A System for Dynamic Simulation of Dressed Walking Characters

V. Dochev, T. Vassilev

Key Words: *Virtual clothing; cloth simulation; collision detection; physically based modelling.*

Abstract. *This paper presents a system for dressing virtual humans with dynamic simulation capabilities. The garments are constructed of cutting patterns, imported from apparel CAD systems. Two approaches for editing clothes were implemented and investigated: a standalone programme and a plug-in to Adobe Illustrator. The system produces dressed virtual humans using as an input a body file and a garment file. It is then possible to carry out simulations of walking dressed humans. For the latter a hybrid approach for cloth-body collision detection was developed. It is based on both image- and object-space techniques with the intention to incorporate their advantages. Images of both static and dynamic simulations produced with the system, along with performance data are given at the end of this paper.*

1. Introduction

Considering the quickly increasing performance of the modern computers real-time cloth simulation looks more and more realistic. The main aim of this work is to develop a system for dressing virtual humans and animating them walking on a catwalk. Such a system can have two major applications. First it will be useful for fashion designers, who could immediately „dress" a new fashion design on a 3D virtual body, make it go on a catwalk and view it from a set of different points.

The second application is in the field of e-commerce. An Internet system can be developed, such that the customer using a virtual model of his body can browse different garments in a computer catalogue, try them on, walk in a virtual environment and buy if he/she is satisfied with the choice. In order to develop such a system the following components are needed: 3D model of the human body, physical cloth model for computer simulation, garment description in an electronic form, which supposes a piece of software for editing garments, and efficient and precise algorithm for collision detection.

The rest of the paper is organised as follows: The next section reviews previous work on cloth modelling and simulation, collision detection for the purposes of cloth simulation and garment representation. Section 3 presents a modification of Provot's cloth model with improved super-elasticity handling. Section 4 explains the combined technique for collision detection. Section 5 gives information about garment description and editing. Section 6 presents some results and Section 7 concludes the paper.

2. Related Work

Cloth Model

Methods to model cloth for computer graphics have been investigated for about two decades. Weil [23] was the first to report a model for cloth objects using a two step geometric process. Around 1986, Feynman [12], developed the first physically-based model which was based on elastic shell theory. Since then researchers have found many ways to describe cloth dynamics that differ mainly in their physical accuracy, numerical stability and computational cost. Mass-spring particle systems are mainly used while some [17,21,6,10] employ finite element methods (FEM). Breen et al [5] first employed particle systems to approximate textile behaviour from a Kawabata Evaluation System (KES) [13]. Eberhardt et al [9] improved Breen's model by adding simulation of hysteresis effects obtained directly from a KES. Provot [15] introduced a simple mass-spring topology (see Figure 1) which is commonly used due to its efficiency and simplicity. Provot uses linear (Hook) springs and applies explicit Euler integration.

The elastic model of cloth is a mesh of lxn 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 spring:

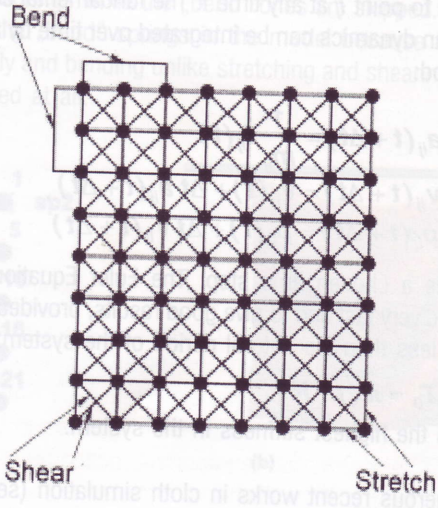


Figure 1. Spring types in the cloth model

- Springs linking vertices $[i, j]$ with $[i+1, j]$, and $[i, j]$ with $[i, j+1]$ are called „stretch" 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 „bend" springs.

As the names indicate, the first type of spring implements resistance to stretching, the second – to shearing and the third – to bending.

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

$$\text{(1)} \quad \mathbf{f}_{ij} = m_{ij}\, \mathbf{a}_{ij},$$

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

Internal forces arising from the tensions of the springs. The overall internal force applied at point ij is a result of the stiffness of all springs linking this point to its neighbours:

$$\text{(2)} \quad f_{int}(p_{ij}) = -\sum_{k,l} k_{ijkl}\left((p_{kl} - p_{ij}) - l^0_{ijkl} \frac{p_{kl} - p_{ij}}{\|p_{kl} - p_{ij}\|} \right),$$

where k_{ijkl} is the stiffness of the spring linking ij and 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 carry out. The most frequently applied forces are:

- Gravity: $\mathbf{f}^{gr}_{ij} = m_{ij}\mathbf{g}$, where $\mathbf{g}$ is the gravity acceleration;
- Viscous damping: $\mathbf{f}^{vd}_{ij} = -C_{vd}(\mathbf{v}_i - \mathbf{v}_j)$, where C_{vd} is a damping coefficient,
- Collision response.

Using the above formulation we may compute the force $\mathbf{f}_{ij}(t)$ applied to point ij at any time t. The fundamental equations of Newtonian dynamics can be integrated over time by a simple Euler method:

$$\text{(3)} \quad \begin{aligned} a_{ij}(t+\Delta t) &= \frac{1}{m_{ij}} f_{ij}(t) \\ v_{ij}(t+\Delta t) &= v_{ij}(t) + \Delta t\, a_{ij}(t+\Delta t), \\ p_{ij}(t+\Delta t) &= p_{ij}(t) + \Delta t\, v_{ij}(t+\Delta t) \end{aligned}$$

where Δt is a chosen time step. The Euler Equations 3 are known to be very fast and to give good results, provided the time step Δt is less than the natural period of the system:

$$\text{(4)} \quad T_0 \approx \pi\sqrt{m/K},$$

where K is the highest stiffness in the system.

Numerous recent works in cloth simulation (see for example [3,7,11]) has shown that improvements in stability are possible by using implicit integration. However, for complex garments with mapping of KES measurements to the spring properties, explicit integration still proves to be beneficial in terms of efficiency in our case [19]. The advantages of Euler integration became particularly apparent when computation of the collision detection and response, which require small time steps, was taken into consideration. Similar results were also indicated by Volino and Magnenat-Thalmann in [22].

There are also commercially available systems for cloth simulation [24]. However, the companies do not describe the simulation methods. Most probably they use some of the approaches and algorithms, mentioned above and published in the scientific journals.

Super-elasticity

Due to its simplicity and effectiveness the above outlined Provot's cloth model has low computational demands, but it has one serious drawback. It is called super-elasticity and it is a result of the ideal nature of the springs used in the model. Most fabric types are far from being ideally elastic and when under stress they have hysteresis behaviour. As a consequence of the super-elasticity in most cases the simulated cloth stretches even under its own weight which has a negative effect on the resemblance of real cloth. Since super-elasticity can drastically reduce the simulation realism, means for counteraction should be considered. One partial solution of the problem is to increase the stiffness of the springs in the system. According to Equation 4 this would result in a decrease of the natural period of the system and would force us to use a smaller time step for keeping the system stable. This in turn would lead to a more costly algorithm due to the increased number of iterations required for the same simulation time which is not desirable.

A better course of action to super-elasticity handling is based on introducing dynamics constraints in the system. To account for super-elongation, caused by the linear springs, Provot [15] constrains particles' positions in a post correction step so that springs can not extend above a certain threshold (usually 5-10% of their natural length, depending on the material properties to be simulated). Vassilev et al [19] improved this by modifying the particles' velocities instead of their positions with the intent to prevent the over-elongation of the springs.

In the above approaches it is either clearly stated that the order in which the points and springs are processed is not taken into account or there are no evidences for a specific order. Usually the order depends entirely on the underlying data structure. As a consequence it is a common case that the adjustment of a spring over-elongates one or more already processed springs and some time is needed for this process to converge. Furthermore in some cases the correction procedure is invoked more than once during a single time step. This is done to speed up the convergence and also with reversal of the springs order for restoration of the balance in the system. The system presented herein employs super-elasticity handling algorithm which corrects the springs when necessary in a specific order. The latter is obtained in advance by the algorithm and is aimed at preventing the over-elongation of already corrected springs and at improving the super-elasticity handling routine in regard to convergence.

Collision Detection

Collision detection proves to be a bottleneck of dynamic simulation algorithms. This is especially true if surfaces are highly discretized owing to the number of potential colliding particles. Most of the existing algorithms [18] for detecting collisions between cloth and other objects in the scene are based on geometrical object-space interference tests. They use geometric calculations to detect penetration between a cloth particle and a face of the object together with strategies to minimise these geometric calculations. Bounding volume hierarchy approaches divide the space into simple bounding volumes such as bounding spheres, axis-aligned bounding boxes (AABB), object oriented-bounding boxes (OOBB) or *k-Dop* volumes [14]. The hierarchies are represented by tree structures which reduce the complexity from $O(n^2)$ of a nanve implementation to $O(n \log n)$, where n is the number of polygons in cloth and object. Additional culling of possible interactions between cloth and cloth can be achieved by taking the surface curvature of cloth into account. The basic idea is that particles close to each other on a low curvature surface are unlikely to collide. Use of surface curvature is described by Volino et al [20] and Provot [16].

By partitioning the space around the cloth into uniformly sized cells (voxels), the complexity of a collision query can be reduced further from $O(n \log n)$ to $O(n)$. This is possible because voxel size and positions are known and reference to them can be made directly. Bigliani et al [4] and Zhang et al [25] report such techniques. In order to reduce memory requirements, they use hash tables for the voxel data.

Vassilev et al [19] use similar ideas as that of the voxel based approaches, described above, but reduce the problem from *3D* to *2 1/2D* depth maps. This approach is an image space based collision detection in which the modern graphics hardware is employed for rasterization and depth map generation for collision tests. Thus. memory requirements are drastically reduced. Furthermore, interpolation units of the graphics hardware are harnessed to compute smooth surface normals for collision response. Vertex velocities of dynamic objects are interpolated in a similar way to allow quick calculation of the response to dynamic collisions. The performance of image space based approaches is independent of complexity of the objects' geometry and therefore particularly well suited for use in combination with highly accurate densely sampled 3D body scans.

Both object and image-space approached have their pros and cons. A common drawback of all object-space based techniques, when used for dynamic simulations and/or simulation of deformable objects, is the necessity of frequent hierarchy updates. The updates are computationally expensive hence simulation speed degrades. In addition, the interference tests in object-space are complicated by nature too. The basic image-space based approach has a serious disadvantage, too, which makes it unusable for animation purposes in scenes with overlapping of objects or parts of objects during the animation. When overlapping occurs it results in loss of essential depth information needed for collision detection.

Another group of image-space based techniques are the so called „depth peeling" algorithms. They do not suffer from the overlapping disadvantage of the basic approach but the process of depth maps generation is computationally expensive which is especially true from in case of complicated 3D scenes.

This work utilizes combined collision detection technique which uses an image-space approach whenever possible because of the efficiency of the latter. In addition it is coupled with object-space based approach which is necessary for handling overlapping and self-overlapping objects. The technique is discussed in more detail in Section 4.

Garment Description

A cloth model such as that described here, is, on its own, not sufficient for garment simulation. While standards to describe garments exist [1,2] they are vulnerable to misinterpretations by different CAD systems, they are not commonly used by designers and do not provide sufficient information for a virtual try-on system. The modern CAD apparel systems export only the geometry of the garment cutting patterns in AutoCAD DXF format which includes no information about seaming, material properties, etc.

3. Cloth Model Modification

The system presented in this work utilizes a modification of Provot's cloth model in regard to super-elasticity handling. It uses a velocity directional modification approach [19] for adjusting the springs but the order in which this is done is carefully chosen. The spring processing order is obtained via algorithm which takes into consideration the setup in the 3D scene. Its idea follows the intuitive concept to process the points (respectively the springs connected) centre of high tension first. The following spring adjustments are then performed in direction towards the high tension centres. This usually minimizes the over-elongation of already corrected springs and eliminates the need to invoke the constraint procedure more than once in a single time step. Subject to adjustment are the „stretch" and „shear" springs while the „bend" ones are skipped. We do not restrict the „bend" springs in the model because real fabrics fold easily and bending unlike stretching and shearing is almost not limited at all.

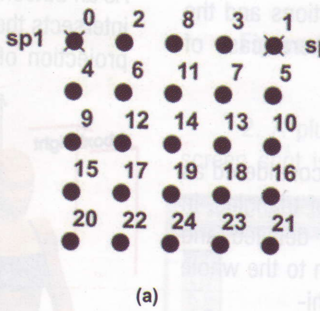


Figure 2. Particular example of point (respectively spring) processing order for super-elasticity handling

The adjustment order of the springs is obtained through series of mass points and springs sort routines. A more detailed explanation of the method can be found in [8].

Figure 2a depicts a particular example of processing order defined by the numbers in the image, for the piece of cloth in the 3D scene in Figure 2b. The springs of the cloth model in Figure 2a are omitted for clarity. Sp (in Figure 2a) denotes a static point.

4. Combined Cloth-body Collision Detection

This technique is an implementation of the idea to utilize an image-space approach whenever possible because of its efficiency, independence of body complexity and hardware support. Furthermore, in order to cancel its drawback the image-space approach is combined with an object-space one. Following this scheme in our case a decomposition of the virtual human model is necessary (*figure 3*). The image-space part of the technique is used for the torso and the head. The reason is that self-overlapping could not occur except in cases of extreme bending for which this approach was not meant initially. The limbs, which most likely obscure or are obscured by the torso or other limbs, are taken care of in object-space.

Image-space Details

The image-space approach is borrowed from [19]. Two depth maps respectively for the front and the back of the torso are needed. They are acquired via two off-screen renderings from the point of view of two orthogonal cameras at the centre of the front and the back face of the body's bounding box (BB). The cameras point at the centre of the BB. An example of a setup for front map acquisition and the respective depth map are shown in *figure 4*. The depth values are floating-point values ranging from 0.0 to 1.0. A value of 0.0 represents a point at the near clipping plane which in this case is set to the front face of the BB (the darker shades in *figure 4b*). A value of 1.0 stands for a point at the far clipping plane set to the back face of the BB (the brighter shades).

The detection of collisions with the help of the depth maps is quite simple. It is carried out through depth values comparison after conversion of the coordinates for each mass point from the cloth to map coordinates. The calculation of the coordinates is simplified due to the use of orthogonal projections and the positions of the cameras in contrast to the general case of projection in screen coordinates.

Object-space Details

In this work the model of a human body is considered as polygonal model, i.e. its surface is constructed of triangles. In addition it is assumed that the model is well detailed and consists of small enough triangles in comparison to the whole body size. This assumption is important for the hierarchy construction stage. Each limb is treated separately of the torso and the other limbs. This results in a decrease of complexity because of the reduction in domain space for partitioning and collision testing. Further simplification is achieved by substituting the more complicated case of deformable surface - body verification with point (cloth vertex) object verification.

In regard to the human model the position of a point from the cloth surface could be acceptable, non-penetrating the body, and unacceptable or penetrating the body. The unacceptable state of a point means that the point belongs to the domain enclosed

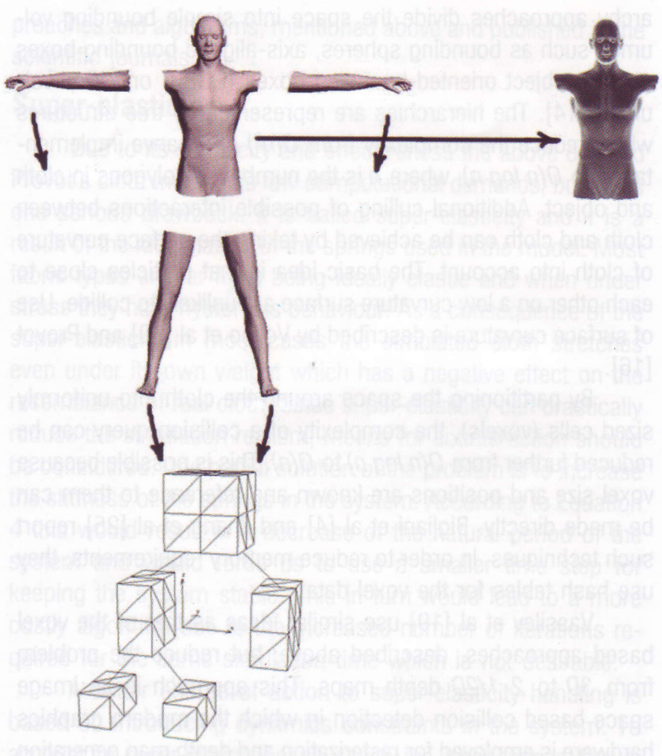


Figure 3. Decomposition of a virtual human model - a prerequisite for applying the combined collision detection

by the surface faces of the human model. This is quite simple from a perceptual point of view but determining the point's state programmatically is quite complex. The strategy followed in this work is to find in the ideal case the closest triangle from the body to the cloth point of interest and then to inspect the positions of the point in two successive simulation steps. The triangle should be in close proximity to both point positions because of the small time steps required for smooth animations. As an outcome of the two tests it is determined whether the point intersects the plane containing the triangle. At the same time the projection of the vector between the respective point and the

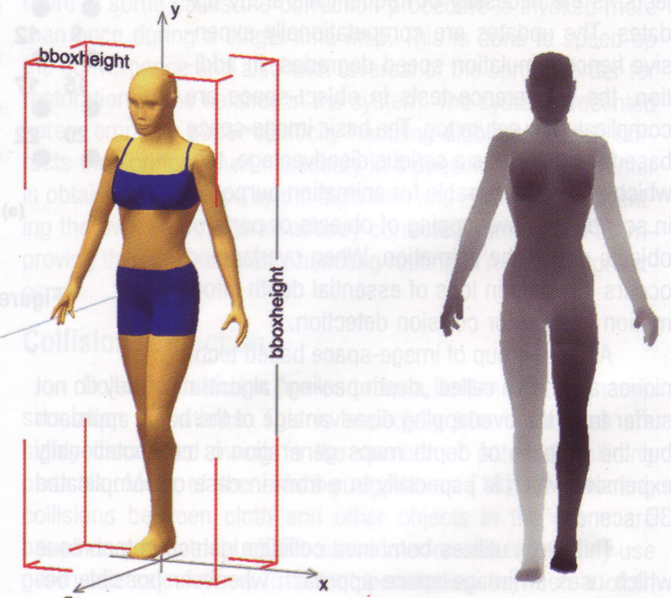


Figure 4. Setup for front depth map acquisition and the respective depth map

origin along the normal vector of the triangle is calculated. This information is useful for computing the distance to the triangle and necessary for the collision response stage.

A few characteristics of the human body are taken into account while considering the hierarchy type which is to be used. Some of them were already mentioned. Another feature influencing the choice of the hierarchy is that the human body and its limbs in particular are rather symmetric. As a consequence the describing triangles are conservatively regularly distributed in relation to the centre of the corresponding BB. Additionally, because of the integrity of the body there are no sparsely occupied spaces.

The above said makes the group of voxel subdivision approaches a good choice. Due to the regularity there is no need for sophisticated subdivision algorithm with criterions about the best split. We simply split the domain of a given limb to halves with a plane parallel to its largest dimension. This strategy proved to produce a balanced hierarchy in most cases. *Figure 5* demonstrates this approach for the entire body of a male virtual human model. Three differently coloured splitting planes along the principal axes, which divide the space occupied by the BB of the body, are shown in the figure.

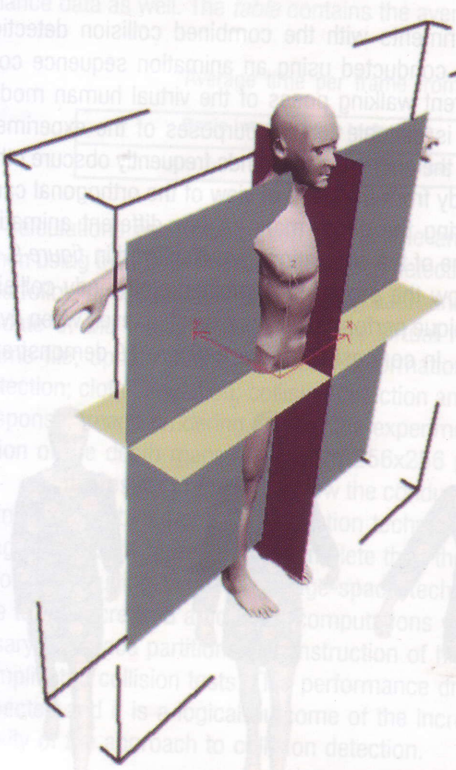


Figure 5. Uniform space partitioning to voxels for building the hierarchy

While building the hierarchy the triangles are kept in the leaf nodes even when intersections with the splitting plane occur. No splitting of triangles is performed; they are simply included in both branches of the hierarchy. This is reasonable because as mentioned above the faces are quite small in comparison to the entire body and further subdivision would not be beneficial.

5. Garment Description and Editing

As mentioned in Section 2, the existing CAD apparel systems only export the geometry of the cutting patterns which is not sufficient for cloth simulation. That is why a proprietary format has been developed for the described system. In addition, a piece of software for editing garments had to be developed, which can import a DXF file, edit the patterns and save the garment in new format.

Two approaches for editing garments were explored in the described system:

1. A stand-alone MS Windows programme for editing garments was developed, which has the following functions: import a garment from a DXF file, geometric manipulations (move, mirror, rotate), specify the panel type (front, back, sleeve, etc.), specify garment type (shirt, trousers) and size, position the panels, specify seaming lines, specify texture (JPEG file) for each panel, save the garment in the proprietary format, etc. (*figure 6*). A freeware open Source DIME library was used for implementing the DXF import.

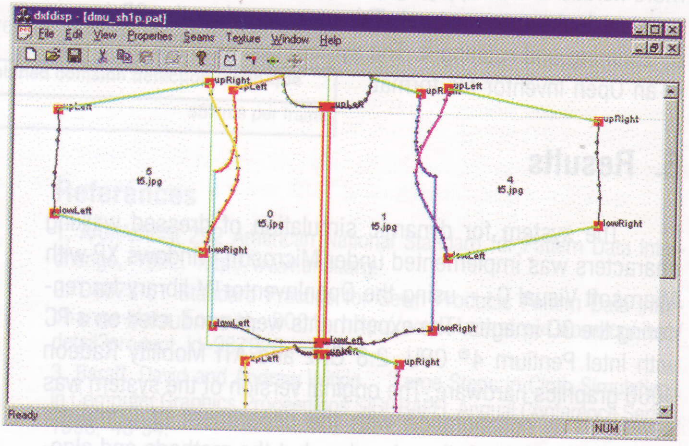


Figure 6. Garment editor implemented as stand-alone MS Windows programme

2. A plug-in to Adobe Illustrator (AI) was developed. A screen shot is given in *figure 7*. This module possesses the

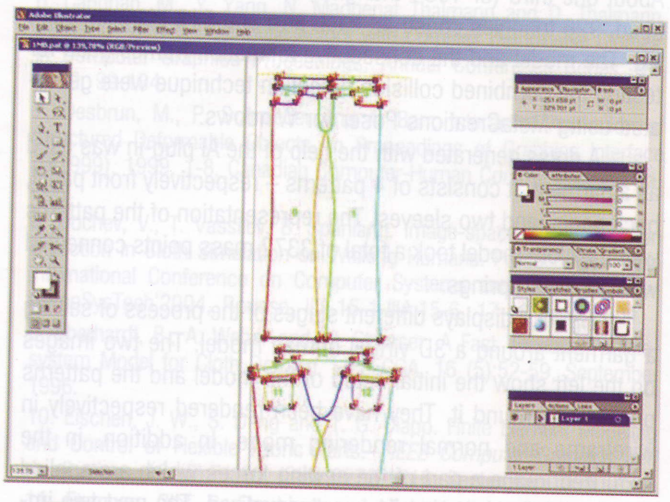


Figure 7. Garment editor implemented as plug-in to Adobe Illustrator

same functionality as its stand-alone counterpart. Its main advantage is that AI has a lot of built-in function for DXF import and manipulating geometric objects, which do not need to be implemented, like: moving, mirroring, rotation, polygon edition, etc.

As a result of the garment editing programmes all necessary information is written in a garment pattern file. The format of the output file for the two programmes is the same. The cloth simulation software reads two input files: one, describing the 3D human body, and the second describing the garment. For each of the seaming patterns, a grid of material mass points is generated, and these panels are automatically positioned around the human body. This is possible, because the body is acquired through a 3D scanner, which extracts important landmarks, like shoulders, waist, etc., which are written in the body file. After that external elastic forces are applied along the seaming lines. After a certain number of iterations the patterns are sewn to each other and the garment is „dressed" on the human body. Then a few more iterations with applying gravity are performed, so that the garment looks more natural. The user can view the 3D image by zooming and rotating it. The system can save the 3D image in an Open Inventor file format.

6. Results

The system for dynamic simulation of dressed walking characters was implemented under Microsoft Windows XP with Microsoft Visual C++, using the OpenInventor™ library for rendering the 3D images. The experiments were conducted on a PC with Intel Pentium 4® CPU, 2.8 GHz and ATI Mobility Radeon 9000 graphics hardware. The original version of the system was developed in collaboration with the department of Computer Science, University College London, but the methods and algorithm described in the paper are sole implementation of the authors.

A 3D virtual human model consisting of 2207 vertices defining 4410 faces (triangles) was used for the experiments. About one third (or 1534 triangles) describe the torso and the head, 926 triangles construct each arm and 512 triangles each leg. The model and the consecutive animation frames used for testing the combined collision detection technique were generated using MetaCreations Poser for Windows.

A dress generated with the help of the AI plug-in was used as a garment. It consists of 4 patterns – respectively front panel, back panel, and two sleeves. The representation of the patterns with the cloth model took a total of 3372 mass points connected with 18 623 springs.

Figure 8 displays different stages of the process of sewing a garment around a 3D virtual human model. The two images on the left show the initial setup of the model and the patterns positioned around it. They have been rendered respectively in wireframe and normal rendering mode. In addition, in the texturized image a part of the sewing forces which are applied along predefined sewing lines are visualized. The next two images on the right demonstrate an intermediate stage of the sewing process and the dressed model of the virtual human after the successful completion of sewing. The latter took an average of 2.264 s to complete.

Figure 8. Sewing a garment around 3D virtual human model via properly applied sewing forces

The experiments with the combined collision detection technique were conducted using an animation sequence consisting of different walking poses of the virtual human model. This sequence is suitable for the purposes of the experiment because during the animation the hands frequently obscure other parts of the body from the point of view of the orthogonal cameras for acquiring the depth maps. A few different animation frames from one of the simulations are displayed in *figure 9*. As the images show the proposed combined cloth-body collision detection technique performs well as expected, even when overlapping occurs. In contrast to *figure 9*, *figure 10* demonstrates

Figure 9. Garment simulation on a 3D model of a virtual walking human

manifestation of the drawback of the pure image-space technique when objects or parts of objects overlap. The attempt to carry out a simulation using the basic image-space technique fails when overlapping which leads to loss of essential depth information emerges. In the particular case in *figure 10* when the right arm gets in front of the torso the front garment panel is incorrectly tested for collisions against the depth information for the right arm. As a consequence the collision response algorithm tries to move the front panel in front of the right arm towards the viewer which ruins the simulation.

Figure 10. Failure during simulation employing the basic image-based approach

During the experiments with each of the collision detection techniques along with the visual data we gathered performance data as well. The *table* contains the average time spent in calculations and visualization of a single animation frame when using the above mentioned collision detection techniques. The following issues take place during each animation frame: update of the triangles describing the virtual human from a frame file; update of the necessary information for collision detection; cloth simulation; collision detection and if necessary response; image rendering. During the experiments the resolution of the depth maps was set to 256x256 pixels.

Average time per frame from animation of dressed walking virtual human model

Basic image-space technique	Combined collision detection technique
405 ms per frame	557 ms per frame

As the results in *the table* show the conducted simulation using the combined collision detection technique took an average of 37.53% more time to complete than the same simulation when using the basic image-space technique. This is due to the increased amount of computations which are necessary for space partitioning, construction of hierarchies and complicated collision tests. This performance drop is not unexpected and it is a logical outcome of the increase in complexity of the approach to collision detection.

7. Conclusions

This paper presented a system for dressing virtual humans capable of dynamic simulations. It utilizes a modified mass-spring cloth model with improved super-elasticity handling. An approach for garment representation and two applications for editing garments for the purposes of the system have been implemented. The above allowed us to achieve quick and realistic static simulations of dressed virtual human models on a regular PC. During the experiments to dress the models the proprietary file format for garment description, which has been developed, proved its applicability. In addition the two editors of garment consisting of patterns were found to be useful. They ease and partly automate the process of marking-up garment for the purposes of simulation. The garment designers find the plug-in to Adobe Illustrator more useful because most of them are familiar with the Adobe products and their interface. The stand alone garment editor is still of benefit in cases when spending money on a licence for AI is not reasonable.

An efficient combined cloth-body collision detection technique which incorporates both image and object-space techniques has been developed for the purposes of the system in regard to dynamic simulations. The conducted experiments with animation sequence which represents a walking virtual human showed that the technique detects collisions adequately. As expected the technique does not suffer from the drawback of the pure image-based algorithms when overlapping in the 3D scene occurs. Despite our efforts to develop a fast collision detection algorithm, the achieved animation speed is still not satisfactory. Further possibilities to speed up the collision detection process in order to achieve real-time animation rates should be investigated.

References

1. AAMA/ANSI 292 American National Standard for Pattern Data Interchange, 1992. http://www.ansi.org.

2. D6673-01 Standard Practice for Sewn Products Pattern Data Interchange-data Format, 2001. http://www.techstreet.com/cgi-bin/detail?product_id=937920.

3. Baraff, David and Andrew Witkin. , 'Large Steps in Cloth Simulation'. In Computer Graphics Proceedings SIGGRAPH, Annual Conference Series, 1998, 43-54.

4. Bigliani, R. and J. W. Eischen. Collision Detection in Cloth Modeling. In D. H. House and D. E. Breen, editors, Cloth Modeling and Animation, 2000, 197-217. A. K. Peters.

5. Breen, D. E., D. H. House and M. J. Wozny. Predicting the Drape of Woven Cloth Using Interacting Particles. In Computer Graphics Proceedings, Annual Conference Series, 94, 1994, 365-372.

6. Carignan, M., Y. Yang, N. Magnenat Thalmann and D. Thalmann. Dressing Animated Synthetic Actors with Complex Deformable Clothes. In Computer Graphics Proceedings, Annual Conference Series, 92, 1992, 99-104.

7. Desbrun, M., P. Schroeder and A. Barr. Interactive Animation of Structured Deformable Objects. In Proceedings of Graphics Interface (GI1999), 1999, 1-8. Canadian Computer-Human Communications Society.

8. Dochev, V., T. Vassilev, B. Spanlang. Image-space Based Collision Detection in Cloth Simulation on Walking Humans. Proceedings of the International Conference on Computer Systems and Technologies – CompSysTech'2004, Rousse, IIIA.15-1-IIIA.15-6, 17-18 June 2004.

9. Eberhardt, B., A. Weber and W. Strasser. A Fast, Flexible, Particle-system Model for Cloth Draping. j-IEEECGA, 16 (5):52-59, September 1996.

10. Eischen, J. W., S. Deng and T. G. Clapp. Finite Element Modeling and Control of Flexible Fabric Parts.– *IEEE Computer Graphics and Applications*, 16 (5):71-80, ISSN 0272-1716, September 1996.

11. Etzmuss, Olaf, B. Eberhardt and Michael Hauth. Implicit-explicit Schemes for Fast Animation with Particle Systems. In Computer Anima-

tion and Simulation, 2000, 138-151. Eurographics, August 2000.ISBN 3-211-83392-7.

12. Feynman C. R. , Modeling the Appearance of Cloth. MS Thesis, EECS Dept, MIT, May, 1986.

13. Kawabata, S. The Standardization and Analysis of Hand Evaluation. The Textile Machinery Society of Japan, 1980.

14.Lin, M. and S. Gottschalk. Collision detection between geometric models: A survey, 1998.

15. Provot, X. Deformation Constraints in a Mass-spring Model to Describe Rigid Cloth Behaviour. Proceedings of Graphics Interface 1995, 141-155.

16. Provot, X. Collision and Self-collision Handling in Cloth Model Dedicated to Design. Computer Animation and Simulation 97, 1997. 177-190, ISBN 3-211-83048-0. Held in Budapest, Hungary September.

17. Terzopoulos, D., J. Platt, A. Barr and K. Fleischer. 1987, Elastically Deformable Models.– *Computer Graphics* (Proc. SIGGRAPH'87), 21 (4), 205-214.

18. Teschner, M., S. Kimmerle, B. Heidelberger, M. Teschner, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P.Cani, F. Faure, N. Magnenat Thalmann, W. Strasser and P. Volino. Collision Detection for Deformable Objects.– *Eurographics*, Sep 2004.

19. Vassilev, T., B. Spanlang and Y. Chrysanthou. Fast Cloth Animation on Walking Avatars.– *Computer Graphics Forum*, 2001, 20(3), 260-267, ISSN 1067-7055.

20. Volino, P. and N. Magnenat-Thalmann. Collision and Self-collision Detection: Efficient and Robust Solutions for Highly Deformable Surfaces. Computer Animation and Simulation '95. September, 1995, 55-65. ISBN 3-211-82738-2.

21. Volino, P. M. Courchesne and N. Magnenat-Thalmann. Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects. In Proc. SIGGRAPH'95, 1995, 137-144.

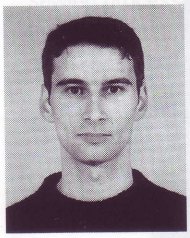
22. Volino, P. and N. Magnenat-Thalmann. Comparing Effciency of Integration Methods for Cloth Simulation. In Computer Graphics International 2001, 265-272, July, 2001. ISBN 0-7695-1007-8.

23. Weil, J. The Synthesis of Cloth Objects. In David C. Evans and Russell J. Athay, Editors.– *Computer Graphics* (SIGGRAPH '86 Proceedings), August 1986, 49-54.

24. Web Site of OptiTex, Fashion and Textile Design Software. http://www.optitex.com.

25. Zhang, D. and M. M. F. Yuen. Collision Detection for Clothed Human Animation. 8th Pacific Conference on Computer Graphics and Applications, October, 2000, 328-337. ISBN 0-7695-0868-5.

Manuscript received on 11.05.2006

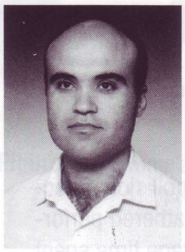


***Vladimir Dochev** has graduated from the University of Rousse, Bulgaria, speciality of Computer Engineering with major in Computer Networks in 1999.*

Since 2001 he has been a system administrator at the University of Rousse, Department of Computing. Currently he is also a Ph.D. student. His research interests include cloth modelling and simulations in 3D CG

Contacts:

e-mail: vdochev@ecs.ru.acad.bg



***Tzvetomir Vassilev** has graduated from the University of Rousse, Bulgaria, speciality of Computer Engineering in 1989. In 1997 he received his PhD in the field of data acquisition systems, processing and modelling of experimental data. Since 2004 he has been an Associate Professor at the University of Rousse, Department of Informatics and Information technologies. His research interests include modelling and approximations of curves and surfaces, cloth modelling and simulations.*

Contacts:

e-mail: ceco@ami.ru.acad.bg