A 3D graphic environment for garments design

Therese Bonte¹, Andrea Galimberti² and Caterina Rizzi²

¹ LAMIH, University of Valenciennes, France

² Dipartimento di Ingegneria Industriale, Università di Parma, Italy

Abstract: This paper outlines a 3D graphic environment for industrial applications in the clothing industry. In particular, it permits the design of a base garment in 3D, verification of its style through the comparison of different types of fabric on a virtual mannequin, and automatic generation of 2D patterns from the 3D representation. The system has been developed within the framework of the Brite Project MASCOT, whose consortium comprises two leading manufacturers of CAD/CAM systems, two research institutes, an industrial company specialised in made-to-measure systems, and three garment manufacturers as end-users. The main modules of the 3D environment are described, as well as the methodologies and techniques adopted.

Key words: 3D modelling, Physically based modelling, clothing industry.

1. INTRODUCTION

Clothing manufacture is highly labour-intensive. At present there are various software tools available for 2D design, including grading, nesting and so on. Although they permit great savings in terms of time and materials, each type of technique covers only a particular step of the garment design process. There is, therefore, a strong need for integrated systems with the capacity deal with the process as a whole, including possibilities for the designer to work directly within a 3D environment.

Traditionally, there are two types of garment design process:

- The first is the most common, and involves drawing 2D patterns, followed by draping on a mannequin. The process is iterated until the best result is achieved and is regarded as effective only if the garment is

close to an existing design. This approach is currently used within various companies, especially if the design process is low in innovation.

- The second technique consists of the design of 3D garments around a digital mannequin. This technique is more natural and creative because it allows direct transference of the garment concept. Unfortunately, it requires a great deal of experience and designers often have to spend a lot of time in order to arrive at the final garment.

In the field of CAD/CAM, computer based systems have been developed based upon the above-mentioned processes. With reference to the first process, there are systems that allow the modellist to design 2D patterns, sew them together, and put them on a 3D mannequin, thereby using physical techniques to provide an estimate of their wear-ability. Industrial packages, such as the PAD system (www.padsystem.com), or research tools (www.cadcam.ust.hk/research/garment.html), (Carignan 1992), (www.ercim.org), (Provot 1997, Imaoka 1984, Okabe 1992), use this methodology to develop 3D garments and 2D patterns. Only a few research teams, such as (Hinds 1990, McCartney 1999) and the University of Valenciennes, work on 3D garment design using the second method. They achieve this by employing geometric techniques to model the 3D garment and generate 2D panels.

In this context, our objective has been to develop a 3D graphic environment for industrial applications in the clothing industry. The aim of this is to permit the design of female base garments in 3D, together with evaluation of their styles. This is attempted through comparison of different types of fabric and the automatic generation of 2D patterns starting from the 3D representation. In this paper, we present a system developed within the framework of the Brite Project MASCOT (<http://www.univ-valenciennes/LGIL/mascot>). Figure 1 shows the reference architecture of our 3D design environment. Two main streams can be identified, corresponding to different work methods. One stream (coloured white) has been conceptualised around the existing technical background of the company - i.e. 2D patterns defined using 2D drafting systems. The other (coloured grey) is based upon the use of 3D graphic tools. We have concentrated our attention on this second stream.

The main modules of the prototype are the:

- *3D Modeller*, allowing the modellist to generate the 3D model of the garment and its corresponding 2D panels;
- *3D Simulator*, allowing the modellist to simulate the behaviour of the garment on the mannequin, taking into account the mechanical properties of the fabric.

The I/O interfaces of the prototype have also been highlighted. These include the:

- *Mannequin Data Base*, providing selection of the proper mannequin, both for design and simulation;
- *Material Data Base*, allowing recovery of the fabric's mechanical properties to be used by the 3D Simulator in its generation of the physically-based model. These are measured by means of the Kawabata Evaluation fabric System (KES) (Kawabata 1980);
- *2D patterns Data Base*, that recovers 2D geometric models of panels and 2D-3D mapping rules to assemble 2D patterns on the mannequin.

In the following, the 3D Modeller and the 3D Simulator are described, as well as the techniques adopted for their development.

Figure 1. Reference architecture.

2. 3D MODELLER

The objective of the 3D Garment Modeller is to provide creative tools for designers. It allows the designer to choose an underlying static mannequin, to create and edit a parametric 3D garment, to define darts, seams, holes and style lines on the 3D shape, and then to flatten 3D panels to generate corresponding 2D panels. An overview of the design process is shown in Figure 2. The design process is as follows:

1. A mannequin is chosen from the *Mannequin database* and used as a reference for the design process.
2. An approximate shape of the 3D cloth is designed around the virtual mannequin. In most CAD/CAM software, the user creates surfaces from scratch, but this approach is difficult for end-users not trained in geometric modelling. In order to avoid this problem we adopted a different method: the end-user browses a library to select the closest garment to the final one and creates the 3D shape by stating values or selecting points or lines on the mannequin.

3. The 3D shape is edited using transformation tools in a manner that is as close as possible to the natural design process. The user can change height, length, etc., with respect to given measurements.
4. Darts, seams, holes and style lines are interactively defined and edited within the 3D shape.
5. Once the 3D garment is complete, the 2D patterns are generated. Tools help the user to choose the best position for seams and darts. The parameters of the flattening process are automatically defined by the system, but may be changed in order to adjust specific garment constraints.
6. Finally, the garment is exported to the simulator.

Figure 2. Design process with the 3D Modeller.

The nature of the user interface is one of the most crucial issues involved in the design of the system due to the end-user's low level of competence in using geometric tools. One of our objectives has been to develop user-friendly interfaces based, as close as possible, upon the designer's knowledge. In addition, we have tried to minimise the end-user's need to deal with geometries through the provision of high-level tools.

2.1 Modeller architecture

Figure 3 shows the main components of the 3D Modeller.

The *Mannequin database* includes a set of static mannequins, the *Design Module* comprehends all the functionalities to design a complete 3D garment, the *Flattening Module* generates the 2D patterns and the *Graphical User Interface* allows the user to design garments in an interactive and user-friendly environment.

Figure 3. Architecture of the 3D Modeller.

The core of the *Design Module* consists of a NURBS surface modeller and a curve modeller. 3D shapes are modelled using one or more connected surfaces. A garment is a set of 3D panels, which are represented in terms of trimmed surfaces defined from the 3D shape. Curves lying on surfaces are used to define holes, darts, seam lines, assemblies or style lines on the 3D garment. They are designed using an intuitive technique, which requires only the interactive selection of curve data points along the surface.

The *Design Module* comprises three sub-modules that allow the generation of 3D parametric garments composed of trimmed surfaces. It also permits the use of convivial editing tools to edit the 3D shape, in addition to the design and modification of darts, seam lines, style lines, and holes as surface curves.

The *Flattening Module* consists of two sub-modules:

- The *Module to optimise darts & seam lines* assists the user during the flattening process to obtain an optimum solution. It visualises the best positions of darts and overlaps through the computation of Gaussian curvature and the display of its distribution on panels.

- The *module to generate 2D patterns* permits 2D patterns to be obtained from 3D panels through optimisation of constraints specific to garments.

The results generated by the system are: the 3D garment, the corresponding 2D patterns to be used by the simulator through *interface 2*, and 2D patterns in standard AAMA format for 2D industrial CAD packages.

In the following, a brief description is provided of the approaches adopted in the development of the system.

2.2 Basic Garment Design

As previously mentioned, the design of 3D garments is based on a library of basic garments. This library has been defined as follows:

- A typology of basic female garments has been defined. Garments have been classified in terms of: skirts, shirts, pants, and sleeves. Other types of garments can also be derived. For example, a dress is an assembly of a skirt and a shirt. Each category is composed of basic garments. A *skirt* may, therefore, be sub-divided into *straight skirt*, *flared skirt*, etc..
- Basic garments have been parameterised in 3D. An analysis of 2D measurement characteristics has been carried out with the help of the end-users (ESMOD 1985, Le Merrer 1989, CETIH 1981) and a correspondence, either on the mannequin or on the 3D garment, has been established.
- All of the useful garment shape features have been defined and, using geometric techniques, the cloth has been modelled without consideration of its mechanical properties.
- Relationships between 3D measurement parameters and the designer's knowledge have been established in order to define the user interface.

Figure 4 shows an example of a flared skirt, in addition to the correspondence between 2D measurement parameters and the dialog box allowing the design of the 3D basic model.

2.3 Design of Curves

The objective has been to develop a user-friendly tool to design holes, darts, seam lines, assemblies, and style lines as curves lying on surfaces. We have selected an algorithm that is very intuitive, fast and interactive. It does not depend on the parameterisation of underlying surfaces but rather on the 3D geometry of the surface (LAMIH 99). Each line (dart, seam line, etc.) is considered as an assembly of curves and is created as follows: the designer selects all data points to be interpolated on the surface and the resulting line is computed, using an extension of De Casteljau's algorithm for 3D surfaces.

Figure 4. Correspondence between sketches, 2D/3D parameters and user interface.

2.4 Flattening

The following tools have been developed to carry out the flattening process:

- Before the flattening process begins, preferential positions of darts and overlaps are displayed using a method based on the use of Gaussian curvature (LAMIH 99).
- The flattening process comprises three steps:
 - 3D panels composed of triangles are defined from surfaces and lines, using a method based on (Piegl 1995) (Piegl 1998);
 - the 3D meshes are then developed: triangles are flattened one by one with a non isometric method whilst trying to preserve lengths and areas as much as possible;
 - a relaxation procedure is applied, which refines 2D points until error is minimised.
- After the flattening process, a map based on colour shading is computed. It displays distortions between 3D dimensions (areas, lengths) and those in 2D.

3. 3D SIMULATOR

Once the 3D model of the garment has been created, the designer can simulate its behaviour (drapability) on the virtual mannequin using the 3D Simulator. The prototype proceeds according to the following steps:

- generation of the physically-based models of the 2D garment patterns according to the fabric's mechanical properties;
- assembly of 2D patterns around the 3D mannequin using 2D-3D mapping rules generated by the 3D modeller;
- sewing of the patterns according to seam and dart lines;

- simulation of the garment's behaviour.

In the following, the physically-based model adopted to represent and simulate garment behaviour is described, in addition to its main components and techniques.

3.1 Adopted model

The mathematical model used to represent and simulate the garment behaviour is of crucial concern; it should permit the description of, not only the geometry, but also physical properties such as elasticity. Physically-based models may provide a solution to this. In the available literature techniques and tools, derived from physically based modelling, can be found; some are specifically oriented to modelling and simulation of cloth (Breen 1994, Carignan 1992, Eberhart 1996, Hing 1996, Provot 1997, Siggraph 1999, Volino 1996). Most of them, however, are oriented towards applications for animation, the main objective of which is to produce images that look good. Only a few have been applied to industrial applications. We adopted the particle-based model that is the most widely known and used discrete model (Breen 1994, Eberhardt 1996, Luciani 1991, Siggraph 1998, Tonnesen 1991, Volino 1997, Witkin 1995). This model describes an object as a set of particles in terms of mass, radius and other physical properties. Both energy based (House, 1998) and force based (Witkin, 1995) techniques are currently available. Of the two the former is not adequate for dynamic analysis, and therefore we adopted the latter, force-based, approach. This involves modelling of the interaction laws among the particles through the use of forces and constraints that determine the dynamic behaviour of the material. The simulation is carried out using the Newton law: $f = ma$. The resulting mathematical model of a particle system is a system of 2nd order ordinary differential equations, that can be reduced to a 1st order equivalent system and, therefore, solved step by step with numerical integration.

3.2 Simulator Architecture

Figure 5 shows the main modules of the 3D Simulator: the *Modelling Kernel* that generates the physically-based model of the 2D patterns, the *Simulation Kernel*, a *Graphic Viewer*, and a *Graphical User Interface*.

The Modeller possesses all the necessary functionality to:

- import the mannequin used by the modellist to design the 3D garment;
- automatically generate the particle model of 2D panels produced by the 3D Modeller according to the material properties;
- assemble 2D patterns over the mannequin using the 2D-3D mapping rules generated by the 3D Modeller;

- impose constraints, such as those involved in sewing.
The Simulation Kernel is a generic solver and comprises:
- an *ODE* (Ordinary Differential Equation) numerical solver;
- a *Constraint Manager*, whose aim is to manage constraints and obstacles during the simulation.

Figure 5. 3D Simulator architecture.

3.3 Generation of the Particle-based model

As previously mentioned, then first step involves generation of the particle model of the 2D panels. To do this, it is necessary to make the 2D panels discrete according to the adopted model, define the distribution of forces, and sew the 2D panels. We model a 2D panel as a *rectangular grid* of *particles* connected by a *set of forces* describing the mechanical properties of the fabric. The two directions of the grid model the warp and weft directions and therefore it is possible to simulate material anisotropy by characterising differently the forces along the two directions. Particular attention has been paid to 2D panel edges to allow the subsequent production of seams with other panels. First, a particle is placed at each vertex of the 2D pattern contour; then particles are added along the pattern edges to have roughly the same resolution present in the inner part. The algorithm splits edges recursively based on the grid step. The particle mass is computed through the analysis of fabric density derived from KES measurements.

The internal forces used are:

- stretching and repelling forces applied between pairs of particles that tend to keep particles at the rest distance;

- bending forces applied to a line of three particles that keep them aligned;
- trellising (shear) forces that act over squared cells of four particles and contrast deformations within the plane.

Parameters describing forces are derived from KES measurements. Spring and bending forces may have different parameters in warp and in weft directions.

Finally, using 2D-3D mapping rules generated by the 3D Modeller, each panel is correctly sewn and placed around the mannequin.

3.4 Constraints and Collision management

To describe the behaviour of deformable objects such as fabric, it is not sufficient to consider only internal forces. It is also necessary to take into account:

- *external forces*, such as gravity, treated in the same manner as the internal ones;
- *constraints*, that restrict allowed positions and movements of an object, i.e., the conditions that must be respected by the object during its motion;
- *collisions with obstacles*, which occur when a flexible object hits a rigid object (e.g. the mannequin) or penetrates itself (self-collision).

Various techniques can be found in literature to handle such constraints (Platt 1988, Witkin 1990). We used the *dynamic constraints* method since it permits the application of multiple constraints to the same particle and ensures that all of the constraints are respected at each step of the simulation. The major drawbacks of this approach are the high computational cost and numerical problems when the linear system becomes ill conditioned.

Collision detection is often the bottleneck of deformable objects simulation systems handling highly discrete objects. The high number of objects and of surface elements forces the simulator to execute a continuous check of the possible situations of collision (collision detection) and takes a great deal of the simulation time. Two aspects should be considered: *collision detection* and *collision response*.

Collision detection

Instead of detecting collisions for each object (described by a set of particles) against all other objects, optimisation techniques have been introduced in order to reduce computational time (Baraff 1998, Badler 1998, Provot 1997b, Volino 1994, Volino 1995). We adopted the *Bounding box hierarchy* technique (Baraff 1998), a good compromise between simplicity and efficiency. The collision test between two objects can be skipped when their bounding boxes do not intersect. This method must be improved in some way because a deformable object could self-collide. Self-collisions occur mainly in fabrics, when they are folded or when they form deep

wrinkles. We approach this problem by splitting objects into regions and applying bounding boxes to each region. Different regions are considered as different objects. The algorithm detects the collisions by exploring the bounding box intersections in the hierarchy.

Collision response

The *collision management* system works to prevent particles from crossing an object's surface. Two objects are regarded as colliding when the distance is less than a threshold value. We adopted this solution to deal with flat objects such as fabrics, since they have no distinction between inner and outer surfaces. Setting an appropriate threshold creates a thin virtual volume around the surface. All of the particles that enter the volume are detected and their velocities modified. The threshold value should be small, yet large enough to ensure detection during simulation.

To ensure system stability it is very important to pay attention to interactions between ODE and the collision solver. In the Simulation Kernel we are using a Eulero ODE, modified according to (Provot 1997b). This updates velocity before position. Using this algorithm when an object is continuously "pressed" against another object leads to some penetrations and loss of ODE stability. This problem arises because the collision solver changes velocities to avoid penetration, but ODE updates positions after the velocity is updated. This means that positions are updated using "incorrect" data because velocities change after force application

Collision with rigid and scene objects

Simulating a garment draped on a mannequin requires an object and a rigid mannequin, the position of which is fixed. To deal with rigid object collision, we adopted an *ad hoc* solution by which it is not necessary to solve the linear system required by the dynamic constraints, thus saving a lot of time. The rigid objects keep fixed positions or trajectories so it is very easy to make a direct computation of the correct collision response. In the simplest case a particle collides with a rigid triangle: the only possible course of action is to revert the velocity component to a position perpendicular to the surface (for a totally elastic collision). Other cases are slightly more complicated, but similar solutions can be found.

Improving efficiency, decoupling detection and response

A further improvement comes from a simple idea: the simulation step is so small that it is not necessary to search for new collisions at each step. During the ODE step it is easy to find the velocity of the speediest particle. In the worst case, this particle is close to collision with a triangle, the distance between the two being only a little larger than the threshold value, dp . Computing the time tc needed for this particle to cover half of dp , we can be sure that no undetected collisions will lead to a penetration in the next tc

seconds. It is possible to make a more conservative computation of tc based on a quarter or less of dp or to be more aggressive using bigger fractions.

4. FINAL CONSIDERATIONS

The system has been developed experimentally for the design of female garments. Validation and assessment procedures have been defined and both prototypes have been assessed by the end-users. The following considerations may be summarised.

In terms of 3D design and flattening, we have selected a set of examples comprising basic garments (e.g., skirts, shirts, sleeves) with different parameters and derived garments. The differences between real garments and calculations were no more than 4% for each 2D parameter defined in (ESMOD 1985) (Le Merrer 1989) (Figure 6). As regards speed, all manipulations were performed quickly and algorithms were solved instantaneously on a PC with Windows NT, except flattening with relaxation procedure, which takes no more than 5s. Few experiments have been made to test the end-users' satisfaction with reference to the design capabilities. Since we have developed the 3D Modeller with the intention of respecting designer's knowledge, 3D geometric aspects have been hidden and do not present any difficulties. Even so, 3D basic garments and 3D editing tools should be improved to increase end-users' creativity. The system also offers the possibility to compute 3D garments and 2D panels in different sizes. Up to now, grading is performed in 2D with empirical rules. Using a library of different sized mannequins, we can deduce garments in different sizes from mannequins of corresponding sizes and garments of basic size by simply reporting offset between the mannequin and the garment.

As far as the simulation is concerned, the procedure is performed automatically and level of user interaction is very low, once the correct parameters for discretisation have been defined (e.g., grid step and force parameters), depending on fabric type. We have selected a set of different garments (Figure 7), and physical prototypes have been realised by the end-users allowing them to be compared with the simulation results and have generally been considered quite good. At this stage, however, comparisons have only been made at a qualitative level.

The main problem remains the calculation time. This depends heavily on garment complexity (discretisation grid resolution/n. of particles and material parameters). The simulation time can take from a few minutes up to several hours, depending on the hardware platform.

In terms of multi-layered clothing, we have simulated a skirt with a shirt, but the overlap was very limited, as were the collisions. Experiments with

the simulator have also been conducted using men's garments. A jacket can be considered one of the most complex garments, multi-layered parts of which have been modelled by changing the force parameters, derived from KES measurements performed on multi-layered specimens. The results were encouraging, although simulation times increased rapidly and further improvements are necessary.

Figure 6. Examples of garments.

Figure 7. Simulation results of different garments.

REFERENCES

- Badler N. I. and Glassner A. S. (1998) *3D Object Modelling*, Course Note n. 12, Introduction to Computer Graphics, SIGGRAPH 1998, pp. 1-14.
- Baraff D. (1998). *Rigid Body Simulation*, Course Note n. 13, SIGGRAPH '98, pp. D1-D68.
- Breen D., House D., Wozny M.J. (1994). *A Particle-based model for Simulating the Draping Behaviour of Woven Cloth*, Textile Research Journal, Vol. 64, N0. 11, November 1994, pp. 663-685.
- Carignan M., Yang Y., Magnenat Thalmann N., and Thalmann D. (1992). *Dressing animated synthetic actors with complex deformable clothes*, Computer Graphics (Proc. Siggraph), vol. 26, n. 2, July 1992, pp. 99-104.

- CETIH (1981). *Construction des vêtements, jupe, buste*. Internal Document.
- Eberhardt B. and Weber A. (1996). *A Fast Flexible, Particle – System Model for Cloth Draping*, Computer Graphics in Textiles and apparel, IEEE Computer Graphics and Applications, vol. 16, n.5, September 1996, pp. 52-59.
- ESMOD (1985). *Méthode de coupe de vêtements féminins*, MPGL, Paris, France.
- Hinds B.K., McCartney J. (1990). Interactive garment design, *Visual Computer*, vol. 6, pp. 53-61.
- Hing N. N.g., and Grimsdale L. (1996). *Computer Graphics Techniques for Modeling cloth*, Computer Graphics in Textiles and apparel, IEEE Computer Graphics and Applications, vol. 16, n.5, September 1996, pp. 28-41.
- House D. H., Breen D. E. (1998) Representation of Woven Fabrics, *Course Notes n. 31 Siggraph 98*.
- LAMIH (1999). *Architecture of 3D modeller*, MASCOT report, Brite project 3542, 1999
- Imaoka H., Okabe H. , Akami H. (1984). Analyse structurale du drapement des vêtements par simulation informatique (translated from Japanese) *Sen'i Kobunshi Zairyo Kenkyujo Kenkyu Hokoku*, n. 142, pp 57-61.
- Kawabata S. (1980). *The Standardization and Analysis of Hand Evaluation*, The Textile Machinery Society of Japan, Osaka.
- Le Merrer J.Y. (1989). *Méthodes de coupe de vêtements féminins classiques*, Editions Vauclair, Paris, France.
- Luciani A., Jimenez S., Florens J.L., Cadoz C., and Roult O. (1991). *Computational Physics: A modeler-simulator for Animated Physical Objects*, Proc. Eurographics 91, pp. 425-436.
- McCartney J., Hinds B.K., Seow B.L., (1999). The flattening of triangulated surfaces incorporating darts and gussets, *Computer Aided Design (CAD) n. 31*, pp249-260.
- Okabe H., Imaoka H., Tomira T., Niwaya H. (1992). Three Dimensional Apparel CAD System, *Computer Graphics*, 26,2, July 1992, pp 105-110.
- Piegl L.A., Richard A.M. (1995). Tessellating trimmed NURBS surfaces, *Computer Aided design (CAD)*, vol. 27, n.1, pp 16-26.
- Piegl L.A., Tiller W. (1998). Geometry based triangulation of trimmed NURBS surfaces, *Computer Aided design (CAD)*, vol. 30, n.1, pp 11-18.
- Platt J., Barr A. H. (1998). *Constraint Methods for Flexible Models*, Computer Graphics, vol. 22, n.4, pp. 279-288.
- Provot X. (1997). *Animation realiste de vêtements*, PhD Thesis, Université René Descartes, Paris V, Mathématiques et Informatiques.
- Provot X. (1997). *Collision and self-collision handling in cloth model dedicated to design garments*, Proc. Graphics Interface 97, pp. 177-189.
- Siggraph, (1998). *Cloth and Clothing in Computer Graphics*, Course Note n. 31.
- Siggraph, (1999). *Physically Based Modeling*, Course Note n. 36.
- Volino P. and Thalmann N. M. (1994). *Efficient self – collision detection on smoothly discretized surface animations using geometrical shape regularity*, Computer Graphics Forum (Proc. Eurographics 94) vol. 13, n.3, Oslo, Norway, September 1994, pp. 155-166.
- Volino P., Thalmann N. M., Jianhua S., and Thalmann D. (1996). *An evolving system for simulating clothes on virtual actors*, Computer Graphics in Textiles and apparel, IEEE Computer Graphics and Applications, vol. 16, n.5, September 1996, pp. 42-51.
- Volino P. and Thalmann N. (1997). *Developing Simulation Techniques for an Interactive Clothing System*, Proc. VSMM 97, IEEE Computer Society, pp. 109-118.
- Witkin A. (1990). *Interactive Dynamics*, Computer Graphics, vol. 24. Proc. Symposium on 3D Interactive Graphics, pp.11-20.
- Witkin A. (1995). *Particle System Dynamics*, SIGGRAPH 95 Course Notes, 34.