EUROGRAPHICS 2002 / G. Drettakis and H.-P. Seidel
(Guest Editors)

Volume 21 (2002), Number 3

Real-time Animation of Dressed Virtual Humans

Frederic Cordier, Nadia Magnenat-Thalmann

MIRALab, CUI, University of Geneva, Switzerland

E-mail: {cordier, thalmann}@miralab.unige.ch

Abstract

In this paper, we describe a method for cloth animation in real-time. The algorithm works in a hybrid manner exploiting the merits of both the physical-based and geometric deformations. It makes use of predetermined conditions between the cloth and the body model, avoiding complex collision detection and physical deformations wherever possible. Garments are segmented into pieces that are simulated by various algorithms, depending on how they are laid on the body surface and whether they stick or flow on it. Tests show that the method is well suited to fully dressed virtual human models, achieving real-time performance compared to ordinary cloth-simulations.

1 Introduction

One of the most challenging areas in research is in the development of a robust methodology for simulating clothes in real-time. In order to define a cloth simulation system that is able to simulate complex garments realistically, whilst maintaining a reasonable computation time, a deeper study of the cloth model and the identification of its behaviour at different levels are necessary.

This study is not intended to integrate yet another more precise physical model of a garments behaviour, but rather focus on the real-time constraints for the simulation and the visual cloth motion features to which an observer is sensitive. Most of the existing approaches use a general-purpose simulation method using collision detection and physical simulation for the whole garment. Unfortunately, simulations that simply calculate all potentially colliding vertices may generate a highly realistic movement, but do not provide a guaranteed frame time. A new simulation model should be implemented that avoids heavy calculation of the collision detection and particle system wherever possible.

Our assumption is that the whole cloth does not need to be simulated with a general-purpose simulation method; instead many optimizations can be made. For example, the trouser will never collide with the arms. Collision detection may be simplified by restricting the collision detection to only potentially colliding surfaces. Also, stretched garments do not need to be simulated with a complex physical method. It can be simply simulated by keeping an offset between the garment and the underlying skin surface. We aim to develop a hybrid approach exploiting the merits of both the physics-based and geometric deformations and predetermined conditions between the cloth and the body model. The method is based on a multi-layer approach where each layer is simulated by its own dedicated method. A pre-processing stage aims to define the three regions corresponding to stretch, loose and floating clothes. During the real-time simulation, they will be animated using different methods.

The paper is organized as follows: The first section reviews previous work on real-time cloth simulation. In the second section we will provide an overview of the method, then, we will describe in detail the garment design and segmentation, and the three cloth layers used for the simulation. In the last section, we will present the results of our method when applied to virtual humans.

2 Previous work

Extensive research has been carried out in cloth simulation. Several of them have focused on the quality of garment simulation, where constraint of real-time was not a priority. Their aim was to develop a physics-based method that is able to simulate the dynamics of cloth independent of its use whether as clothing or other situations like in furnishing tablecloth. They integrated complex collision detection and they were able to simulate the physical behaviour of garments [1], [2], [3], [4].

Other research has focused on the real-time aspect of the animation of deformable objects using physical simulation.

Baraff et al. [5] have used the Implicit Euler Integration method to compute the cloth simulation in real time. They stated that the bottleneck of real-time cloth simulation is the fact that the time-step must to be small in order to avoid instability. They described a method that can stably take large time steps, suggesting the possibility of real-time animation of simple objects.

Meyer et al. [6] and Desbrun et al. [7] have used a hybrid explicit/implicit integration algorithm to animate real-time clothes, integrated with this is a voxel-based collision detection algorithm. Their method seems to be limited by the maximum number of polygons they can animate in real-time.

Other research has focused on the collision detection, stating that it is one of the bottlenecks to real-time animation. Vassilev et al. [9] proposed to use the z-buffer for collision detection to generate depth and normal maps. Computation time of their collision detection does not depend on the complexity of the body. However, the maps need to be pre-computed before simulation, restricting the real-time application.

Another approach presented by Grzeszczuk et al. [10] uses a neural network to animate dynamic objects. They replaced physics-based models by a large neural network that automatically learns to simulate similar motions by observing the models in action. Their method works in real-time. However, it has not been proven that this method can be used for complex simulation such as cloth.

James et al. [11] have also worked on real time simulation; their paper describes the boundary integral equation formulation of static linear elasticity as well as the related Boundary Element Method discretization technique. Their model is not dynamic, but rather a collection of static postures, limiting its potential applications.

Debunne *et* al. [12] have recently introduced a technique for animating soft bodies in real time. However, their method works on volumetric meshes. Therefore, it is not applicable to thin objects such as cloth.

Some other researchers have used geometrical approaches [13] [14] [15] [16]. Geometrical models do not consider the physical properties of the cloth, therefore providing techniques that produce fast results. However, these techniques are not able to reproduce the dynamics of clothes. Moreover, geometrical techniques require a considerable degree of user intervention. They can be regarded as a form of advanced drawing tools.

Hybrid approaches try to combine geometrical and physical deformations. Kang et al. [17] [8] improved the visual quality of the garments of small number of polygons by tessellating the triangles. With a wrinkled cubic spline curve, their tessellation algorithm is able to simulate the wrinkles. Oshita et al. [18] use a similar approach. Both these methods are mainly applicable to flat surfaces where physical simulation can be done with a very small number of polygons. However, highly curved surfaces, such as sleeves, need to be simulated with a higher number of polygons.

To our knowledge, no previously established technique exists for simulating fully dressed virtual humans in real-time using a physical simulation. The performance leap necessary to obtain the real-time simulation of complete garments cannot be obtained by further optimization of classic simulation techniques, despite the recent developments of simple models using particle systems, implicit integration and optimized collision detection. They require more drastic simplifications of the simulation process to be carried out, possibly at the expense of mechanical and geometrical accuracy. In this paper, we propose a method that is based on a hybrid approach where the cloth is segmented into various sections where different algorithms are applied.

3 Overview of the approach

When observing a garment worn on a moving character, we notice that the movement of the garment can be classified into several categories depending on how the garment is laid on and whether it sticks to, or flows on, the body surface. For instance, a tight pair of trousers will mainly follow the movement of the legs, whilst a skirt will float around the legs. The first part of the study is to identify all the possible categories:

- Garment regions that stick to the body with a constant offset. In this case, the cloth follows exactly the movement of the underlying skin surface.
- Garment regions that flow around the body. The movement of the cloth does not follow exactly the movement of the body. In case of a long skirt, the left side of the skirt can collide with the right legs.
- Garment regions that move within a certain distance to the body surface are placed in another category. The best examples are shirtsleeves. The assumption in this case is that the cloth surface always collides with the same skin surface and its movement is mainly perpendicular to the body surface.

These three categories are animated with three different cloth layers. The idea behind the proposed method is to avoid the heavy calculation of physical deformation and of collision detection wherever possible, i.e. where collision detection is not necessary. The main interest of our approach is to pre-process the target cloth and body model so that they are efficiently computable during runtime. The skin and the garment are divided into a set of segments and the associated simulation method is defined for each. For each layer, we propose solutions and explain why they have been chosen.

4 Body and garment segmentation

Any simulation of clothes requires choosing a body model and designing the garments. This section provides a description of the body model and the way it is deformed as the skin deformation is an important component of our cloth deformation model. A description of the garment segmentation is also supplied.

4.1 Body model preparation

We start from a body model that is described by a polygonal mesh, which can be animated using skeletal deformation [19]. The first step in body model preparation is to define the attachment of the skin surface to the joints of a skeleton, i.e. which vertex belongs to which joint. In our current implementation, this is done using an external application [20]. This attachment information is later used for skeletal deformation. Next, the skin mesh is segmented using the weight values defined in the attachment information. Each triangle belongs to the joint corresponding to the highest weight. During the segmentation, vertices that are located on the boundaries of segments are duplicated. They share the same attachment information. At the end, each segment is attached to the corresponding joint on the H-ANIM skeleton [21]. The result of the segmentation is an H-ANIM compliant human body model.

4.2 Skeletal deformation

The skeletal deformation uses vertex weights. Vertex weights are what tie the joints and the mesh together. Each vertex in the mesh is tied to any number of joints. Each vertex-joint relationship has a scalar weight. The position of vertices is calculated using the weight values and the transformation matrix of the joints. The skeletal deformation is also used to compute the normal to the mesh surface. With the Cosmo3D library [24], these normals need to be updated whenever the mesh is deformed.

On the skin mesh, the movement of vertices that belong to a single joint is not calculated but automatically moved as they are attached to the joint. Duplicated vertices that are on the boundaries have the same position as they share the same attachment information. Thus, the boundaries among segments are not visible. The result is a segmented body that appears as a seamless body in the rendering view port. This method combines the speed of the deformation of segmented bodies with the visual quality of seamless bodies.

4.3 Garment design

The garment creation is done using our in-house software [22]. The garment designer is assisted in drawing 2D patterns and defining seaming lines on the borders of the garment patterns, referring to the polygon edges that are to be joined during the garment construction process. The patterns are then tessellated into a triangular mesh and are placed around the 3D virtual body (Figure 9(a) on color section). Next, the remaining garment shape is computed, as illustrated in Figure 9(b) on color section. The shape of the body model guides the surface of the cloth as a result of the collision response.

4.4 Segmentation of the cloth

As described in the previous section, we use a multi-layer approach. The cloth is segmented into one of the three layers. Each layer contains vertices that belong to the same behaviour category. These cloth layers are simulated using different methods. From the garment in its rest shape on the initial body, the distance between the garment and the skin surface is used to determine to which category the cloth triangles belong. Associated with each segment are distances from the skin surface that are used to determine the category. Each segment falls into one of three categories: tight, loose and floating clothes. Cloth vertices that are located closely to the skin surface belong to the first or the second layer. Cloth vertices that do not collide with any skin surface belong to the third layer (Figure 9(c) on color section). Once the automatic segmentation is done, the user can adjust it if necessary.

When the garment is composed of several layers, our simulation method severely limits its target and only the outermost garment is animated. As soon as the garment segmentation is completed, hidden layers and the skin parts that are covered by the outer layer are automatically removed. Figure 9(d) on color section shows the body model after such optimization on the initial model illustrated in Figure 9(b) on color section. Since they will never be visible during the real-time simulation, their deformation does not need to be computed.

5 Layer 1: "Stretch clothes"

Tight clothes keep constant distances with the underlying skin surface. The deformation of this layer follows the deformation of the underlying skin. We choose to use the skin deformation method to animate this layer. This method does not involve any collision detection or physical deformation. It has no impact on the calculation time. Therefore, it is necessary to construct the skeletal information of this cloth layer. This information is defined by mapping the attachment information of the underlying skin to these cloth vertices. Each vertex of the garment mesh is associated to the closest triangle, edge or vertex of the skin mesh.

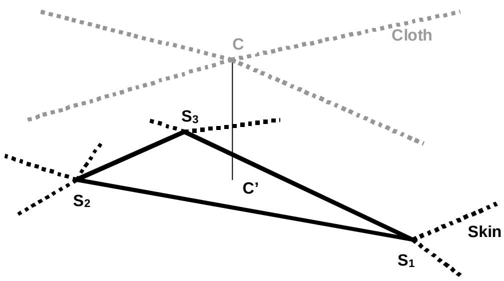

Figure 1: *Mapping of attachment information*

In the Figure 1, the garment vertex C is in collision with the skin triangle $S_1S_2S_3$. We define C' as the closest vertex to C located on the triangle $S_1S_2S_3$. We then define the barycentric coordinates of C' with S_1, S_2 and S_3. The attachment information of C is calculated by combining the attachment information of S_1, S_2 and S_3 weighted with the barycentric coordinates.

6 Layer 2 "Loose cloth"

For loose clothes, the relative movements of clothes to the skin remain relatively small, keeping a certain distance from the skin surface. To get an intuitive understanding of such cases, consider the movement of sleeve in relation with the arm: for a certain region of the garment, the collision area falls within a fixed region of the skin surface during simulation. With this in mind, the scope of the collision detection can be severely limited. A basic assumption made is that the movement of the garment largely depends on that of the underlying skin and yet it should not follow the skin surface rigidly. It is necessary to simulate the local displacement of the garment from the skin surface.

Two different methods have been developed, one for cloth deformation on the limbs (trousers and sleeves), the other one for the deformation of cloth on the trunk.

6.1 Sleeves and trousers

In this approach, we use the assumption that cylinders can approximate the limbs. Each vertex is kept on a disc that is attached to the skin.

- Initially, the position of the disc centres are calculated using the attachment information defined in the pre-processing stage. They are obtained by mapping the attachment information of the underlying skin to the cloth vertices (see Section 5). The axe of the discs is parallel to the limb joint.
- During simulation, the movement of the vertices on their disc follows the equation of the rigid body motion. Vertices move independently to each other. Using the 2^{nd} Newton's law, velocity and position are easily calculated with gravity force. In case a vertex leaves its disc, a kinematic correction [2] is applied to the velocity and the position to put the vertex back on the disc surface. The Figure 2 shows a cross-section of a limb with a garment.

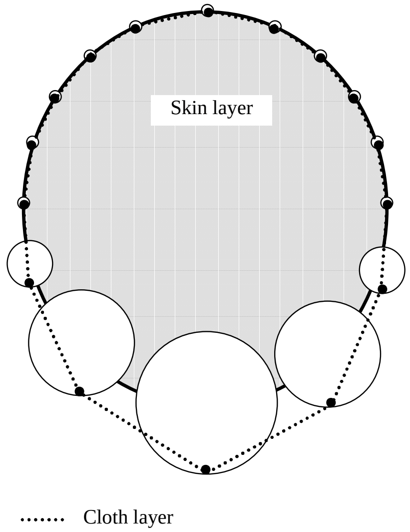

Figure 2: *Cross section of a limb with a garment*

The size of the disc is a function of the angle between the cloth acceleration vector and the normal to the skin surface. The cloth acceleration vector is the subtraction of the gravity to the acceleration of the associated skin surface. This function is defined in a way that the size of the disc follows roughly the catenary shape, i.e. the shape of a hanging wire. It can be proven that if a heavy flexible cable is suspended between two points, then it takes the shape of a curve with the equation:

$$y = c + a\cosh\left(\frac{x}{a}\right).$$

Figure 2 gives an example of the variation of the size of the disc along the body surface.

We approximate the limbs (arms, legs…) to cylinders and then in order to determine the size of the discs for the cloth deformation we consider the equation of the circle (the cylinder that approximates the limb) and the equation of the catenary (the garment holt by the limb) as shown on the graph in Figure 3.

The equation of the half disc of diameter *2r* in Cartesian coordinate system (x, y) is:

$$y = \pm\sqrt{r^2 - x^2}\ .$$

The equation of the catenary in the same coordinate system (Figure 3):

$$y = a\left(\cosh\frac{x}{a} - \cosh\frac{r}{a}\right)$$

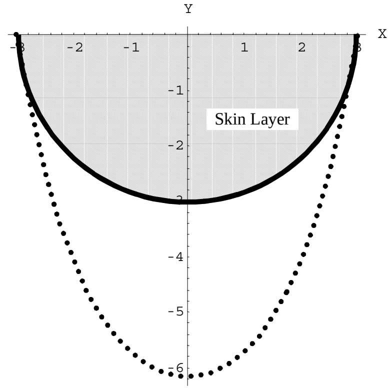

Figure 3: *Equation of the catenary and the disc*

where *2r* is the diameter of the limb and *a* is the scaling factor of the catenary curve. It is used to control the size (largeness) of the garment. We define the ideal size of the disc as the difference of the two equations above:

$$a\left(\cosh\frac{x}{a} - \cosh\frac{r}{a}\right) - \sqrt{r^2 - x^2}$$

We define a quadratic equation to approximate the size of the disc. This equation should be fast to compute, as it will be used intensively to calculate the movement of every vertex at every frame. This approximation is done by fitting the polynomial function *P(x)* to the three points P_0(x=-r), P_1(x=0) and P_2(x=r).

$$P(x) = C\left(\left(\frac{x}{r}\right)^2 - 1\right) \text{ where } C = a\left(1 - \cosh\frac{r}{a}\right) + r$$

In this equation, *C* is a constant that is related to the size (largeness) of the cloth. This parameter is defined by the user at the pre-processing stage. The term $\left(\frac{x}{r}\right)^2$ is calculated with $\sin\theta = \frac{x}{r}$.

$\sin\theta$ is evaluated by projecting the acceleration component $\overline{OA}$ to the normal $\overline{ON}$ of the skin surface as shown in Figure 4. $\overline{OA}$ is the acceleration applied to the cloth. It is the subtraction of the acceleration of the skin vertex *O* to the gravity component.

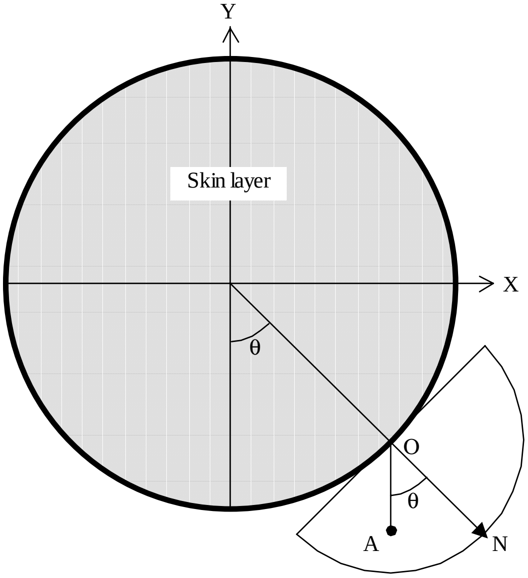

Figure 4: *Computation of the disc diameter*

This method ensures that the rest shape of the garment takes the shape of real garment hanged on a limb. All the discs containing the vertices are parallel to their corresponding limb joints. This ensures that every vertex of the same limb will fall down to the same side on the limb, even in case the direction of the limb is almost vertical. If the limb is perfectly vertical, the result is unpredictable. However, this case never happens in practice.

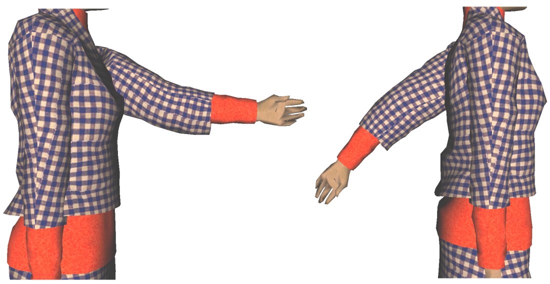

Figure 5: *deformation of sleeves*

In this method, no collision detection on surfaces is necessary. Our particle system is very simple, each vertex moves independently. Therefore, the acceleration applied on vertices is constant. The stability of the system does not depend on the time step duration. The resulting motion of the garments provides the appearance of dynamic movements as well as the shape of real clothes. An example of a sleeve deformation is shown on Figure 5.

6.2 Clothes on the trunk

The hybrid technique described in Section 6.1 works well if the skin surface can be approximated by a moderately thin cylinder and the movement of the garment is more or less perpendicular to the skin surface. This assumption is true for the arms and legs; the friction prevents the garments to move along the limbs. In such cases, vertices are considered to move independently of each other. However, such an assumption cannot be made for garment regions around the trunk where, unlike the arm, the movements of the garment are driven not only by the torso but also by the potentially complex movement of the shoulder and the arm. To get an intuitive understanding of the problem, consider a shirt worn by a virtual human who is raising its arms. The whole garment around the torso part moves along the vertical direction as the arms are raised. This implies the necessity to simulate the elastic property of the tissue. Therefore, a dedicated method for cloth deformation on the trunk has been developed.

A simplified mesh composed of 42 nodes is animated using a simplified mass-spring system. The movement of these nodes is driven by two interactions: gravity and springs in connection with neighbouring nodes. Each of these nodes is a control point used to deform the cloth mesh on the trunk. Figure 10 on color section shows an overview of the simplified mesh. These control points are uniformly placed on the trunk, they form a grid of 3x4x3 points. Six additional control points are attached to the shoulder. These control points maintain the grid mesh on the trunk and prevent the mesh from falling due to the gravity. Each control point is included in a half-sphere that is attached to the skeleton. The collision detection algorithm verifies if each control point is contained within its corresponding hemisphere. In case the control point is not in its hemisphere, a kinematical correction is applied on the position and the speed.

The cloth mesh is deformed with the Freeform Deformation method [26] using the position of the nodes of the simplified mesh. Our implementation uses a 3x4x3 control point lattice for the FFD and uses Bernstein polynomials as the basis functions. The example in shows the grid, composed of the 36 green nodes. The weight values associated to the vertices, deformed by the FFD, are pre-calculated.

7 Layer 3 "Floating cloth"

Layer 3 is composed of vertices that freely float around the body. This will take care of cases, such as a large skirt floating around the legs. Any part on this skirt can collide with any part of the leg. The simulation of this layer uses a classical approach with particle system and collision avoidance.

- **Particle system**: We use a simple mass-spring system. The simulation is performed using the Implicit Euler Integration proposed by Barraf et al. [5] and Volino et al. [1]. The garment is modelled by a simple mass-spring system. We consider two interactions: gravity and forces applied by the springs joining the particles.
- **Collision response**: Calculating collision detection between the cloth and the skin mesh would not be feasible in real-time, therefore we use a simplified model of the body. Given the assumption that the floating clothes are mainly skirts, the collision detection is calculated for the legs only and two cylinders model each leg. By simply calculating the distances between the skirt and the four leg segments, the collision detection can be performed. It is possible to further optimize by restricting the number of collision distances that we compute for each segment. During the pre-processing stage, a list of possible colliding vertices is defined for each segment. This list is defined by calculating the distance between the vertices and the legs and the normal orientation for each vertex on the skirt. The collision response consists of applying a kinematical correction to the position and velocity of the colliding vertices.

8 Deformation on layer boundaries

Vertices located at the boundaries of layers need to be animated with a hybrid method in order to keep a smooth deformation on these boundaries.

On the layer 3, vertices that belong to the boundary are used as anchors to keep the floating garment on the hips. These vertices are solely deformed by the neighbouring layer. This implies that the layer 3 is deformed at last because it uses the resulting deformation of the neighbouring layer.

For other cases, we used a hybrid deformation where the position of the vertices is calculated with the two deformation methods of the layers. The two positions are then summated with associated weights. The weight coefficient defines how much a vertex belongs to a layer. This method provides smooth surface interpolation among layers.

9 Surface tessellation

Due to the time constraint, the maximum number of polygons used for simulating the garments is limited. In order to improve the visual quality of clothes, we have implemented a tessellation algorithm that replaces flat triangles with curved patches and higher-order normal variation. After calculating the deformation of the three layers, each triangle is decomposed into three sub-triangles. The positions of the newly created vertices are calculated using the approach described in [25]. Surface tessellation is mainly used for the third layer where the speed simulation is critical. During the pre-processing stage, the user can define surfaces that will be tessellated during the real-time animation.

10 Results and discussion

Our method has been tested on several dressed virtual humans. Figure 6 illustrates how computational time changes with an increasing number of cloth triangles. The code was executed on a 1GHz PC with 512 MB RAM and a GeForce2 graphics card. The computational times do not include the deformation of the avatar skin and the real-time rendering. A simulation step corresponds to 0.04 seconds worth of real-time animation.

The deformation of Layer 3 is in average thirteen times slower than the deformation of the Layer 2. This is because Layer 2 uses an optimized model with a simplified collision detection method and no integration method. The deformation of Layer 1 is even faster because it uses skeletal deformation. The Layer 3 algorithm is able to simulate a maximum of 1,000 non-tessellated polygons in real-time. In most cases, this is sufficient to produce aesthetically pleasing results. Another advantage is that most of the clothes can be animated using Layer 2. Layer 3 is used in only a limited number of cases, such as large skirts or large trousers.

Figures 11, 12 and 13 on color section demonstrate three examples of real-time animations. These simulations (rendering, cloth and skin deformations) ran at a speed of 25 to 30 frames per second (fps).

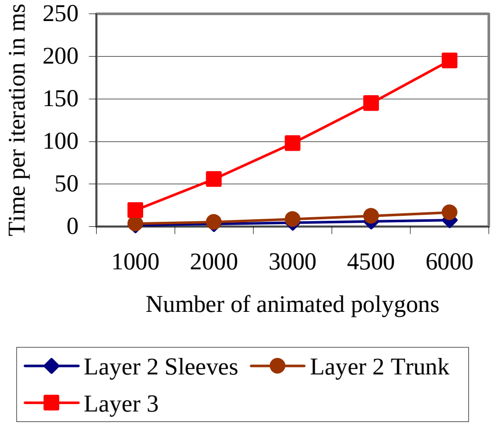

Figure 6: *Computing time per iteration versus number of triangles.*

The Figure 8 summarizes the performances of these three animations.

Model Type / Nb. of Polygons	Skirt and Jacket	Dress	Trouser and Pullover
Original model Cloth & Body	8413	5887	7445
After optimization Cloth & Body	4874	4526	5187
On Layer 1	542	847	2697
On Layer 2	1505	0	0
On Layer 3	618	885	593
Performance (fps)	29	26	25

Figure 7: *Performance table*

Our approach has demonstrated that simulating full-dressed virtual humans with middle-range PC is feasible. Such approach has certain number of limitations due to its optimization.

No self-collision is calculated. In addition, collision detection is restricted to garments and their associated body segment: right sleeve with right arm, left sleeve with left arm, and lower body with dress… With such approach, simulating the hand grasping a dress is not possible. This approach is suitable for any movement that does not involve body-cloth interactions such as grasping clothes.

Another limitation of the proposed approach is its physical model for the layer 2. The cloth deformation on the layer 2 does not define any interaction among vertices (see section 6). Vertices move independently to each other. Their movements are restricted to the perpendicular plan to the limb. This approach does not allow simulating the tensile behaviour of garments and therefore wrinkles.

11 Conclusion

In this paper, we proposed an approach that is able to animate in real-time dressed virtual humans. Calculating all potentially colliding vertices with physical simulation is simply not feasible in real-time. We used the fact that not all the cloth vertices need to be animated with a versatile physical method. Our approach is to partition the clothes into layers and use for each of these layers the most appropriate method that fulfils the minimum requirement: the visual aspect. Since the proposed method uses the deformation of the underlying skin, this method is applicable only on top of deformable objects such as virtual human. Nevertheless, our principal goal is real-time performance. This method has many potential applications anywhere animation of dressed virtual humans is required: Virtual Try-On, Web applications and games, etc.

Acknowledgements

This work is a part of the European project Melies (Modelling Equipment for Live Interactive Electronic Streaming, IST-2000-28700). Many thanks to Chris Joslin, Hyewon Seo and Neeharika Adabala for reviewing the paper, Pascal Volino for his help and ideas and Marlène Arevalo and Christiane Luible for providing the 3D garments and body models.

References

[1] Pascal Volino, Nadia Magnenat-Thalmann, "Comparing Efficiency of Integration Methods for Cloth Animation", Proceedings of CGI'01, Hong-Kong, IEEE Press, July 2001.

[2] Pascal Volino, Nadia Nadia Magnenat-Thalmann, "Implementing fast Cloth Simulation with Collision Response". Proceedings of CGI'00, IEEE Press, June 2000.

[3] Pascal Volino, Nadia Magnenat-Thalmann, "Accurate Collision response on polygonal Meshes". In Computer Animation 2000, Annual Conference Series, IEEE Press, May 2000.

[4] B. Eberhardt, A. Weber, W. Strasser, "A Fast Flexible Particle-System Model for Cloth Draping", Computer Graphics in Textiles and Apparel (IEEE Computer Graphics and Applications), IEEE Press, pp 52-59, Sept.1996

[5] David Baraff and Andrew Witkin. Large steps in cloth simulation. In Michael Cohen, editor, SIGGRAPH 98 Conference Proceedings, Annual Conference Series, pages 43–54. ACM SIGGRAPH, Addison Wesley, July 1998

[6] Mark Meyer, Gilles Debunne, Mathieu Desbrun, Alan H. Barr. Interactive Animation of Cloth-like Objects in Virtual Reality. Journal of Visualization and Computer Animation 2000, John Wiley & Sons.

[7] Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Interactive Animation of Structured Deformable Objects. In Graphics Interface'99 proceedings, pages 1–8, June 1999

[8] Y. M. Kang, J. H. Choi, H. G. Cho, D. H. Lee, C. J. Park, "Real-time Animation Technique for Flexible and Thin Objects", WSCG proceedings, pp 322-329, 2000

[9] T. Vassilev, B. Spanlang, Fast Cloth Animation on Walking Avatars, Eurographics, Blackwell Publishers, September 2001.

[10] Radek Grzeszczuk, Demetri Terzopoulos, and Geoffrey Hinton. Neuroanimator: Fast neural network emulation and control of physics-based models. SIGGRAPH 98 Conference Proceedings, Annual Conference Series, pages 9–20. ACM SIGGRAPH, Addison Wesley, July 1998

[11] Doug James and Dinesh Pai. Accurate real-time deformable objects. SIGGRAPH 99 Conference Proceedings, Annual Conference Series, pages 65–72. ACM SIGGRAPH, Addison Wesley, August 1999

[12] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan Barr. Adaptive simulation of soft bodies in real-time. In Computer Animation 2000, Annual Conference Series, IEEE Press, May 2000.

[13] J. Weil, The synthesis of Cloth Objects, Computer Graphics, SIGGRAPH 86 Conference Proceedings, Annual Conference Series, Vol. 20, pp. 49-54. ACM SIGGRAPH, Addison Wesley, 1986.

[14] T. Agui, Y. Nagao, and M. Nakajma, An Expression Method of Cylindrical Cloth Objects-An Expression of Folds of a sleeve using Computer Graphics, Trans. of Soc. Of Electronics, Information and Communication, Col. J73-D-II, No. 7, 1990, pp1095-1097.

[15] B.K. Hinds and J. McCartney, Interactive Garment Design, Visual Computer, Springler-Verlag, Vol. 6, 1990, pp. 53-61.

[16] H. Ng and R.L. Grimsdale, GEOFF-A Geometrical Editor for Fold Formation, Lecture Notes in Computer Science Vol. 1024: Image Analysis Applications and Computer Graphic, R. Chin, et al., eds., Springer-Verlag, Berlin, 1995, pp124-131.

[17] Y. M. Kang, J. H. Choi, H. G. Cho, D. H. Lee, An efficient animation of wrinkled cloth with approximate implicit integration, The Visual Computer Journal, 2001. Springer-Verlag.

[18] M. Oshita, A. Makinouchi, Real-time Cloth Simulation with Sparse Particles and Curved Faces, Computer Animation, IEEE Press, 2001.

[19] Jason Weber, Run-Time Skin Deformation, In: Proc of Game Developers Conference 2000. http://developer.intel.com/ial/3dsoftware/mrm.htm.

[20] BonesPro, a plug-in to 3DS Max, http://www.digimation.com/

[21] H-Anim specification: http://www.h-Anim.org

[22] P. Volino and N. Magnenat-Thalmann, Virtual Clothing – Theory and practice, Springer-Verlag, 1998.

[23] Lander, Jeff, Skin Them Bones: Game programming for the Web Generation, Game Developer Magazine, May 1998, pp 11-16.

[24] OpenGL Optimizer and Cosmo3D libraries: http://www.sgi.com/software/optimizer/

[25] A. Vlachos, J. Peters, C. Boyd, J. L. Mitchell, Curved PN Triangles, 2001 ACM Symposium on Interactive 3D Graphics, ACM Press, 2001.

[26] A. H. Barr. Global and local deformations of solid primitives. SIGGRAPH 84 Conference Proceedings, Annual Conference Series, pages 21–31. ACM SIGGRAPH, Addison Wesley, July 1984.