3D Clothes and Fashion Show

Nadia Magnenat Thalmann, Pascal Volino, Ying Yang [1]

MIRALab, University of Geneva
24 rue du Général-Dufour
CH 1211 Geneva 4

Shen Jianhua, Daniel Thalmann

Computer Graphics Lab
Swiss Federal Institute of Technology
CH 015 Lausanne
Switzerland

Abstract

In this paper, we first describe the State-of-the-Art in cloth modeling and animation, followed by our first approach to modeling a skirt, then any dress, for a fixed synthetic actress. In this system, clothes were assembled from panels directly on the actress. We then describe a more general system able to create autonomous clothes which may be worn by any synthetic human. We explain the method for creating both the clothes and the virtual humans.

keywords: cloth modeling and animation, body modeling, collision detection, collision response

1. Introduction

The idea behind this paper is to show the evolution of cloth modeling and animation with the ultimate objective of simulating virtual actors able to dress/undress themselves.

In most computer-generated films involving virtual humans, clothes are simulated as a part of the body, and have no autonomous motion. However, in recent years, software has been developed and applied to the interactive design of 2D garment panels and to optimizing the layout of garment panels on fabric. In Hinds and McCartney's work[1], a static trunk of a mannequin's body is represented by bicubic B-spline surfaces. Garment panels are considered to be surfaces of complex 3D shapes. The garment panels are designed around the static mannequin body, and then are reduced to 2D cutting patterns. This approach is contrary to the traditional approach to garment design. The garment is modeled by geometric methods. To visualize the folds and drapes, harmonic functions and sinusoidal functions are superimposed on the garment panels. Mangen and Lasudry [2] proposed an algorithm for finding the intersection polygon of any two polygons. This is applied to the automatic optimization of the layout of polygonal garment panels in 2D rectangular fabrics. Both of these projects concern stages of garment design and manufacturing in real industrial contexts.

For modeling more realistic clothes, two separate problems have to be solved: the motion of the cloth without collision detection and the collision detection of the cloth with the body and with itself.

---

[1] Present address: Alias Research, Toronto, Canada

Previous works on deformable object animation using physically based models have permitted animation of cloth-like objects in many kinds of situations. Weil [3] pioneered cloth animation using an approximated model based on relaxation of the surface. Haumann and Parent [4] produced animations with flags or leaves moving in the wind, or curtains blowing in a breeze. Kunii and Godota [5] used a hybrid model incorporating physical and geometrical techniques to model garment wrinkles. Aono [6] simulated wrinkle propagation on a handkerchief using an elastic model. Terzopoulos et al. [7] developed a general elastic model and applied it to a wide range of objects including cloth.

Deformable objects may be represented by different geometrical models. Triangular grids are the most common, but polynomial surfaces [8] [9] and particle systems [10] [11] are also used for solutions to specific mechanical simulations. Yielding nice and accurate deformations, they constrain both the initial shape and the allowed deformations. Several researchers [12] [13] [14] [15] [16] have also devoted significant efforts to representing and deforming the human body shape.

Collision detection and response has been used mainly for stopping cloth from penetrating the body and, more marginally, for preventing self-collisions between different parts of the cloth. The first time-consuming problem was to extract the possible colliding elements from the whole set of elements composing the cloth and the body surfaces. Many techniques have been developed, based on different ideas and adapted for various surface representations ([17] [18] [19] [20] [21]). Unfortunately, these techniques are not well suited for efficient detection on deformable surface animations, as they require either expensive z-buffer rendering or the construction of the objects' convex hull at each frame.

In Section 2, we describe the history of our dressed virtual actors. Section 3 gives an overview of the evolution toward our new software including the improvements. The new software itself is described in Section 4. We emphasize the innovative aspects: the mechanical simulation engine, collision detection, and human body construction. As the interface is a key aspect of our new system, it is presented in detail in Section 5.

2. Our Physics-based Approaches

2.1 Flashback: the animation of a skirt

In our first approach [22] to producing a synthetic actor with animated clothes, we considered a decomposition of the actor into sections, where each section is dressed separately. On selected sections, a cloth was positioned, endowed with a rectangular discretized grid. This required the specification of body curves where the cloth must be attached (e.g. waist, shoulders). These curves were generated by a modeler, positioned in a section, and then numbered. For example, we created Marilyn's skirt using a closed curve composed of equidistant points (waist). The figure was obtained by cutting Marilyn's body at the waist level and filtering the resulting curve to obtain equidistant points. The skirt was a simple conic shape with given length and angle. For this example, we used a method of collision avoidance that created a very thin force field around the surface of the obstacle to be avoided. This force field acted like a shield rejecting the points. Figure 1 shows an example from the film Flashback.

Figure 1. Collision detection in the film Flashback

2.2 A more general software based on the elastic model

Based on the skirt example, we developed a more general software able to dress a synthetic actress with any cloth. After some comparisons [23], Terzopoulos' elastic surface model [7] was chosen for our system with the damping term replaced by one more accurate. The fundamental equation of motion corresponds to an equilibrium between internal forces (Newtonian term, resistance to stretching, dissipative force, resistance to bending) and external forces (collision forces, gravity, seaming and attaching forces, wind force):

$$\rho(\mathbf{a})\frac{d^2\mathbf{r}}{dt^2} + \frac{\delta}{\delta\mathbf{r}}\iint_{\Omega}\|\boldsymbol{E}\|^2\,da_1\,da_2 + \frac{\delta}{\delta\mathbf{v}}\iint_{\Omega}\|\dot{\boldsymbol{E}}\|^2\,da_1\,da_2$$

$$+\frac{\delta}{\delta\mathbf{r}}\iint_{\Omega}\|\boldsymbol{B}-\boldsymbol{B}_0\|^2\,da_1\,da_2 = \Sigma\,\mathbf{F}_{\mathrm{ex}} \qquad (1)$$

$$\dot{\boldsymbol{E}}_{ij}(\mathbf{r}(\mathbf{a})) = \frac{d}{dt}\,\boldsymbol{E}ij = \frac{1}{2}\dot{\boldsymbol{G}}\,\mathrm{ij} = \frac{\partial\mathbf{r}}{\partial ai}\cdot\frac{\partial\mathbf{v}}{\partial aj} + \frac{\partial\mathbf{r}}{\partial aj}\cdot\frac{\partial\mathbf{v}}{\partial ai} \qquad (2)$$

We choose to replace the third term (dissipative force) because the one used in [7] is scalar. So, no matter where energy comes from, it will be dissipated. For example, gravitational energy is dissipated, resulting in a surface which achieves a limiting speed and so is not continually accelerated. In our case [24], we use Raleigh's dissipative function [25] generalized for a continuum surface [26]. As $\boldsymbol{E}$ is the strain (a measure of the amount of deformation), dE/dt is the "speed" at which the deformation occurs. This means that the surface integral may be considered a rate of energy dissipation due to internal friction. This implies that the variational derivative with respect to velocity of the surface integral will minimize the "speed" of the deformation. With this approach, no dissipation occurs when the surface undergoes rigid body displacement, as when falling in an air-free gravity field. This improves the realism of the motion.

To apply the elastic deformable surface model, the polygonal panel should be discretized using the finite difference approximation method. We have proposed a new algorithm to calculate the elastic force on an arbitrary element. This algorithm is effective for discretizing not only an arbitrary polygonal panel (concave or convex), but also other kinds of polygonal panels with holes inside them.

2.3 The introduction of seaming and attaching forces

In the animation of deformable objects consisting of many surface panels, the constraints that join different panels together and attach them to other objects are very important. In our case, two kinds of dynamic constraints are used in two different stages. When deformable panels are separated, forces are applied to the elements in the panels to join them according to the seaming information. The same method is used to attach the elements of deformable objects to other rigid objects.

After the creation of deformable objects, another kind of dynamic constraint is used to guarantee seaming and attaching. For the attaching, the elements of the deformable objects are always kept on the rigid object, so they have the same position and velocity as the elements of the rigid object to which they are attached. For the seaming and joining of the panels themselves, two seamed elements move with the same velocity and position, but the velocity and position depend on those of the individual elements before seaming. According to the law of momentum conservation, the total momentum of the elements before and after seaming should remain the same.

2.4 A more general collision algorithm

Basically, collisions are detected before a cloth's vertices come through the body's polygons and we must find the position of the point of impact on the polygon, the velocity, and the normal of that point. Moreover, all forces (including internal forces) acting on vertices should be computed.

Although the method works for the simple case of a skirt, the use of this type of force is somewhat artificial and cannot provide a realistic simulation with complex clothes. In fact, the effect degrades when the force becomes very strong, looking like a "kick" given to the cloth.

To improve realism, we have proposed [24] using the law of conservation of momentum for perfectly inelastic bodies. This means we consider all energy to be lost within a collision.

Our user interface [27] allows us to work as a tailor does, designing garments from individual two-dimensional panels seamed together. Figure 2 shows an example of a panel. Figure 3 shows an example of clothes.

Our body modeling and deformation system was built with two layers: a skeleton and an outer skin layer. The envelope is made of polygons. Body deformation was realized by Joint-dependent Local Deformation (JLD) operators [12], which are local deformation operators specific to the nature of the joints. The system has been very successful for modeling and animating several famous synthetic actors, such as Marilyn Monroe and Humphrey Bogart.

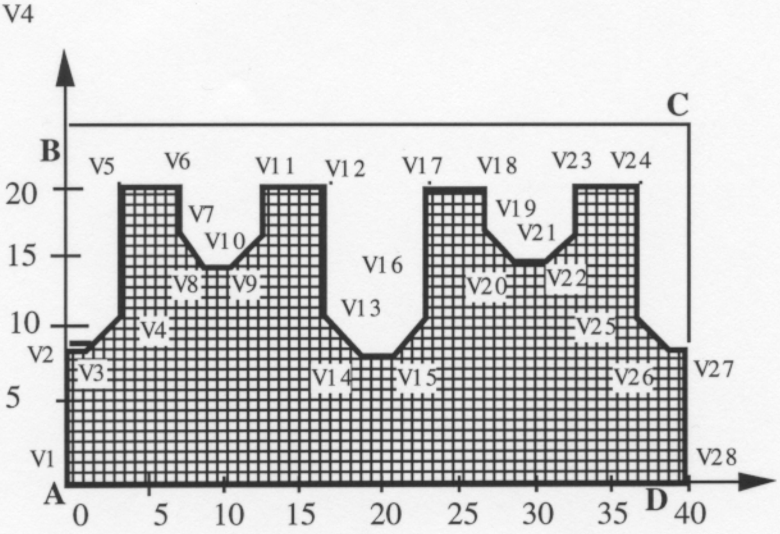

Figure 2. Panel

Figure 3. A synthetic actress with clothes

3. A survey of the new software

3.1 History and evolution

The cloth software described in the previous section was a powerful system that provided very nice realistic results for clothes worn by synthetic actors. It basically opened the way for building clothes and dressing an actor by seaming 2D panels together, in the same way as real-life clothes are built. The panels were assembled around the body using mechanical simulation and the result was a dressed actor that could then be animated.

The 2D panel system has proven to be a very efficient and intuitive way for designing clothes. Furthermore, this approach is close to the process found in the retail industry, where the models are also designed from an assembly of 2D panels. This approach has been maintained in the new clothing system, using the already existing 2D design interface.

However, since the realization of the previous software, animation goals have evolved. Before, focus was put on realistic cloth deformation and appearance on the animated actor. Now, focus is more oriented toward extending the simulation possibilities and animating cloth in very diversified situations, such as clothes worn on several layers, clothes being put on and taken off by an actor, and folded, piled, or crumpled clothes.

The new situations required for such simulations are quite different from those required when an actor simply wears clothes. The model should now be able to handle clothes more like an independent object, rather than a set of panels dependent on the actor's body. Furthermore, several new mechanical situations should be considered, such as high deformation and bending. Collision and self-collision detection should also be very efficient for detecting collisions on the multilayers of crumpled situations.

Generalizing the application range of the software, the simulation engine could be extended for animating a very wide range of deformable objects simply by importing triangular meshes and giving them mechanical properties. Rigid object simulation could also be included (as a new class of animated objects) in the system.

Another of our goals is to dress any type of virtual human. However, in our previous software, the creation of new synthetic actors was a tedious and delicate task, because it required the input of the significant vertices that defined the surface by either 3D digitizing or interactive transformation of existing polygonal models. Another problem was the difficulties in controlling the realistic evolution of the surface across joints. Surface singularities or anomalies could be produced. Simple observation of human skin in deformation reveals that (besides the skeleton configuration) it is very much influenced by internal muscle structure,. Moreover, thousands of vertices are needed to specify a reasonably detailed human body, which is not convenient for storage and communication.

3.2 Improvements of the new software

In order to cope with these new possibilities, the simulation engine, the collision detection procedures, and the modeling of the human bodies have been completely rewritten.

The main concern is to be able to handle clothes as independent objets that do not rely on the underlying body. Thus, the data structure has been completely changed, and any triangular mesh surface, whether regular or not, can be handled by the simulation engine. A garment may still be generated by an assembly of panels, but when the panels are seamed together, it becomes a single and independent object.

Such a clothes model may interact with the body or other clothes using collision response and attach points, but the body may also be removed and the cloth simulated on its own as, for example, when cloth falls down and gets crumpled on the ground.

Generally, crumpling, high deformation and important collision interaction are the primary difficulties encountered in cloth simulation. These problems are dealt with as follows:

- A new collision detection algorithm was used for high efficiency in detecting collisions, particularly self-collisions. This algorithm uses hierarchisation of the surface mesh and takes advantage of curvature regularity within surface regions for determining the possibility of self-collision within these regions [28]. Using such an algorithm, self-collision detection is no longer a time critical process, and can be fully performed for accurate handling of cases including numerous self-collisions, such as crumpling.

- The mechanical model has been modified for robust handling of cases implying high nonlinear deformations resulting from severe elongation and bending, despite a discretisation that may be irregular and rough. Rather than using global integration methods such as finite elements in which nonlinearities and discontinuous behaviors can not easily be integrated, direct integration of Newton's second law was implemented. Several possibilities result from such an approach, including acting directly on position and speed (for direct and interactive manipulation), and integrating highly non linear and time-varying behaviors (high deformations, collision response, stability control) [29].

- Particular attention has been given to the efficiency of collision response. In order to handle in an accurate and stable way complex situations with numerous interacting collisions such as crumpling and multilayer surfaces, response is not computed using force feedback. Instead response is directly performed using position and velocity correction according to the mechanical conservation laws. Thus, we avoid the use of strong discontinuous fields requiring very small time steps for accurate handling, and all collisions can be handled independently of other mechanical computation within the same unaltered time step. An iterative process is then used locally on elements involved in several collisions. With such a technique, we obtain good stability even in complex situations and the global computation time is not severely affected by extra forces and the reduced time step otherwise created by collisions.

- The data structure has been unified between all the different kinds of objects, providing increased simulation versatility. Any object of the scene can be animated as a deformable surface, as a rigid body, as a pre computed animation or as an immobile scene object. All these different objects share the same file format.

- The interface has been completely rebuilt in order to take advantage of all the possibilities brought by the new simulation engine.

Thus, the improvements mainly address versatility and provide the tools for simulating clothes and other deformable objects in many different conditions.

Another innovation of our new software is the possibility of creating any kind of person to wear our clothes. This means that we may dress men, women, and children of various sizes. More generally, our goal is to make efficient and realistic human modeling and deformation capabilities available to the general engineering and entertaining community without the need for physical prototypes or scanning devices. We have developed a highly effective multi-layered approach for design and animation of human bodies. Our layered construction is based on three interrelated levels:

1. The first layer is an underlying articulated ***skeleton*** hierarchy composed of only articulated line segments whose movements are pre-specified.

2. The second layer is composed of grouped volume primitives. These primitives are arranged in an anatomically-based approximation and attached to the proximal joints

of the skeleton. By transforming and deforming those primitives, we can mimic the gross behavior of bones, fat and muscles.

3. The third layer is the skin surface of the body which is automatically derived from the position and shape of the first and second layer.

Our model can be easily integrated into existing environments, producing a body mesh with fixed topology. Since files for implicit models are typically at least two to three orders of magnitude smaller than those modeled with polygon or parametric patches, a compact, parameterized body model, suitable for communication, can be obtained.

4. General description of the simulation software

4.1 The software architecture

The whole software is divided into several parts (see Figure 4):

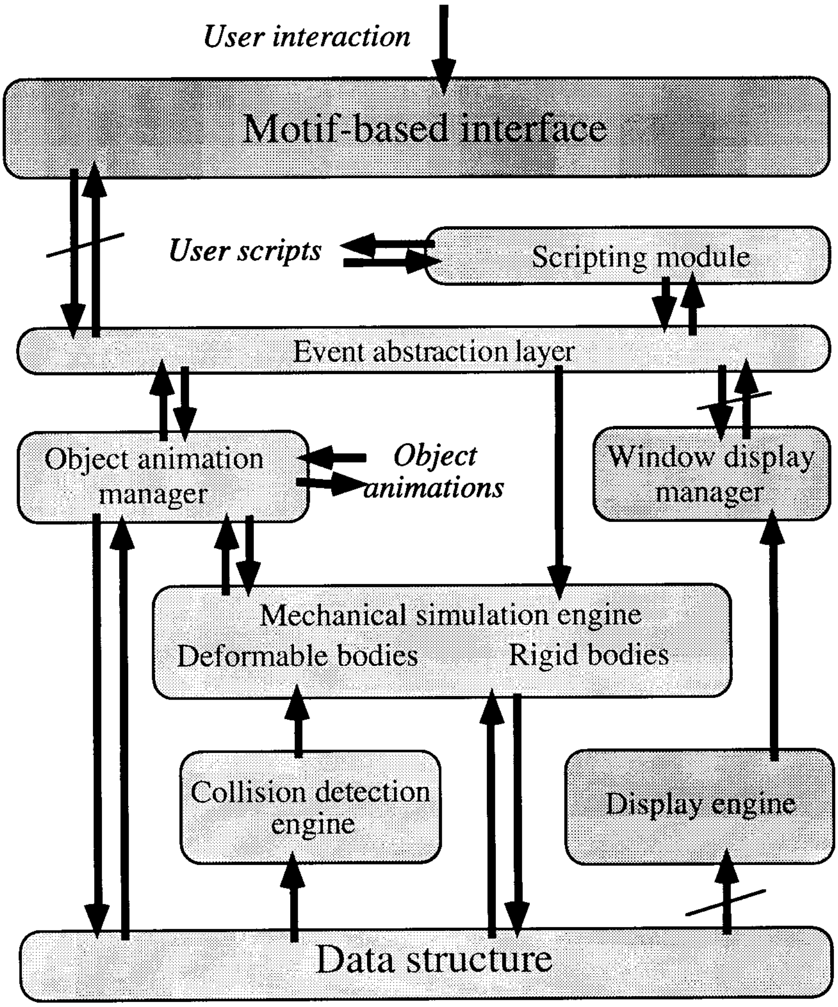

Figure 4. The software modules

- The data structure, storing all object information

- The collision detection engine, which computes the collisions between all the objects in the structure.

- The mechanical simulation engine, which animates objects according to different mechanical models for rigid or deformable objects.

- The software for human body modeling and deformation

- Display modules, handling either window management or object display, which also provide feedback on user graphical interaction (selection, displacements).

- A set of object and event managers, handling all the object attributes and visualization parameters, which processes the interface events and the scripting.

- A motif-based interface, providing the high-level interactive control to the user.

The scripting system allows the program to run without any user interaction. The interface component, at well as the display components, may be suppressed, and the resulting program will run in a complete non-graphical context, allowing, for example, batch-processing on remote sites.

For the whole software, modularity and abstraction has been an important development idea. For example, using event abstraction, an interface system substitution would only require modifying a small number of high-level modules.

The data flow (Figure 5) is basically a computation loop where several animation engines animate their respective types of objects, which are deformable surfaces, rigid objects, animated sequences, transformation-based animations and immobile scene objects. Collision detection is performed for all the objects, and provides collision response for the mechanical simulation engines. The objects are imported from several sources, including animations, static objects in various formats, and cloth panel descriptions. The output is a collection of frames of the computed animation.

4.2 The mechanical simulation engine

The mechanical engine is specially designed for handling deformable surfaces in very severe mechanical situations, such as those encountered in crumpling surfaces. It has to deal with high deformations and nonlinear mechanical behaviors. Despite severe bending and wrinkling, the discretisation has to remain coarse enough to be able to simulate several garments concurrently with reasonable computation time. Furthermore, numerous and interacting collisions resulting from crumpling or multilayer cloth have to be handled efficiently and in robustly.

To comply with all these constraints, the model is based on the direct integration of Newton's second law applied to a particle system model [29]. Such a model allows us to handle each surface element independently, including independent manipulation for simulating position and speed constraints, and linearities such as collision response. Furthermore, such a model allows explicit and simple formulation of any kind of complex and non linear behavior, allowing precise modelling for big deformations.

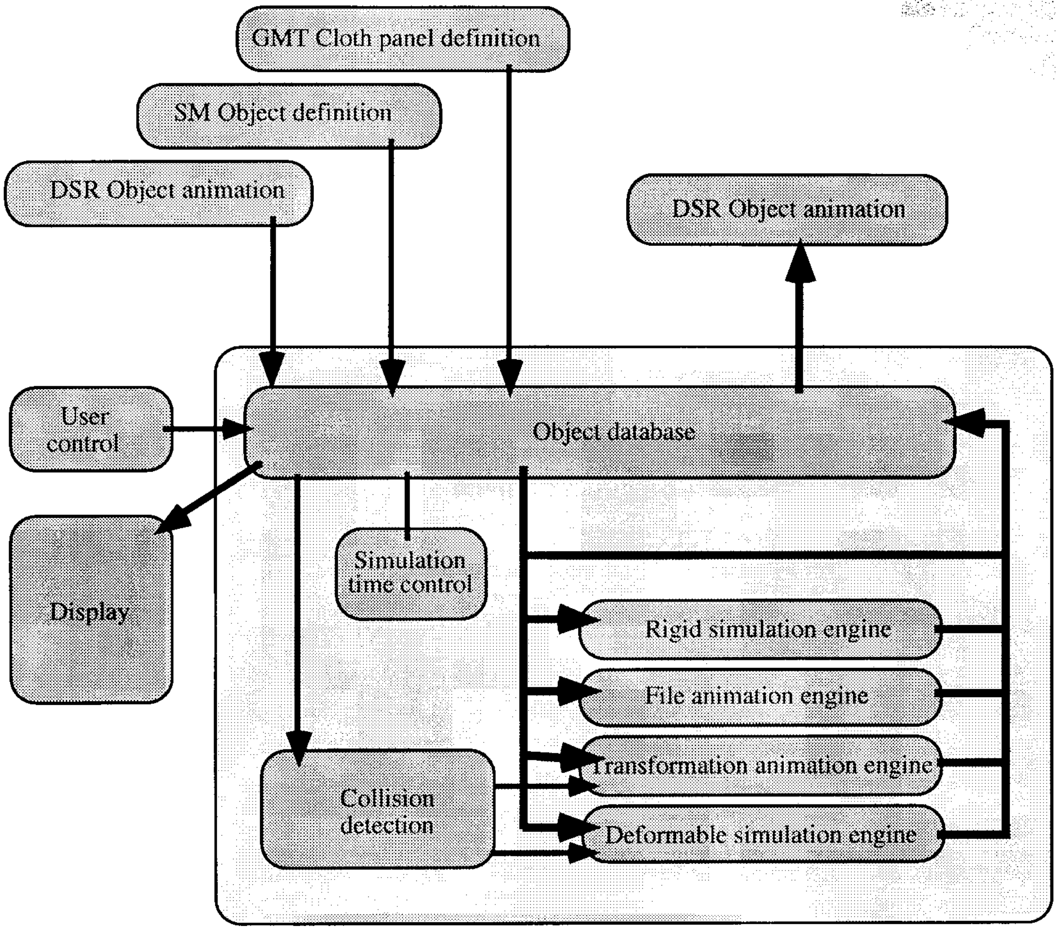

Figure 5. The simulation data flow

Such a model provides particularly efficient collision response processing. After performing deformation calculations without taking into account constrains, a second step performs direct position and speed correction on the colliding elements according to mechanical conservation laws. All the collisions are thus processed independently from the deformation computation, leaving the time step unaltered. By using such a technique, we also avoid the use of strong and discontinuous force fields that alter the simulation efficiency.

Several other facilities have been implemented taking advantage of the model's flexibility. For instance, direct and interactive manipulation is possible as the computation is running. "Elastics" that bring together couples of vertices can be added interactively, providing an efficient technique for adding attachments and seaming lines.

4.3 Collision detection

Collision detection is often the bottleneck of deformable surface simulation programs that handle highly discretized objects, particularly when complete self-collision detection is required, as in wrinkle and crumple situations.

The program takes advantage of an efficient collision detection algorithm, which is based on hierarchisation and takes advantage of surface curvature within and between adjacent surface regions for optimizing self-collision detection [28]. The hierarchisation is performed once at the beginning of the simulation. After this, only bounding boxes and normals are recomputed for all the hierarchy elements and the detection is performed within an element or between two adjacent elements only if their curvature is compatible with the existence of collisions.

Such an algorithm is particularly efficient for huge and regular surfaces, independent of the refinement of the discretisation. In normal cloth simulations, full self-collision detection shows up to be less than 10% of the total collision detection time. This algorithm allows the program to make extensive use of self-collision detection without making any hypothesis about which surface parts may collide, providing a great flexibility on the difficult situations involving complex wrinkling and crumpling.

Because we deal with both dressed bodies and wrinkling, there is no way to define an "inside-outside orientation" for all the simulated surfaces. Thus, it is not possible to say what side should be repelled from the surface in the case of collision. Additionally, some situations may occur where interacting collisions and high deformation cause the collision response to temporarily fail from preventing surface crossover. However, in such case, the system should be robust enough to correct the situation during the next simulation steps. To do this, extra algorithms have been implemented for correcting collision orientations in case of crossover by statistically analyzing the orientations of the neighboring collisions. Such a system allows the simulation to recover from crossover situations, despite the lack of preset orientation information.

Associated with the previously described mechanical simulation engine, this collision detection algorithm preserves its efficiency and robustness in most kinds of situations encountered in cloth simulation.

4.4 The creation of the bodies

As already mentioned, our body representation is based on 3 levels: skeleton, volume (muscles, fat and bones), and skin, as shown in Figure 6.

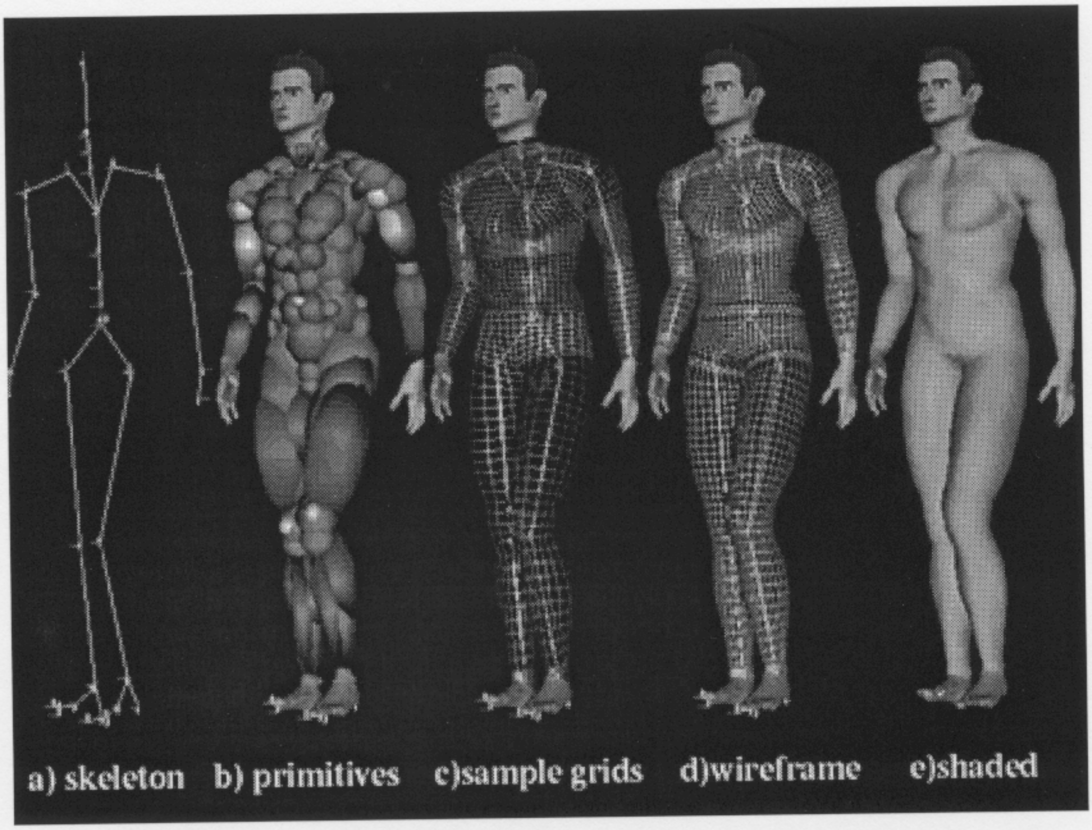

Figure 6 Layered human body model

The volume primitives are based on metaballs, a particular subset of implicit surfaces[30] [31] [32]. An implicit surface is the set of points $\mathbf{x} \equiv (x,y,z) \in \mathbf{R}^3$ such that $f(\mathbf{x}) = 0$. Implicit surfaces are typically defined by starting with simple building block functions and then creating new implicit functions using the sum, min, or max of the simpler functions. They offer the opportunity of modeling and animating complex organic shapes at a fraction of the data points cost compared to more common patching techniques. The final object is constructed by blending the primitives, and as the primitives are moved and deformed the resulting blended surface changes shape.

The volume primitives in our system are categorized into two types: *blendable volume* which will blend with other blendable volumes in the same group; *unblendable volume* which will not blend with other primitives. There is a tradeoff between using complicated volume primitives or more simple ones. For simplicity and efficiency, we currently use only ellipsoids for *unblendable volume* primitives and isosurface with ellipsoidal density distribution for *blendable volumes* primitives. Each *deformable primitive* is associated with *a reference joint*, whose value dynamically determines the center and shape of that primitive. When the underlying skeleton moves, all primitives attached to their relevant joints undergo the joint hierarchy transformations as rigid body motions. Then the deformable primitives change their size, center and orientations according to the current value of their reference joints.

In some sense, the human structure is a sum of different parts. Every part is made up of mass blocks, such as bones, muscles, and fat tissue. We build our human models with the same philosophy. A human body is considered in several parts. In each part, volume primitives are used to approximate the shape of internal structures which have observable effects on surface form. Each primitive is attached to its proximal joint, defined in the joint local coordinate system of the underlying skeleton. Each primitive is assigned to a group in accordance with the body part it contributes to . Some primitives, located near a joint, can fall into several group, as they may have contributions to multiple parts.

To obtain the skin, an implicitly defined surface is sampled with ray-casting on semi-regular cylindrical grids[33]. These sample points are used directly as cubic B-spline control points to smooth out the skin surface. Individual B-spline patches are triangulated, and these triangular meshes are stitched together to connect different parts of the human body for final rendering and output.

In order to automatically generate human models with different sizes and proportions, five normalized parameters (Figure 7) are used to scale the standard skeleton template to accommodate variations in age, sex and race. Body scaling with such parameters is straightforward in our model. Let the skeleton be in a default posture and assume **x**, **y**, **z** axes of the global frame represent *lateral*, *frontal* and *height* direction of the skeleton respectively. We associate a *tag* to indicate the correspondence of each principal direction of a primitive to the frontal, lateral, or vertical direction of the skeleton, and then selectively scale the corresponding axis length and center according to the current skeleton ratio while keeping the same orientation. The tag of an ellipsoidal primitive is established by finding the dominant component of its principle directions in the global frame. That is, if **R** represents one principle direction in global frame, then

$$\mathbf{R} \Leftrightarrow \textit{lateral} \quad \textit{if}\ |\mathbf{R}_x| = \max(|\mathbf{R}_x|,|\mathbf{R}_y|,|\mathbf{R}_z|),\quad \mathbf{R} \Leftrightarrow \textit{frontal} \quad \textit{if}\ |\mathbf{R}_y| = \max(|\mathbf{R}_x|,|\mathbf{R}_y|,|\mathbf{R}_z|),$$

$$\mathbf{R} \Leftrightarrow \textit{high} \quad \textit{if}\ |\mathbf{R}_z| = \max(|\mathbf{R}_x|,|\mathbf{R}_y|,|\mathbf{R}_z|)$$

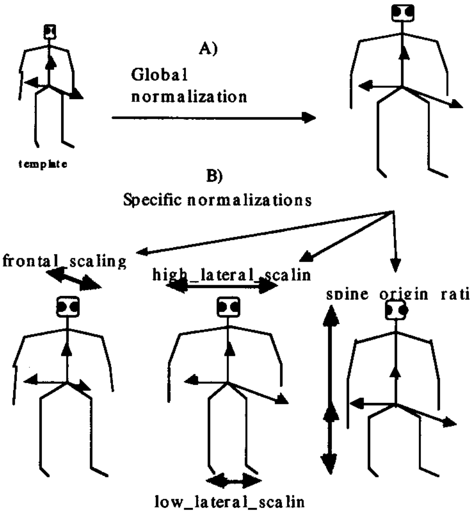

Figure 7 Skeleton scaling

The five scaling operations can be applied successively to automatically generate a variety of human shapes.

5. The interface

The new clothing software is more than just a cloth program designed for dressing an actor. It is indeed a complete and very general tool for performing rigid or deformable mechanical simulation on any kind of object, whether cloth or not.

5.1.1 Interface principle

The software is basically an object animator that simultaneously moves and deforms several objects concurrently with interactions such as collisions and attachments.

The software handles four types of objects:

- Static objects
- Animated objects, moved either by an animation sequence or a geometrical transformation
- Rigid mechanical objects animated by mechanical computation.
- Deformable mechanical objects animated and deformed by mechanical computation.

Each object may be loaded from different sources (various file formats). They are all concurrently animated and collision detection is performed accordingly.

Animation is then run and mechanical computation is performed for the rigid and deformable objects. At each frame instance, the animated objects are saved with their respective frame numbers. User interaction such as modifying parameters or manipulating objects, is possible at any time.

5.2 The cloth design and simulation process

Cloth are animated as any kind of deformable objects. They differ only by how they are constructed, that is by assembling 2D panels. The clothing design and simulation process is divided into two parts:

- The 2D panel design process, consisting of designing garment models as flat fabric panels, using a 2D drawing software [27]. Seams are defined around the borders of the panels.

- The 3D simulation process, that basically consist of assembling the garment panels in the context where the animation will take place using mechanical simulation, and then continuing the simulation on the animated scene and characters.

The scene may contain several objects, static or animated, that will interact with the garments through collision. In particular, the actor to be dressed is typically an animated object.

Garments are loaded from a file containing the description of the 2D panels. The panels are discretized into triangle meshes and then interactively placed in an initial position that is suitable for seaming. When dressing an actor, the initial position is around the actor body. Then, using mechanical simulation, the panels are pulled together along the seaming lines using "elastics". Once the seaming lines are close enough, they are topologically merged, and the set of panels becomes one unique object. Such an object can then be handled independently and relies neither on the former panel definition, nor on the body that supported it when it was built. Figure 8 shows the seaming process.

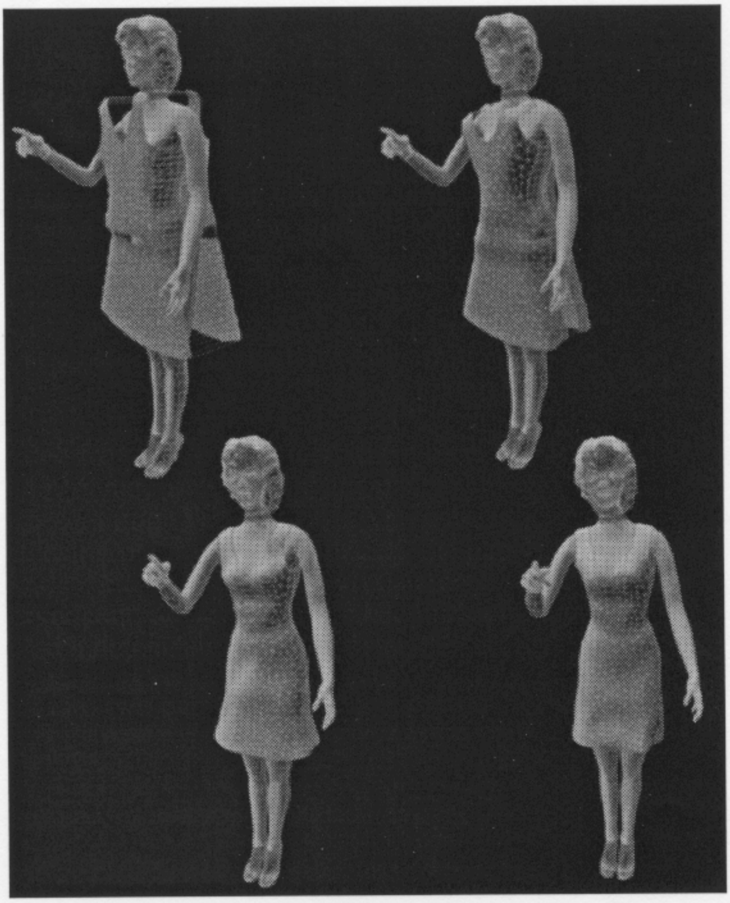

Figure 8. Seaming garment panels around tho body

Once garment is defined in this way, the mechanical computation may proceed with the animated scene for computing the final animation. At any time, extra elastics may be added as attach points within the cloth or between the cloth and other objects, adding new design possibilities.

Complex dressings containing several garments are animated concurrently by the program. Full interaction is provided by collision detection, and optimization for multilayer animated objects provides stability of the overall system. Incremental collision detection is also used when relative movements are slow enough.

All the operations performed during simulation can be recorded in scripts, that are executed for setting up subsequent automatical simulations. Scripts can also be organized into specialized libraries, providing tools for setting up materials, fabric types, simulation conditions,etc.

The animated garments are finally recorded frame by frame as animations, which can be re-used as input data for subsequent computations. This allows incremental garment design for complex cloth. Figure 9 shows an example.

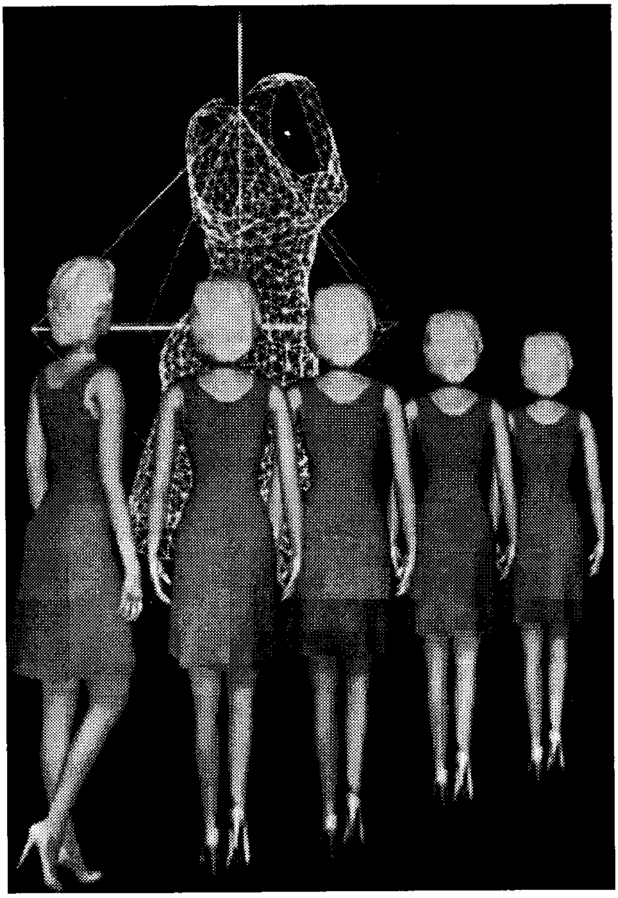

Figure 9. Animation sequence

The interface was designed for providing maximal interaction for controlling the simulation. The user can, at any time, block some points of the animated objects, add or remove elastics, move interactively vertices of the deformable objects interactively, and move whole objects.

5.3 The interface for the creation of top-models

5.3.1 Interactive primitive editor

The implicit surface technique is inherent to interactive design. One can start with a rough shape consisting of just a few primitives, then add details by simply editing these primitives. For example, we can add, delete, transform, or adjust the parameters of primitives. We have written an interactive primitive editor for shape design. The editor can display shaded or wireframe colored primitives and skin surfaces in near real time. Blendable primitives are displayed independently as ellipsoids either with the effective radius or threshold radius. The "threshold" mode shows the visible size of a primitive, while the "effective" mode shows the influence range. Some widget panels are used to interactively adjust the size, weight, center, and orientation of the primitives. A primitive can be switched between the state of blendable or nonblendable, and deformable or nondeformable. Different types of primitives are displayed with different colors. The designer can interactively create, delete, pick, joint attach/detach and group primitives. By turning on/off various display entities of different layers, the designer can selectively check the skeleton, primitives, contours, and skin envelope simultaneously. A Spaceball or trackball enables the user to rotate the model or camera around in space for different viewing. The user can get quick feedback of the resulting skin form during editing. Models can be saved into a file and loaded later for further sculpting.

5.3.2 Body design with proportion box

Users can create new models either from scratch or by modifying existing ones. They begin with specifying the skeleton height (in mm), and do some scaling if necessary. A proportion system is very helpful in 3D design of a full figure. It can guide both the construction of bony sketches and the development of fleshy masses and surface details. Since proportion is concerned with relativity, there must be a unit of comparison. The head-length (from the crown to the tip of the chin) scheme proposed by Dr. Paul Richor [34] is widely adopted by artists. In Figure 10, the height of the figure is equal to 7.5 heads. Different proportions for different genders and ages are also described in [34] (pp216-219). We have incorporated this scheme into our system. It is particularly valuable for designing aesthetic figures. Also, one can efficiently scale these "ideal" models to get excellent individual variation.

The interactive editor, proportion box and body scaling provide a set of intuitive tools for animators to design a rich variety of human shapes. Volumetric primitives enables compact specification of human bodies suitable for communication.

6. Conclusion

In this paper, we have shown how virtual actors could be dressed with autonomous clothes. We first reviewed our early work in this area, and then presented a more general software with a new simulation engine, a new collision detection procedure, and a software for modeling human bodies. We have also shown that we can now create clothes for any kind of virtual actor. Our main concern was to be able to handle cloth as independent objets, that do not rely on the underlying actor body. Such a cloth model may interact with the body or other clothes using collision response and attach points, but the body may also be removed, and the cloth simulated on its own, when it falls and gets crumpled on the ground.

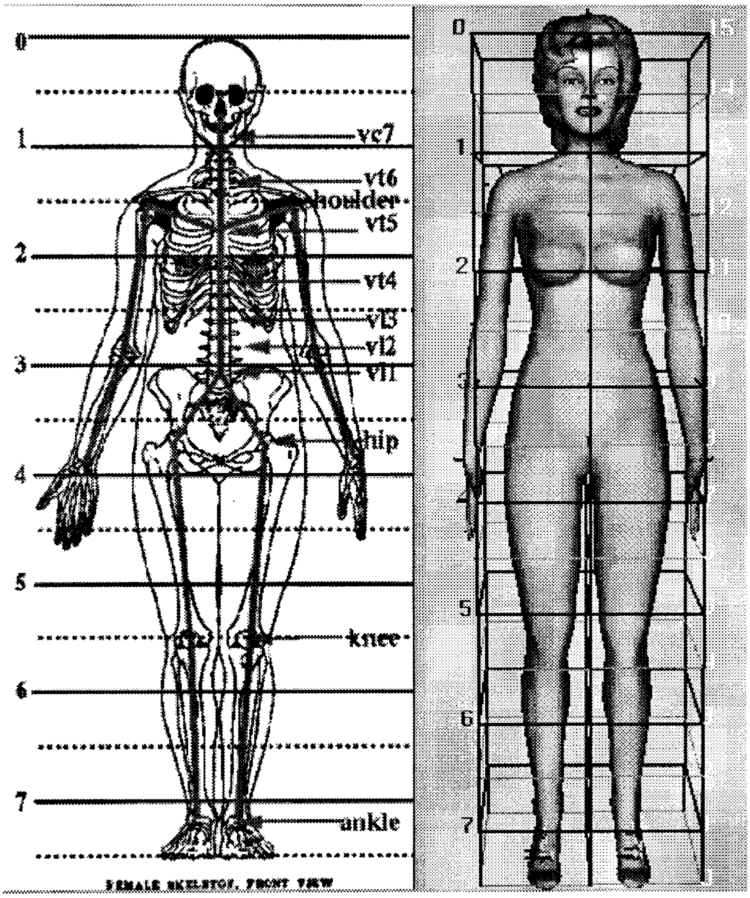

Figure 10. Proportion box mapping on the skeleton

7. Acknowledgments:

The authors are grateful to Mehdi Davary, and Jean-Claude Moussaly for their help in the production of some images. They also thank Marcia Riley for improving the English text. This work is supported by the Swiss National Research Foundation and the European ESPRIT HUMANOID-2 project.

8. References

1 Hinds BK, McCartney J. “Interactive garment design”, In The Visual Computer, 1990, Vol. 6, pp.53-61.

2 A. Mangen, N. Lasudry, “Search for the Intersection Polygon of any Two Polygons: Application to the Garment Industry”, Computer Graphics Forum 10 (1991), pp.19-28.

3 J. Weil, "The synthesis of Cloth Objects", Computer Graphics (proc. SIGGRAPH'86), 4, pp 49-54, 1986.

4 D.R. Haumann, R.E. Parent, "The Behavioral Test-Bed: Obtaining Complex Behavior With Simple Rules", The Visual Computer, Springer-Verlag, 4, pp 332-347, 1988.

5 T.L. Kunii, H.Gotoda, "Modeling and Animation of Garment Wrinkle Formation processes", Computer Animation Proc., Springer-Verlag, pp 131-146, 1990.

6 M. Aono, "A Wrinkle Propagation Model for Cloth", Computer Graphics International Proc., Springer-Verlag, 1990.

7 D. Terzopoulos, J.C. Platt, H. Bar, "Elastically Deformable Models", Computer Graphics, Proc. SIGGRAPH'87, 21, 1987, pp 205-214.

8 A. Witkin, W. Welch, "Fast Animation and Control of Non-Rigid Structures", Computer Graphics, Proc. SIGGRAPH'90, 24, 1990, pp 243-252.

9 D. Baraff, A. Witkin, "Global Methods for Simulating Flexible Bodies", Computer Animation Proc., Springer-Verlag, pp 1-12, 1994.

10 D.E. Breen, D.H. House, M.J. Wozny, "Predicting the Drape of Woven Cloth using Interacting Particles", Computer Graphics (proc. SIGGRAPH'94), 28(4) , 1994, pp 365-372.

11 D. E. Breen, D. H. House, P. H. Getto, "A Physically-based Particle Models of woven cloth", The Visual Computer, 8(5-6), Springer-Verlag, Heidelberg, 1992, pp 264-277.

12 N. Magnenat Thalmann, D. Thalmann, "The direction of Synthetic Actors in the Film Rendezvous a Montreal", IEEE CG&A, 1987, Vol.7, No 12, 1987, pp.9-19.

13 Komatsu, K., "Human skin model capable of natural shape variation", The visual computer (1988)3, pp.265-271

14 D.R. Forsey, " A surface model for skeleton-based character animation", Second Eurographics Workshop on Animation and Simulation, Sept.,1991,Vienna, pp.55-71

15 J.E. Chadwick et.al, "Layered construction for deformable animated character", Computer Graphics, 23, (3), 1989

16 M.P.Gascuel et al., "A modeling system for complex deformable bodies suited to animation and collision processing", The Journal of Visualization and Computer Animation, Vol 2,.1991

17 D. Baraff, "Curved Surfaces and Coherence for Non-Penetrating Rigid Body Simulation", Computer Graphics, Proc. SIGGRAPH'90, 24(4) , 1990, pp 19-28.

18 B. Von Herzen, A.H. Barr, H.R. Zatz, "Geometric Collisions for Time-Dependant Parametric Surfaces", Computer Graphics, Proc. SIGGRAPH'90, 1990, 24(4), pp 39-48.

19 J.M. Snyder, A.R. Woodbury, K. Fleisher, B. Currin, A.H. Barr, "Interval Methods for Multi-Point Collisions between Time-Dependant Curved Surfaces", Computer Graphics annual series, 1993, pp 321-334.

20 M. Moore, J. Wilhelms, "Collision Detection and Response for Computer Animation", Computer Graphics, Proc. SIGGRAPH'88, 22(4), 1988, pp 289-298.

21 M.C. Lin, D. Manocha, "Interference Detection between Curved Objects for Computer Animation", Computer.Animation Proc., Springer-Verlag, 1993, pp 43-55.

22 B. Lafleur, N. Magnenat Thalmann, D. Thalmann, "Cloth Animation with Self-Collision Detection", Proc. of the IFIP conference on Modeling in Computer Graphics, Springer, 1991, pp 179-187.

23 Y. Yang, N. Magnenat Thalmann, D. Thalmann, Three-Dimensional Garment Design and Animation -- A new design tool for garment industry, Computer in Industry, Elsevier, 19, 1992, pp.185-191

24 M. Carignan, Y. Yang, N. Magnenat Thalmann, D. Thalmann, "Dressing Animated Synthetic Actors with Complex Deformable Clothes", Computer Graphics, Proc. SIGGRAPH'92, 1992, 26(2), pp 99-104.

25 Eringen AC, Suhubi, ES. Elastodynamics, Vol.1, Academic Press, NY.,1974.

26 J.C. Platt, A.H. Barr Constraints Methods for Flexible Models , Proc. SIGGRAPH'88 , Computer Graphics, Vol.23, No.3, 1988, pp.21-30

27 H.M. Werner, N. Magnenat Thalmann, D. Thalmann, "User Interface for Fashion Design", Graphics Design and Visualisation, IFIP Trans. North Holland, 1993, pp 197-204.

28 P. Volino, N. Magnenat Thalmann, "Efficient Self-Collision Detection on Smoothly Discretised Surface Animations using Geometrical Shape Regularity", Computer Graphics Forum, Proc.Eurographics ‘94, 13(3) , 1994, pp 155-166.

29 P. Volino, M. Courchesne, N. Magnenat Thalmann, "Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects", Computer Graphics, 29, 1995, pp 137-144.

30 Blinn, J.F. “A generalization of Algebraic Surface Drawing”, ACM Transactions on Graphics. 1, 3, 1982, pp.235-256

31 G. Wyvill, C. McPheeters, B. Wyvill "Data Structure for soft objects", The Visual Computer, Vol.2, No.4, 1986.

32 Bloomenthal,J., Shoemaker, K.,"Covolution Surfaces ", Computer Graphics, Vol 25, no.4, 1991, pp.251-256.

33 J. Shen, D. Thalmann, “Interactive shape design using metaballs and splines”, Implicit Surface’95, Eurographics workshop, Grenoble, France, 1995, pp.187-196.

34 S. R. Peck, “Atlas of Human Anatomy for the Artist", Oxford University Press, 1982.