Therefore, the new velocity of P_1 and P_2 are calculated as

$$\mathbf{v}_1^+ = \mathbf{v}_1^- + (1 - w_1)\left(\frac{\tilde{\mathbf{I}}}{m}\right)$$

$$= \mathbf{v}_1^- + \frac{(1 - w_1)}{w_1^2 + (1 - w_1^2) + w_2^2 + (1 - w_2^2)}(\Delta\mathbf{v}_3), \text{ and}$$

$$\mathbf{v}_2^+ = \mathbf{v}_2^- + (w_1)\left(\frac{\tilde{\mathbf{I}}}{m}\right)$$

$$= \mathbf{v}_2^- + \frac{w_1}{w_1^2 + (1 - w_1^2) + w_2^2 + (1 - w_2^2)}(\Delta\mathbf{v}_3).$$

Strain Control and Size Stability

The size stability of a pattern means the changes in its perimeter and area remain in the given limits in the entire transformation. The size stability is essential for assessment of a virtual garment fit. To ensure the size stability of the pattern, the strains of all the connecting springs have to be controlled under a tolerance. Although some other researchers chose 10 % as a strain limit for a mass-spring system [7, 11], we found that only when the strain limit is set as low as 1 %, the stability of a pattern can be effectively maintained. A spring may incur an "over-elongation" or "over-compression" problem due to a loose strain limit [7].

Figure 8(a) shows the positions ($\mathbf{x}_0$ and $\mathbf{x}_1$) and velocities ($\mathbf{v}_0$ and $\mathbf{x}_1$) of the two end-points of a spring at time t. Their next positions at $t + \Delta t$ can be estimated using the Euler integration i.e.

$$\mathbf{x}_0(t + \Delta t) = \mathbf{x}_0(t) + \mathbf{v}_0(t) \cdot \Delta t, \text{ and}$$

$$\mathbf{x}_1(t + \Delta t) = \mathbf{x}_1(t) + \mathbf{v}_1(t) \cdot \Delta t.$$

The spring strain at the new position will be $|\mathbf{x}_0(t + \Delta t) - \mathbf{x}_1(t + \Delta t)|/l$ (l is the natural length of the spring). If the strain at $t + \Delta t$ exceeds the strain limit (1 %), $\mathbf{x}_0(t + \Delta t)$ and $\mathbf{x}_1(t + \Delta t)$ can be moved along the same direction until the strain falls into the limit (see Figure 8(b)). To ensure the two end-points reach their required new positions $\mathbf{x}_0^{new}(t + \Delta t)$ and $\mathbf{x}_1^{new}(t + \Delta t)$ at $t + \Delta t$, the velocities of the two end-points at t need to be adjusted as follows:
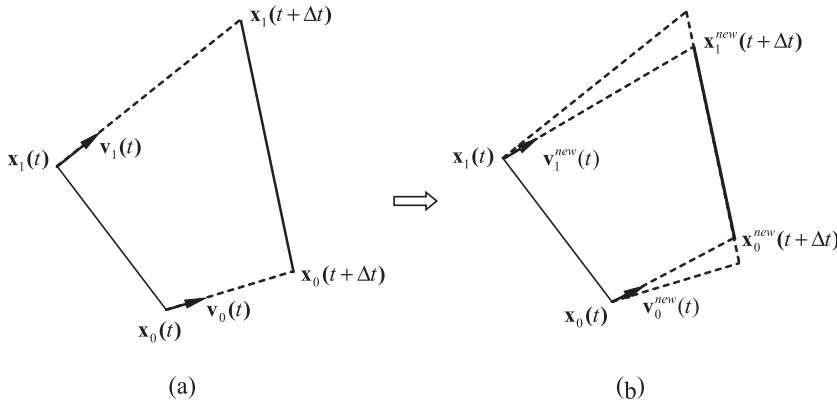
Figure 8 Strain control through velocity adjustment.
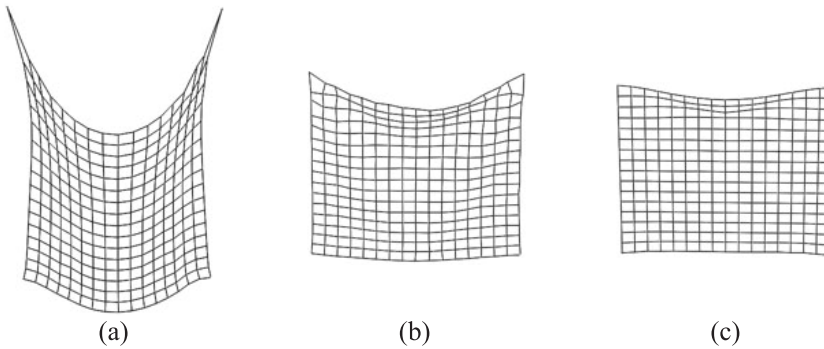
Figure 9 A hanging cloth with different strain control methods: (a) no strain control; (b) strain control with the position adjustment; and (c) strain control with the velocity adjustment.

$$\mathbf{v}_0^{new}(t) = [\mathbf{x}_0^{new}(t + \Delta t) - \mathbf{x}_0(t)]/\Delta t, \quad \text{and}$$

$$\mathbf{v}_1^{new}(t) = [\mathbf{x}_1^{new}(t + \Delta t) - \mathbf{x}_1(t)]/\Delta t,$$

To prevent a potential penetration, the routine of the collision detection and response needs to be invoked when the new positions $\mathbf{x}_0^{new}(t + \Delta t)$ and $\mathbf{x}_1^{new}(t + \Delta t)$ are calculated.

According to the momentum-conservation law, the changes of the velocities induce impulses on the two endpoints of the spring. Theoretically, these impulses will have impacts on other nodes of the mass-spring network and, therefore, should be spread throughout the whole system with an iteration method, such as the well-known Jacobi iteration or Gauss-Seidel iteration, to reach an ideal situation that changing the strain on one spring does not change the strains of others. In practice, since the strain control is executed at every step, the velocity changes at one particular step do not have to be significant. Therefore, the average of the three iterations of velocity adjustments at one time step can be used as the final adjusted velocity for simplicity. The actual experimental results confirm that this simplification neither causes any instability nor compromises the effectiveness of the strain control scheme.

To verify the strain control performance of the velocity adjustment, we tested it with a 17 × 17 mesh model of a hanging cloth, which was also experimented by Provot with the position adjustment scheme [7]. Due to the stress concentration at the hanging corners, the springs connected to the corners are over elongated when there is no strain control scheme implemented, as shown in Figure 9(a). Figure 9(b) displays the hanging effect reproduced using the position correction scheme in which the strain limit was set at 10 %, and Figure 9(c) is the strain control result of using the velocity adjustment scheme. Both methods demonstrate that the over-elongations are curtailed effectively at the hanging corners, while the visual effect of the velocity adjustment seems to be "stiffer" than that of the position adjustment. This is because the velocity adjustment allows a more strict strain control (1 %).

Figure 10 displays the changes of the total area and the perimeter of all the triangles in a panel during the dressing simulation. The two sets of curves clearly demonstrate that a panel's area and perimeter vary dramatically when, and maintain at the stable levels when the strains are controlled using the velocity adjustments. The maximum changes in the area and the perimeter are only at 1.44 % and 0.87 %, which ensure the size stability in the virtual dressing to be comparable to that in the real dressing.
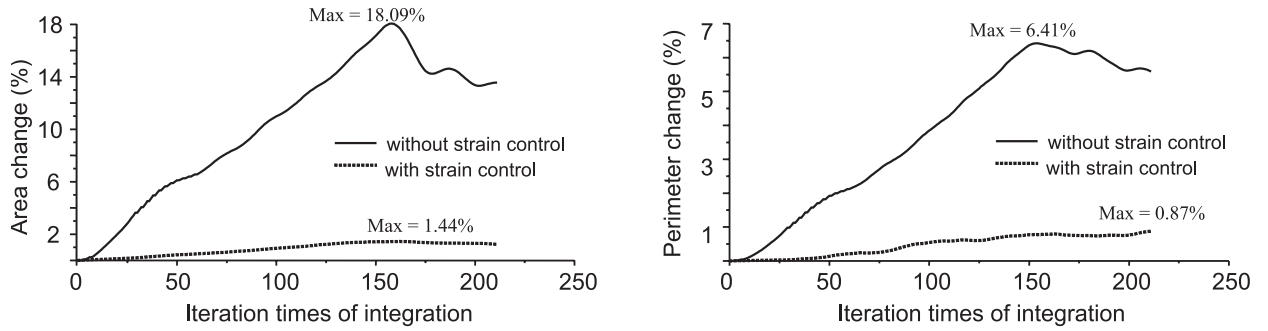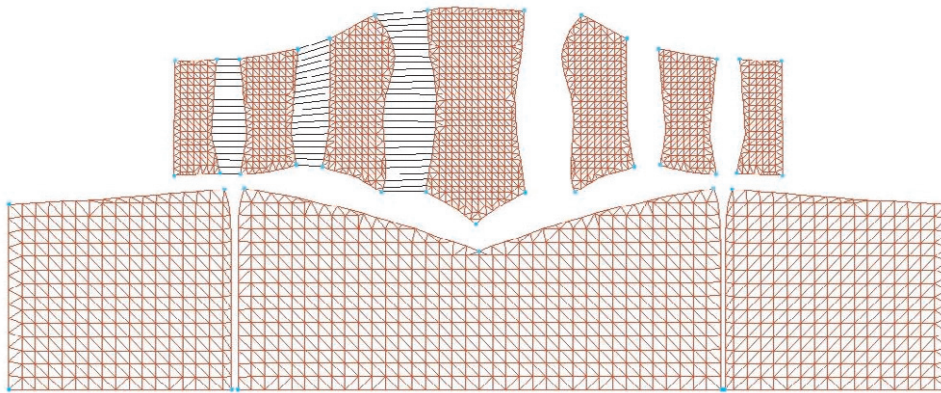
Figure 10 Changes in the area and perimeter of a panel during the dressing simulation.

Figure 11 Image series of virtual dressing.

(a) Meshing the input patterns.

CIRCUMFERENCE MEASUREMENTS :
1. Bust/Chest:	93.64
2. Waist:	70.66
3. Abdomen:	85.97
4. Hip:	98.00

Unit: cm /inch

UPPER TORSO (bodice):
5. Center length:	
Front:	39.20
Back:	40.99
6. Full length:	
Front:	42.38
Back:	45.00
7. Shoulder slope:	
Front:	38.72
Back:	38.86
8. Strap:	
Front:	0.00
Back:	0.00
9. Bust depth/radius:	0.00 0.00
10. Bust span:	32.85
11. Side length:	0.00
12. Back neck:	0.00
13. Shoulder length:	0.00
14. Across shoulder:	
Front:	20.09
Back:	0.00
15. Across chest:	0.00
16. Across back:	0.00
17. Bust arc:	0.00
18. Back arc:	0.00
19. Waist arc:	
Front:	0.00
Back:	0.00
20. Dart placement:	0.00

LOWER TORSO (Skirt / Pant):
21. Abdominal arc:	
Front:	0.00
Back:	0.00
22. Hip arc:	
Front:	0.00
Back:	0.00
23. Hip depth:	
Front:	0.00
Back:	0.00
24. Vertical trunk:	0.00
25. Waist to ankle:	107.94
Waist to floor:	0.00
Waist to knee:	0.00
26. Crotch depth:	30.38
27. Crotch length:	74.53
28. Upper thigh:	55.77
Mid-thigh:	47.91
29. Knee:	35.31
30. Calf:	3.92
31. Ankle:	1.81

ARM:
32. Elbow:	0.00
33. Wrist:	0.00

FREEHAND MEASUREMENT:
Girth:	0.00
Linear distance:	0.00
Height:	0.00
Curve length:	0.00
Slope girth:	0.00
Free cut torso girth:	0.00
Free cut Left Arm girth:	0.00
Free cut Right Arm girth:	0.00
Free cut Left Leg girth:	0.00
Free cut Right Leg girth:	0.00
Sliding slope girth:	0.00

Model calibration scale:	0.12
Pattern calibration scale:	0.00
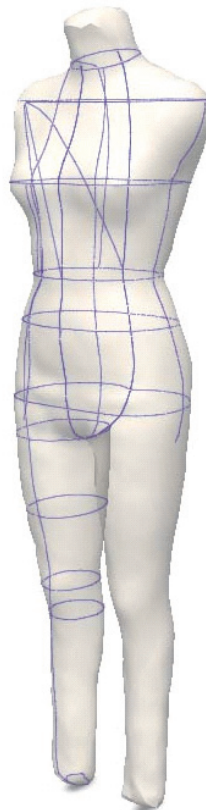
(b) Body scanning and modeling.

Figure 12 An integrated system for virtual dressing.

Initial positioning   Wrapping   Draping

(c) Virtual dressing.

Photo   Simulation   Ease map
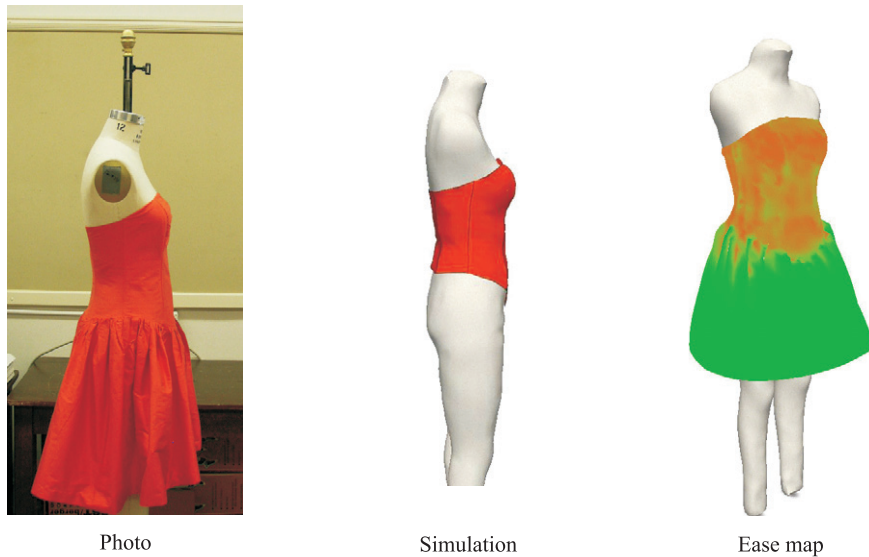
(d) Fit analysis with the ease map.

Figure 12 An integrated system for virtual dressing. *(Continued)*

Implementation

The 3D dressing simulation, including the algorithms elaborated above, can be implemented in the following procedure:

1. Select a time step Δt and set $t_{n+1} = t_n + \Delta t$.
2. Compute the force applied on each node.

3. Adjust the initial velocity of each node and rectify those nodes and edges of triangular facets colliding with the subject surface.
4. Calculate the new velocities and positions at t_{n+1} through the numerical integration.
5. Carry out the collision detection and response at t_{n+1} to resolve the collisions.

(a) Unsewable.   (b) Sewable.
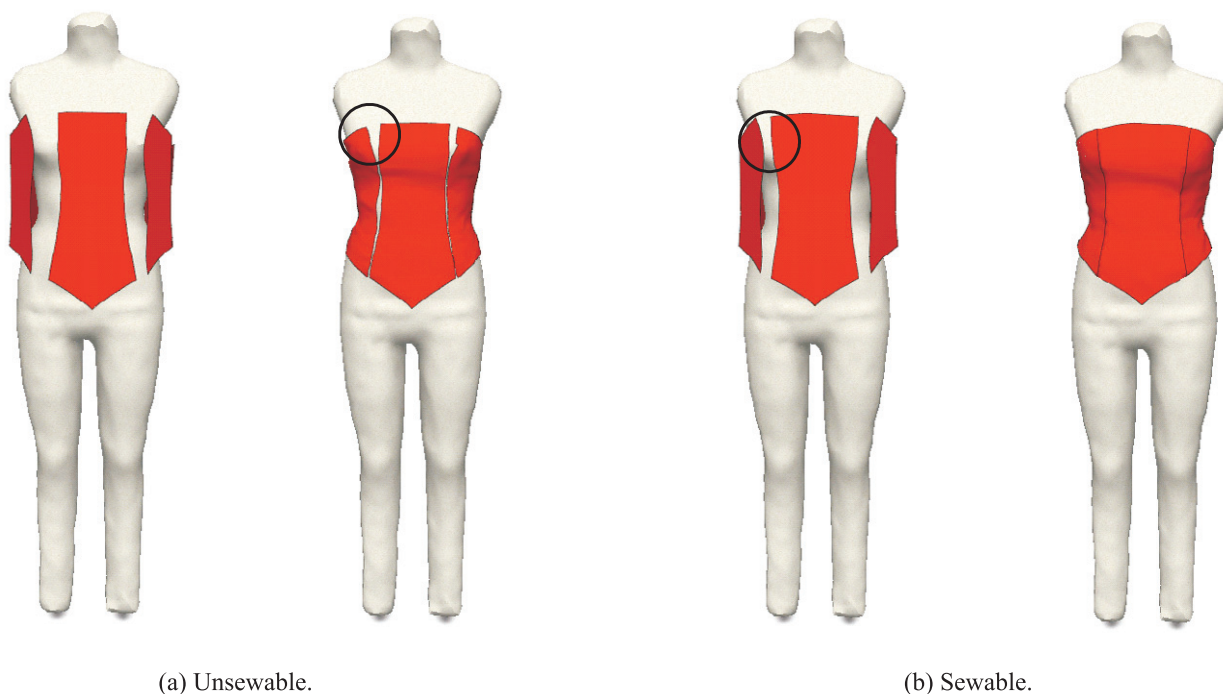
Figure 13 Sewability of the pattern.

6. Go to step 2) if the particles on the seamlines have not been sewn together and a satisfied draping effect has not been achieved.

To check the visual effects of the 3D garment dressing, we performed the dressing of different styles of garments onto different poses of a digital human model. As shown in Figure 11, the first series of images demonstrate a single layer dressing of a dress on a walking pose, while the second series of images present a multi-layer dressing of a pant and a jacket on a standing pose. The pant is dressed prior to the jacket. Both image series prove that the virtual dressing methods discussed in this paper can provide vivid, realistic dressing effects.

Figure 12 presents a few pictures output by an integrated 3D scanning and dressing system developed at the University of Texas at Austin. The system can automatically generate the mass-spring meshes for the pattern of a garment [13] (Figure 12(a)), scan and reconstruct a subject surface [14, 18] (Figure 12(b)), perform the virtual dressing of the pattern on the scanned subject (Figure 12(c)), and analyze the fit using an ease map that illustrates the localized gaps between the garment surface and the subject surface (Figure 12(d)). The garment example included in the pictures is a woman's dress that has ten separate panels.

After the mesh generation [13], the system also provides an interactive interface for the user to specify the sewing relationships among the seamlines. The number of the nodes (mass particles) on a pair of associated seamlines must be identical. The body dimension data can be extracted at the defined landmarks [19], which are needed for the fit analysis. Figure 12(c) exhibits the three phases of the virtual dressing – initial positioning of the panels, wrapping, and draping. Figure 12(d) includes a real photograph of the mannequin and the dress made with the same pattern, and a simulated picture of the bodice panels of the dress; both appear to have a good match. The ease map gives the quantitative data for the fit analysis.

Since the strain control can facilitate maintaining the designed dimensions of the pattern, the virtual garment can be used to assess the sewability of the pattern before the garment is physically produced. If two associated seamlines cannot be sewn together around a targeted body, the panels are considered unsewable (Figure 13(a)), and their dimensions must be increased based on the corresponding body dimensions. Unsewable panels can be revealed visually and modified immediately until they are sewable and a desired fit is realized (Figure 13(b)).

Summary

The 3D dressing simulation can be divided into three main stages: initial positioning, wrapping, and draping. In the initial positioning, the orientations of the body parts (e.g. arms or legs) were employed to determine the alignments of corresponding pattern pieces. Adaptive sewing forces were constructed based on the distances between mass particles and their nearest facets of the body surface, and were imposed on each mass particle to make the wrapping smooth and stable. The collision detection was accelerated through an axis-aligned bounding box (AABB) hierarchical tree. The size stability was achieved by controlling the spring strains through velocity adjustments at each time step. These presented algorithms seemed to be effective and efficient in simulating garment dressing.

Literature Cited

1. Baraff, D., and Witkin, A., Large Steps in Cloth Simulation, *ACM Trans. Graph. SIGGRAPH 1998 Comput. Graph. Proc.* 43–54 (1998).
2. Breen, D., House, D., and Wozny, M., Predicting the Drape of Woven Cloth using Interacting Particles, *ACM Trans. Graph. SIGGRAPH 1994 Comput. Graph. Proc.* 365–372 (1994).
3. Carignan, M., Yang, Y., Magenenat-Thalmann, N., and Thalmann, D., Dressing Animated Synthetic Actors with Complex Deformable Clothes, *ACM Trans. Graph. SIGGRAPH 1992 Comput. Graph. Proc.* 99–104 (1992).
4. Choi, K., and Ko, H., Stable but Responsive Cloth, *ACM Trans. Graph. SIGGRAPH 2002 Comput. Graph. Proc.* 604–611 (2002).
5. Eberhardt, B., Weber, A., and Strasser, W., A Fast, Flexible, Particle-system Model for Cloth Draping, *IEEE Comput. Graph. Appl.* **16**(3), 52–59 (1996).
6. Meyer, M., Debunne, G., Desbrun, M., and Barr, A., Interactive Animation of Cloth-like Objects in Virtual Reality, *J. Vis. Comput. Anim.* **12**(1), 1–12 (2001).
7. Provot, X., Deformation Constraints in a Mass-spring Model to Describe Rigid Cloth Behavior, *Graph. Interface* 147–154 (1995).
8. Terzopoulos, D., and Fleischer, K., Deformable models, *Vis. Comput.* **4**(6), 306–331 (1988).
9. Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K., Elastically deformable models, *ACM Trans. Graph. SIGGRAPH 1987 Comput. Graph. Proc.* 205–214 (1987).
10. Caramana, E., Burton, D., Shashkov, M., and Whalen, P., The Construction of Compatible Hydrodynamics Algorithms Utilizing Conservation of Total Energy, *J. Comput. Phys.* **146**, 227–262 (1998).
11. Bridson, R., Fedkiw, R., and Anderson, J., Robust Treatment of Collisions, Contact and Friction for Cloth Animation, *ACM Trans. Graph. SIGGRAPH 2002 Comput. Graph. Proc.* **21**(3), 594–603 (2002).
12. Volino, P., and Magnenat-Thalmann, N., "Virtual Clothing: Theory and Practice," Spring-Verlag, Berlin Heidelberg, Germany (2000).
13. Xu, B., and Zhong, Y., 3D Technology for Apparel Mass Customization, Part III: Visualization of 3D Garments, *J. Textile Inst.* **94**(1), 92–102 (2003).
14. Xu, B., Yu, W., and Chen, T., 3D Technology for Apparel Mass Customization, Part II: Human Body Modeling from Unorganized Range Data, *J. Textile Inst.* **94**(1), 81–91 (2003).
15. Barequet, G., Chazelle, B., Guibas, L., Mitchell, J., and Tal, A., BOXTREE: A Hierarchical Representation for Surfaces in 3D, *Comput. Graph. Forum* **15**(3), 387–396 (1996).
16. Bergen, G., Efficient Collision Detection of Complex Deformable Models using AABB Trees, *J. Graph. Tools* **2**(4), 1–14 (1997).
17. Gottschalk, S., Lin, M., and Manocha, D., OBB-tree: A Hierarchical Structure for Rapid Interference Detection, *ACM Trans. Graph. SIGGRAPH 1996 Comput. Graph. Proc.* 171–179 (1996).
18. Xu, B., and Huang, Y., 3D Technology for Apparel Mass Customization, Part I: Rotary Body Scanning, *J. Textile Inst.* **94**(1), 72–80 (2003).
19. Zhong, Y., and Xu, B., Automatic Segmenting and Measurement on Scanned Human Body, *Int. J. Clothing Sci. Technol.* **18**(1), 19–30 (2006).