Dressing Virtual People

Tzvetomir I Vassilev
Department of Computer Science, University College London
Gower Street, London WC1E 6BT, United Kingdom

ABSTRACT

This paper describes a fast technique for dressing virtual humans with different pieces of clothing. The garments are constructed of cutting patterns seamed around the body. The system reads a body file and a cutting pattern file and produces a new body dressed with the specified product. The method exploits a mass-spring model of cloth but applies a new velocity modification approach to overcome its super-elasticity. This results in a more realistic cloth simulation with one more control parameter – stretching threshold. The algorithm for cloth-body collision detection is based on a tree of bounding boxes with the leaves being faces on the body surface and their normal vectors. Responses to collisions are resolved modifying velocities of the cloth vertices and not applying penalty forces or energy fields as in other approaches. As the results show the system produces realistic images and makes it possible to sew a garment around a virtual human body in several seconds.

Keywords: virtual clothing, cloth simulation, physically based modelling, 3D scanning

1. INTRODUCTION

The main objective of this work is to develop a technique for clothing virtual humans. The technique should be fast enough for an interactive system for dressing virtual people. The body model is acquired with a 3D scanner, so it represents very accurately the 3D body shape of a real person. The final goal is to implement a system on the WEB, where customers will be able to upload the 3D virtual representation of their body, browse different types of clothes, try them on, and buy, if they are satisfied. Because of the accuracy of the 3D scanning technology it will be possible not only to try on different types of clothes, but also to fit different sizes.

State of the art in body capturing technology

The fast development of 3D scanning technology evolved to whole body 3D scanners capable of capturing the 3D shape of the human body [1]. Now there are several manufacturers on the market producing such scanners: Hamamatsu Photonics (Japan), Cyberware (USA), TelMat (France), Vitonic Viro 3D (Germany), etc. The output of the scanners is a cloud of 3D points, but using additional software [2], a triangulated surface of the body can be obtained.

Previous work in cloth simulation

Physically based cloth modelling has been a problem of interest to computer graphics researchers for more than a decade. First steps, initiated by Terzopoulos et al. [3,4], characterised cloth simulation as a problem of deformable surfaces and used the finite element method and energy minimisation techniques borrowed from mechanical engineering. Since then other groups have been formed challenging the cloth simulation, using particle [5, 6] or energy [7, 8] based approaches. A more detailed survey on cloth modelling techniques can be found in the paper by Ng and Grimsdale [9].

Many of the approaches described above have a good degree of realism simulating the cloth but their common drawback is the low speed. A relatively good result demonstrated by Baraff and Witkin [8] is 14 seconds per frame for the simulation of a shirt with 6,450 nodes on a SGI R10000 processor. This is the main reason why these techniques cannot be applied to an interactive system working on the Internet.

Provot [10] used a mass-spring model to describe rigid cloth behaviour, which proved to be much faster than the techniques described above. Its major drawback is the super-elasticity. In order to overcome this problem he applied a position modification algorithm to the ends of the over-elongated springs. However, if this operation has modified the positions of many vertices, it may have elongated other springs. That is why this approach is applicable only if the deformation is locally distributed, which is not the case when simulating garments on a virtual body.

Contribution of this work

Our method also exploits a mass-spring cloth model but we introduce a new approach to overcome its super-elasticity. Instead of modifying the positions of end points of the springs that were already over-elongated, the algorithm checks their length after each iteration and does not allow elongation more than a certain threshold modifying velocities of the corresponding vertices.

Most of the existing techniques [3, 4, 8] resolve collisions applying repulsing or penalty forces, which makes the resulting system of equations complex and time-consuming to solve. After detecting collisions the algorithm, described in this paper, modifies velocities of the vertices on the cloth.

As a result of using only velocity modification for both coping with the super-elasticity and responding to collisions, our algorithm allows a very fast draping simulation. For example, the time for simulating a shirt of 1800 vertices draping on a human body of 5520 vertices is just about 30 ms per iteration on a SGI with R12000 processor.

The rest of the paper is organised as follows. The next section outlines the cloth model and its integration over time. Section 3 describes the technique for constraining the super-elasticity, while sections 4 and 5 are dedicated to dealing with collisions. Section 6 gives more details on the way of dressing a human body. Section 7 shows experimental results and section 8 outlines future work.

2. MASS-SPRING MODEL OF CLOTH

Topology

The elastic model of cloth is a mesh of $m \times n$ mass points, each of them being linked to its neighbours by massless springs of natural length greater than zero. There are three different types of springs:

- Springs linking vertices $[i, j]$ with $[i+1, j]$, and $[i, j]$ with $[i, j+1]$ are called "structural springs";

- Springs linking vertices [*i*, *j*] with [*i*+1, *j*+1], and [*i*+1, *j*] with [*i*, *j*+1] are called "shear springs";
- Springs linking vertices [*i*, *j*] with [*i*+2, *j*], and [*i*, *j*] with [*i*, *j*+2] are called "flexion springs".

As one can guess from the names the first type implements resistance to stretching, the second – resistance to shearing and the third – resistance to bending.

Forces

Let $\mathbf{p}_{ij}(t)$, $\mathbf{v}_{ij}(t)$, $\mathbf{a}_{ij}(t)$, where i=1,…, m and j=1,…, n, be correspondingly the positions, velocities, and accelerations of the mass points at time t. The system is governed by the basic Newton's law:

$$\mathbf{f}_{ij} = mass\ \mathbf{a}_{ij}, \tag{1}$$

where *mass* is the mass of each point and $\mathbf{f}_{ij}$ is the sum of all forces applied at point $\mathbf{p}_{ij}$. The force $\mathbf{f}_{ij}$ can be divided in two categories.

The **internal forces** are due to the tensions of the springs. The internal force applied at the point $\mathbf{p}_{ij}$ is a result of the stiffness of all springs linking this point to its neighbours:

$$\mathbf{f}_{int}(\mathbf{p}_{ij}) = -\sum_{k,l} k_{ijkl}\left(\overrightarrow{\mathbf{p}_{kl}\mathbf{p}_{ij}} - l^0_{ijkl}\frac{\overrightarrow{\mathbf{p}_{kl}\mathbf{p}_{ij}}}{\left\|\overrightarrow{\mathbf{p}_{kl}\mathbf{p}_{ij}}\right\|}\right), \tag{2}$$

where k_{ijkl} is the stiffness of the spring linking $\mathbf{p}_{ij}$ and $\mathbf{p}_{kl}$ and l^0_{ijkl} is the natural length of the same spring.

The **external forces** can differ in nature depending on what type of simulation we wish to model. The most frequent ones will be:

- Gravity: $\mathbf{f}_{gr}(\mathbf{p}_{ij}) = mass\ \mathbf{g}$, where $\mathbf{g}$ is the gravity acceleration;
- Viscous damping: $\mathbf{f}_{vd}(\mathbf{p}_{ij}) = -C_{vd}\mathbf{v}_{ij}$, where C_{vd} is a damping coefficient.

For more information on how to model wind see Provot [10].

All the above formulations make it possible to compute the force $\mathbf{f}_{ij}(t)$ applied on point $\mathbf{p}_{ij}$ at any time t. The fundamental equations of Newtonian dynamics can be integrated over time by a simple Euler method:

$$\left|\begin{array}{l}\mathbf{a}_{ij}(t+\Delta t) = \dfrac{1}{mass}\mathbf{f}_{ij}(t)\\ \mathbf{v}_{ij}(t+\Delta t) = \mathbf{v}_{ij}(t) + \Delta t\ \mathbf{a}_{ij}(t+\Delta t)\\ \mathbf{p}_{ij}(t+\Delta t) = \mathbf{p}_{ij}(t) + \Delta t\ \mathbf{v}_{ij}(t+\Delta t)\end{array}\right., \tag{3}$$

where Δt is a chosen time step. More complicated integration methods, such as Runge-Kutta [11], can be applied to solve the differential equations. This, however, reduces the speed significantly. Since efficiency is our main objective in this work, we decided not to implement expensive integration techniques. The Euler Eq. (3) are known to be very fast and give good results, when the time step Δt is less than the natural period of the system $T_0 \approx p\sqrt{mass/K}$. In fact our experiments showed that the numerical solving of Eq. (3) is stable when

$$\Delta t \le 0.4p\sqrt{\frac{mass}{K}}, \tag{4}$$

where K is the highest stiffness in the system.

3. CONSTRAINING SUPER-ELASTICITY

The major drawback of the mass-spring cloth model is that it is very elastic. This means that the deformation rate for some springs is too high and as a result the cloth stretches under its own weight, something that does not happen to real cloth.

Increasing stiffness

One way to avoid the super-elastic effect is to increase the stiffness of the springs. For the same external forces (in our case gravity and damping) the elongation rate should be lower for stiffer springs. This approach, however, has an obvious drawback. Eq. (4) shows that increasing the stiffness coefficient K will require decreasing the time step Δt. This means that for the same animation time the number of iterations will be greater and the algorithm will be more time consuming. Another drawback is that even if the stiffness is increased, the resulting material behaves more like rubber rather than cloth.

Position modification

Provot [10] proposed to cope with the super-elastic effect using position modification. His algorithm checks the length of each spring after each iteration and modifies the positions of the ends of the spring, if it exceeds its natural length with more than a certain value (10% for example). This modification will adjust the length of some springs but it might over-elongate others. So, the convergence properties of this technique are not clear. It proved to work for locally distributed deformations but no tests were conducted for global elongation.

Velocity modification

The main problem of the position modification approach is that first it allows the springs to over-elongate and then tries to adjust their length modifying positions. This of course is not always possible because of the many links between the mass points. Our idea was to find a constraint that does not allow any over-elongation of springs.

The algorithm works as follows. After each iteration it checks for each spring whether it exceeds its natural length by a pre-given threshold. If this is the case, the velocities are modified, so that further elongation is not allowed. The threshold value usually varies from 1% to 10% depending on the type of cloth we simulate.

Let $\mathbf{p}_1$ and $\mathbf{p}_2$ be the positions of the end points of a spring found as over-elongated, and $\mathbf{v}_1$ and $\mathbf{v}_2$ be their corresponding velocities (see Figure 1). The velocities $\mathbf{v}_1$ and $\mathbf{v}_2$ are split into two components $\mathbf{v}_{1t}$ and $\mathbf{v}_{2t}$, along the line connecting $\mathbf{p}_1$ and $\mathbf{p}_2$, and $\mathbf{v}_{1n}$ and $\mathbf{v}_{2n}$, perpendicular to this line. Obviously the components causing the spring to stretch are $\mathbf{v}_{1t}$ and $\mathbf{v}_{2t}$, so the algorithm sets them to zero. If this is applied to all strings, the stretching components of the velocities are removed and in this way further stretching of the cloth is not allowed. As the results show this approach works fine not only for local but also for global deformations.

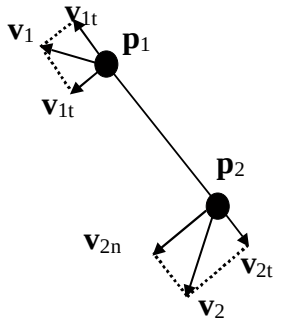

Figure 1. Velocity modification for over-elongated springs

4. COLLISION DETECTION

Our algorithm checks for collisions between each point of the cloth and each face of other objects in the scene. In our case this is the customer's scanned body, but it could also be any other rigid object. In order to avoid a big number of $O(n^2)$ comparisons, we first build a quad tree of bounding boxes for each solid object on which the cloth drapes. The leaves of the tree are the faces of the body surface with their normal vectors. A quad tree was implemented for the following reason. After processing the 3D scanner raw data the human body is segmented in six parts: head, torso, two arms and two legs. The surface of each body part is presented as a rectangular topology mesh (quad-mesh) of $m \times n$ vertices, where m is the number of horizontal body slices and n is the number of points within a slice. Subdivision in each of the two directions gives the natural quad structure. One such tree is built for each body part. The recursive search of the tree leads to $O(n \log_4 n)$ comparisons. For a static rigid body the tree and the normals for each face are computed in advance which speeds up the algorithm.

5. RESOLVING COLLISIONS

After a collision was detected, the algorithm has to compute a proper response of the whole system. Our approach does not introduce additional penalty, gravitational or spring forces; it just manipulates the velocities, as proposed by Moore and Wilhelms [12], which proved to be very efficient.

Let **v** be the velocity of the point **p** colliding with the surface **s** (Figure 2). The surface normal at the point of collision is denoted by **n**. If $\mathbf{v}_t$ and $\mathbf{v}_n$ are the tangent and normal components of **v**, then the resultant velocity can be computed as:

$$\mathbf{v}_{res} = C_{fric}\mathbf{v}_t - C_{refl}\mathbf{v}_n, \quad (5)$$

where C_{fric} and C_{refl} are a friction and a reflection coefficients, which depend on the material of the colliding objects.

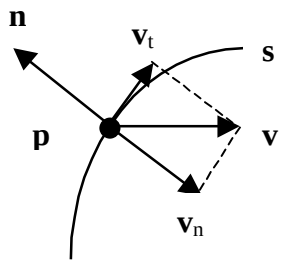

Figure 2. Resolving collisions manipulating velocities

6. DRESSING THE HUMAN BODY

Once a cloth model and algorithms for collision detection and response have been implemented, the only component needed to complete the system for dressing virtual humans is cutting patterns. Our software reads an input body file and a garment text file, describing the number of cutting patterns and necessary seaming information, and produces an output file of a dressed human body. If the body presented in the input file is already dressed, then the old clothing is considered as a solid object in the scene, i.e. it participates in the collision detection process. In this way it is possible to put on several layers or several products as shown in section 7.

Cutting pattern

The ideal solution would be to read output text files from the existing clothing CAD/CAM systems such as GERBER. They produce standard DXF or HPGL files describing the cutting pattern geometry and seaming information. Since we did not have such files at hand, for the first version we used standard drawing packages to create our own example patterns. Then we exported the outlines in a text file and typed in the necessary seaming information, i.e. along which lines the patterns are to be seamed together.

Meshing the patterns

As explained in the previous sections our cloth model is a rectangular topology mesh of $m \times n$ mass points connected with springs. So, in order to apply the algorithm, we had to implement meshing software that reads the cutting pattern outlines and produces the rectangular topology as shown on Figure 3.

After the cutting patterns have been produced and meshed, they are positioned around the body and elastic forces are applied along the seaming lines. After a certain number of iterations the patterns are seamed, i.e. the shirt is 'put on' the human body. Then several more iterations are performed applying gravity in order to drape the cloth.

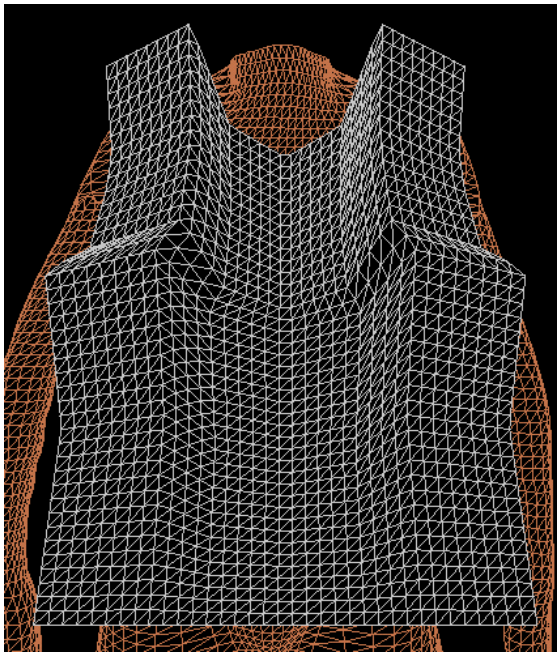

Figure 3. Cutting pattern of a shirt with a rectangular topology mesh

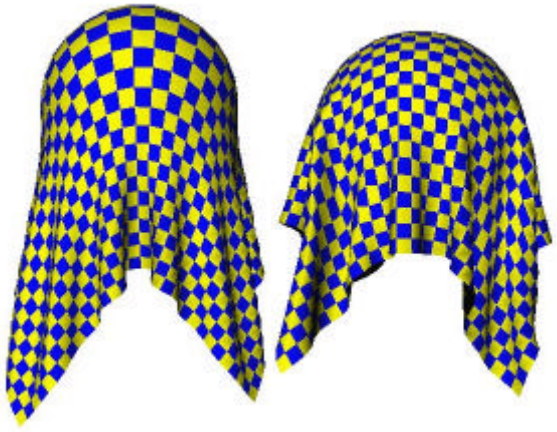

Figure 4. Left: original elastic model; right: model with velocity modification

7. RESULTS

The algorithms were implemented on a SGI workstation with a R12000 processor using the Open Inventor library for rendering the images. Figure 4 shows the draping of a rectangular piece of cloth over a sphere using the original elastic model with no restriction to stretching (left) and applying the velocity modification approach described above with a 5% elongation threshold (right). As one can see the original cloth model stretches over its own weight which is never the case with real cloth. The model with velocity modification overcomes this problem.

Figure 5. Draping the cloth over simple objects applying texture

Figure 5 shows rectangular pieces of cloth draping over simple objects: table and sphere. Incorporating textures makes images very realistic.

Figure 6 exhibits dressed human bodies with different pieces of clothing. The bodies were acquired at University College London using a Hamamatsu Photonics whole body 3D scanner. The dress, the skirt and the sleeveless shirt were built out of two cloth patches, the trousers – out of four and one patch was needed for each sleeve of the female shirt. An elasticity threshold of 5% was set for the cloth model. The numbers of vertices on the cloth are listed in Table 1, while the body consists of 5520 vertices.

Table 1 indicates the times necessary to dress a human body with different garments. The numbers of vertices were the minimum needed to produce visually pleasing results. A smaller number results in 'holes' on the cloth, that is the collision detection does not work properly because of the very rough mesh. The table does not include the set-up times necessary to read the files, generate the bounding box trees, etc.

Table 1. Times necessary to dress a virtual body with the listed products

Product	Number of cloth vertices	Number of iterations	Total time in sec	Time per iteration in ms
skirt	1152	80	1.25	15.6
shirt	1800	149	4.4	29.5
trousers	2048	147	4.7	31.9
dress	3840	150	7.8	52.0

Figure 6. Dressing virtual people with different pieces of clothing

Figure 7 illustrates how computational time changes with increasing the number of vertices on the cloth. As one can see, the dependence is linear which indicates the low $O(n)$ complexity of the algorithm.

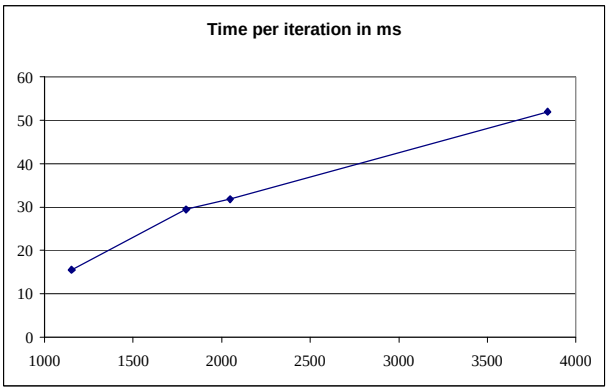

Figure 7. Time per iteration versus number of vertices on the cloth

8. CONCLUSIONS AND FUTURE WORK

An efficient technique for cloth simulation has been presented. It uses a mass-spring model with velocity modification methods for both coping with the super-elasticity of the original model and resolving responses to collisions. As a result the speed is good enough for almost real time simulation of draping virtual garments on virtual humans. The body surface is acquired through a 3D scanner, which makes the approach very accurate and allows users to try on different sizes of clothing and see how they fit them.

So far we conducted experiments only on a static human body. Our future plans are to implement dynamic simulation of garments on a moving virtual actor using the presented approach, evaluate its efficiency and see how it compares to other existing techniques.

9. ACKNOWLEDGEMENTS

This work is a part of the Foresight LINK award project "3D Centre for Electronic Commerce" at the University College London. Many thanks to Laura Dekker and Ioannis Douros for supplying the 3D body models and to Prof. Philip Treleaven and Yiorgos Chrysanthou for their help and ideas.

10. REFERENCES

[1] C. Horiguchi, Hamamatsu, "BL (Body Line) Scanner", International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, Part 5, 1998.

[2] L. Dekker, I. Douros, B.F. Buxton, P. Treleaven, "Building Symbolic Information for 3D Human Body Modelling from Range Data", Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling, IEEE Computer Society, 1999, pp. 388-397.

[3] D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, "Elastically Deformable Models", Computer Graphics (Proc. SIGGRAPH), Vol. 21, No 4, 1987, pp. 205-214.

[4] D. Terzopoulos and K. Fleischer, "Deformable Models", Visual Computer, Vol. 4, 1988, pp. 306-331.

[5] D.E. Breen, D.H. House and M.J. Wozhny, "Predicting the drape of woven cloth using interacting particles", Computer Graphics (Proc. SIGGRAPH), Vol. 28, 1994, pp. 23-34.

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

[7] M. Carignan, Y. Yang, N. Magnenat Thalmann and D. Thalmann, "Dressing animated synthetic actors with complex deformable clothes", Computer Graphics (Proc. SIGGRAPH), Vol. 26, No. 2, 1992, pp. 99-104.

[8] D. Baraff and A. Witkin, "Large Steps in Cloth Simulation", Computer Graphics (Proc. SIGGRAPH), 1998, pp. 43-54.

[9] N.H. Ng and R.L. Grimsdale, "Computer graphics techniques for modelling cloth", IEEE Computer Graphics and Applications, Vol. 16, 1996, pp. 28-41.

[10] X. Provot, Deformation constraints in a mass-spring model to describe rigid cloth behaviour, Proc. of Graphics Interface, 1995, pp. 141-155.

[11] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, Numerical Recipes in C: the Art of Scientific Computations, Cambridge University Press, 1992.

[12] M. Moore and J. Wilhelms, "Collisions detection and response for computer animation", Computer Graphics (Proc. SIGGRAPH), Vol. 22, No. 4, 1998, pp. 289-298.