Ontologies for Virtual Garments

Arnulph Fuhrmann
Fraunhofer Institut für Graphische Datenverarbeitung, Darmstadt, Germany
afuhr@igd.fhg.de
http://www.igd.fhg.de/igd-a3

Clemens Groß
Fraunhofer Institut für Graphische Datenverarbeitung, Darmstadt, Germany
cgross@igd.fhg.de
http://www.igd.fhg.de/igd-a3

Andreas Weber
Institut für Informatik II, Universität Bonn, Germany
weber@cs.uni-bonn.de
http://cg.cs.uni-bonn.de

Abstract

We give an ontology for garment patterns that can be incorporated into the simulation of virtual clothing. On the basis of this ontology and extensions to garments we can specify and manipulate the process of virtual dressing on a higher semantic level. Moreover, we show how the process of collision detection can be enhanced by using the induced semantic information.

Keywords: ontologies, garment classification, cloth simulation

1 Introduction

Using a physics-based modeling layer above the basic geometric layer is one of the very successful modeling technique for virtual garments. In the last decade the physics-based layer of virtual garments has been investigated in depth in all its aspects, ranging from physics-based models of the cloth itself and solution techniques [1, 2, 3, 4, 5] to methods for collision detection [6, 7, 8, 9, 10, 11] and collision response [7, 12].

However, in contrast to other areas of computer animation in which modeling layers above a physic-based layer have been used successfully—e.g. in the so called "cognitive modeling hierarchy" for character animation [13]—few work has been done in the context of virtual garments on using such higher modeling layers. An exception is [14], in which a method for interaction free dressing of virtual humans is developed that uses an abstract representation of cloth patterns and the human body, a geometric "pre-positioning" algorithm and finally physics-based techniques.

Although an ontology for cloth patterns is implicit in [14] it has not made explicit. In this paper we give an explicit ontology for cloth patterns—that can be formally described by current ontology languages such as OWL [15]—and we show how it can be extended to ontologies for garments. The information given by the corresponding properties are a coarsening of the geometric information that can not only be used for modeling on a more abstract level but is also suitable for retrieval of garments and supervised learning of various properties. Moreover, we provide a "semantic-based collision detection algorithm" that incorporates information provided by the properties of the garments into collision detection. The described algorithm tolerates difficult initial intersections and allows efficient and robust collision detection between several layers of garments.

2 An Ontology for Garments

2.1 Abstract description of a human body

Our ontology for the garments has to involve an abstract description of a human body.

We use a class `bfp` of *body feature points* which has 11 members, such as `girthPoint`, `rightNipple`, etc. The feature points are contained in Table 1.

For the computation of the feature points from 3D laser range scans of humans several systems are available and have been described in the literature [16, 17, 18].

In addition we use a class `bsh` of *body segment hulls*, which has 8 members that are also listed in Table 1.

We use an object property, i.e. a binary relation `featurePointsOnSegment` $\subseteq$ `bsh` $\times$ `bfp`: for every body segment hull it gives the set of corresponding feature points. Table 1 lists the required body segment hulls and feature points, i.e. the relation.

Clearly, some feature points are in relation to several body segment hulls, e.g. `waistToHipPointRight` is on `rightLeg` but also on `bothLegs` and `torsoAndLegs`.

Figure 1 shows a virtual human with body segment hulls for neck, torso, arms and legs. Also, the feature points are shown. The cylindrical hulls can be obtained by computing the segments' principal axis and projecting the segment points along this axis onto a normal plane. Then the two-dimensional convex hull of the projected points serves as a basis for the body segment hull.

During the pre-positioning all computations will be done with respect to the body segment hulls, therefore we also project the feature points on the surface of the hulls.

body segment hulls	**feature points**
neck	neck back, neck front, right neck, left neck
leftArm	left upper arm
rightArm	right upper arm
torso	neck back, neck front, left nipple, right nipple, waist girth point, waist to hip point right, waist to hip point left
leftLeg	waist to hip point left
rightLeg	waist to hip point right
bothLegs	waist to hip point left, waist to hip point right, waist girth point
torsoAndLegs	neck back, neck front, left nipple, right nipple, waist girth point, waist to hip point right, waist to hip point left

Table 1: The body segment hulls and the corresponding feature points (the relation `featurePointsOnSegment`).

2.2 An ontology for cloth patterns

For the class `cp` of *cloth patterns* we can now define the following properties.

- A binary relation

  $$\texttt{isDirectlySewedWith} \subseteq \texttt{cp} \times \texttt{cp}$$

  that gives all pairs of cloth patterns that are directly sewed together and the transitive property `isSewedWith` that is the transitive closure of the `isDirectlySewedWith` relation. Notice that in OWL transitive properties can be defined.

- A binary relation

  $$\texttt{liesOn} \subseteq \texttt{bfp} \times \texttt{cp}.$$

  Informally, this relation is expressing that (the normal projection of) a certain body feature point is lying on a pattern. Notice that in the following this property might be given only for some patterns and some feature points.

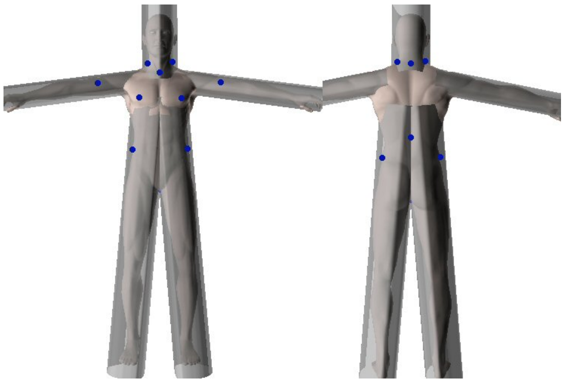

Figure 1: Virtual human with feature points and body segment hulls for neck, arms and legs.

- A binary relation

$$\texttt{isLyingOn} \subseteq \texttt{cp} \times \texttt{bsh}.$$

  Informally this relation is expressing that (the projection of) a certain pattern is mainly lying on a certain body segment hull. The formal interpretation requires the algorithmic methods that are described below.

The necessary information to build these relations can be extracted algorithmically from the CAD information of the cloth patterns and the annotations that we give below. In the following subsection we also show how the precise "geometric" interpretation of these properties has to be obtained.

2.2.1 Information on cloth patterns

Within a CAD system garments are represented by their two-dimensional cloth patterns. The bounding curves of the patterns are given by a sequence of two-dimensional vertices. In addition to the bounding curves, there must be given some sewing information. This is done by a set of seams that indicate, which part of one pattern has to be sewn together with corresponding parts of other patterns. Note that the two corresponding curves of a seam need not have necessarily the same length. In Figure 2 the bounding curves and sewing information of a female jacket are shown.

So the relation `isDirectlySewedWith` is an obvious coarsening of this CAD information that can easily be extracted algorithmically from these CAD information.

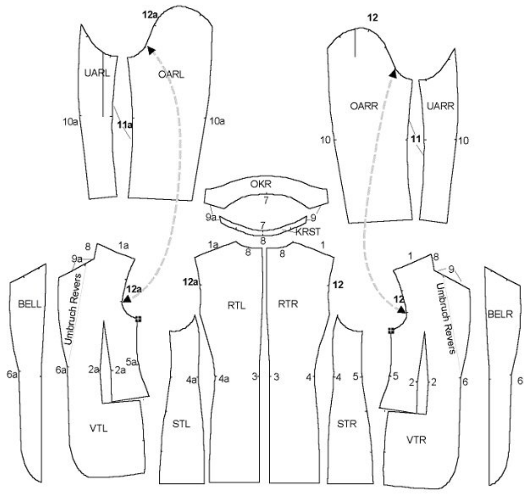

Figure 2: Cloth patterns of a jacket and the sewing information.

For interaction-free pre-positioning of cloth patterns additional information is needed. For example, a skirt can be rotated around the hip and you may not be able to determine the intended positions. Thus you need the position and orientation in relation to the human body of at least one pattern to determine the right positions of the cloth patterns. This is done by giving a two-dimensional position of a nearby feature point in relation to this pattern and an up-vector which points in the direction of the principal axis of the appropriate body segment.

For our pre-positioning algorithm, we require the position of one feature point for one pattern per body segment. Clearly, this information is independent from the size of the human figure and thus must be stored only once. Figure 3 shows the structure of a piece of clothing containing all the information that is required for the pre-positioning.

From the garment data structure the properties `liesOn` and `isLyingOn` can be extracted. The pre-positioning algorithm is described in more detail in [14]. It gives a reasonable geometric interpretation to this properties: the pre-positioning and a physics-based "end positioning" yield the geometric shape of the collection of virtual garments with respect to a specific 3D virtual human.

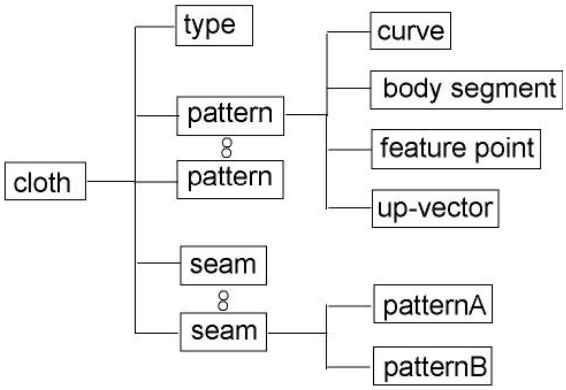

Figure 3: Garment data structure.

2.3 Ontologies for garments

We can now introduce a class `garments` for single pieces of garment. Such a piece of garment might be defined to be a collection of cloth patterns that are sewed together. This can be formally defined in languages like OWL that provide transitive properties. Commonly used descriptions of garments such as `jackets`, `trousers`, `skirts` and `dresses` can now be introduced as sub-classes of `garments`. The property

$$\mathtt{isPartOf} \subseteq \mathtt{cp} \times \mathtt{garments}$$

denotes that a certain cloth pattern is part of a certain piece of garment.

Using the ontology for cloth patterns it seems to be possible to formally define these sub-classes of garments in good coincidence with the common usage of the terminology *trousers, skirts,* etc. For instance trousers might be defined as a piece of garment that has a pattern X for which

$$\mathtt{isLyingOn}(X, \mathtt{leftLeg})$$

and has a pattern Y for which

$$\mathtt{isLyingOn}(Y, \mathtt{rightLeg}).$$

Whereas a skirt might be defined as a piece of garment that has a pattern X for which

$$\mathtt{isLyingOn}(X, \mathtt{bothLegs}).$$

However, in order to match the commonly used terminology one might have to use negation, e.g. to state that for a skirt there is no pattern Z which is lying around the `rightLeg`, i.e.

$$\neg\mathtt{isLyingOn}(Z, \mathtt{rightLeg}),$$

as certain trousers also have patterns, for which the positive property given for skirts is true, too.

Instead of trying to give formal definitions of these concepts it might be more useful to try to learn these concepts from the relations given by their instances by supervised learning methods.

2.4 Collections of garments

The virtual clothing usually is given by collection of garments, e.g. trousers, shirt, and jacket. The “world knowledge” that there is a reasonable order in which to dress with the different garments—e.g. that a jacket is above a shirt—can be formalized by a binary relation `isDressedAfter` on the garments.

We can define a related property `isDressedAfterP` on the level of single patterns. As a default implication we can state that the property $\mathtt{isDressedAfter}(A, B)$ for two garments A and B implies the property $\mathtt{isDressedAfterP}(X, Y)$ for all patterns X of garment A and all patterns Y of garment B.

However, there are cases in which the property `isDressedAfterP` has to be defined differently, e.g. when a pullover is worn over a shirt but the collar of the shirt should be worn over the pullover, cf. Sec. 3.1 and Figure 6.

3 Applications

3.1 Abstract modeling

By changing some values of the properties of the ontologies we have the possibility to model virtual clothing on an abstract level above the physics-based modeling layer. For instance, the order in which a virtual dressing of a collection of garments is performed has severe consequences on the resulting geometry but can be specified with our ontology by simply changing the property `isDressedAfter`.

Figure 4 shows the pre-positioning of trousers, which are given the name `trousersA`, under a shirt—given the name `shirtB`—onto a male human. In this case the

Figure 4: Pre-positioning and simulation of trousers under a shirt.

property states

$$\texttt{isDressedAfter}(\texttt{shirtB}, \texttt{trousersA}).$$

In order to reverse the setting we only have to change the property to

$$\texttt{isDressedAfter}(\texttt{trousersA}, \texttt{shirtB}).$$

Figure 5 shows the results after changing the property and restarting the pre-positioning and simulation.

Figure 5: Pre-positioning and simulation of a shirt under trousers.

In Figure 6 a pullover, which is given the name `pulloverC` below, is worn over the shirt and the trousers. By specifying in our ontology that the collar of the shirt is worn above everything else, this pattern remains on top, whereas the rest of the shirt is hiding under the pullover. Using the name `collarP` for the collar this can be formally defined by

$$\texttt{isDressedAfter}(\texttt{trousersA}, \texttt{shirtB}),$$

$$\texttt{isDressedAfter}(\texttt{pulloverC}, \texttt{trousersA}),$$

$$\texttt{isDressedAfterP}(\texttt{collarP}, X)$$

for all patterns X of the pullover. Please note that we simulated the whole shirt and not only the top of it. This can be seen for instance in the area of the legs where the trouser is bulging due to the textile material of the shirt.

In order to get robust simulations for several layers of garments we had to improve on the physics-based layer. This will be described in the next section.

Using the physics-based layer yields the desired result—if the virtual garments have a "reasonable size" for the given 3D-geometry of a virtual human.

As can be seen in reality when young children try to get dressed, it is not a priori clear which is the front part and which is the back part of the garment. We can easily simulate such a mistake of wearing a dress with the front part at the back. In Figure 7 we show the example of a simple dress, which solely consists of patterns lying on the `torsoAndLegs` hull and for which the one specifying `waist to hip point left` is changed to `waist to hip point right`. Without any further interaction the pre-positioning algorithm and the physics-based end-positioning computes a dress worn with the front part at the back.

Similarly, for garments involving sleeves we can abstractly specify whether the left sleeve is worn on the left arm or on the right arm etc.

3.2 Cloth-cloth collision detection

One problem in cloth modeling is the efficient and robust collision detection between several layers of garments. Previous approaches have used various algorithms for solving this problem, all of which were relying purely on the geometry of the cloth meshes. In our approach we are also using a standard method for self collision detection. Additionally, we use the information given by our ontology in form of the property `isDressedAfterP`. Our approach is motivated by the fact that for most garments a

Figure 6: An example of four layers of garments: A shirt, trousers, a pullover and the collar of the shirt. Although the shirt is dressed at first, its collar has the property of being on top and hence it lies above the pullover.

Figure 7: Example of a dress with its back worn at front; as a comparison the correct wearing of the dress is given on the lower right.

Figure 8: A shirt dressed above a trousers. During simulation the dressing sequence is changed. The semantic-based collision detection robustly takes care of this (from left to right).

well-defined layer can be assigned to each pattern and that this layering does not change during animation. So, if we know that a piece of garment is worn below another piece, the collision detector should know this as well.

Previous methods perform collision detections on the geometric modeling layer. Some algorithms use the history of the garment [12, 8]. These methods fail as soon as a little error due to numerical precision or other reasons occurs. After such an error is in the system, the collision detector can not resolve this anymore.

In [19] the problem of several garment layers is solved by simply removing the parts of a garment which are hidden in an initial simulation. Then, the remaining visible parts of the garments are animated in real-time. As a consequence, further interaction between garment layers is not possible.

A history-free collision response algorithm is described in [10]. By performing a global intersection analysis of the cloth mesh self-intersections can be detected and resolved. The algorithm also allows to recover from errors of a previous simulation frame. The method works quite well for closed regions of intersection. But there are cases of intersecting boundaries between meshes that the method can not resolve.

3.2.1 Semantic-based collision detection

As mentioned above our algorithm uses the layering information provided by the garment ontology to circumvent the problems of previous methods. In a pre-processing stage we assign a layer number to each mesh element. The layer numbers are derived by using the property `isDressedAfterP`. Usually, those numbers do not change during simulation. We assume that the characters body is rigid between simulation steps. So, its geometry can be treated as being static between two steps and it is used as a reference surface in the following.

During a simulation step we process all intersections in a standard way and generate adequate collision responses. Then the resulting state of the cloth mesh is checked for proper layering. For this we compare the distances to the reference surface. If a mesh element on a lower layer is farther away than a nearby mesh element on a higher layer, we have to correct this. We can simplify this part of our algorithm by just considering the relations of vertices to the other mesh elements. So, we can compute the closest feature v_{cf} on the mesh for each vertex v_i. Then we compare the distances to the reference surfaces of v_{cf} and v. If the distances do not correspond to the layering information, i.e. if v is closer to the characters body but on a higher layer, we have found a wrong layering.

For the collision response we have three options in general: We can change the position of a vertex, the position of the closest feature, or both. In our implementation we simply changed the position of the vertex by moving it closer to the character body in the direction of its surface normal. Distance values and normals can be computed efficiently by computing a distance field for the characters body at the beginning of the simulation step [20, 21, 9].

The described collision response algorithms tolerates difficult initial intersections and even a wrong dressing sequence of the garments can be corrected easily as is shown in Figure 8. We are not aware of other systems which have similar capabilities. As a drawback our method can not handle garments which are partially covered by another garment and concurrently are covering the other one, e.g. a shirt worn under a trouser can not hang over the trouser when pulled out a little bit.

4 Conclusion and Future Work

We have shown that incorporating semantics and ontologies into the simulation of virtual garments is beneficial in several aspects.

It can be used for specifying high-level properties of collections of garments in an intuitive way, from which the 3D-geometry of the garments with respect to a specific human body can be computed without any further user interaction. Simple changes in the abstract specifications might result in the desired and often considerable changes of the resulting geometries.

Also the fundamental process of collision detection between different pieces of cloth can be enhanced considerably. Whereas it is well known that the process of collision detection, which is based on the geometric layer can be enhanced in various aspects using the physics-based layer—e.g. by estimating possible accelerations or minimal curvatures—we do not know of any other work that uses information provided by higher modeling layers.

Currently, we have access to garment databases of rather small sizes only. So we could not test the assumption that the coarsening of the geometric garment information to the properties specified by our ontologies is a useful step for contend based retrieval for garments. However, as our ontologies involve discrete properties only, standard indexing and retrieval techniques [22] can be used for them so that they are suitable even for huge collections. As huge collections of CAD garment data exist and new collections are designed at a very high rate the task of retrieval of garment will become more and more important and will be a topic of future work.

References

[1] Donald H. House and David E. Breen, editors. *Cloth Modeling and Animation*. A. K. Peters, Natick, MA, USA, 2000.

[2] David Baraff and Andrew Witkin. Large steps in cloth simulation. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings,* Annual Conference Series, pages 43–54, Orlando, FL, USA, July 1998. ACM SIGGRAPH.

[3] B. Eberhardt and A. Weber. A particle system approach to knitted textiles. *Computers & Graphics*, 23(4):599–606, September 1999.

[4] Mark Meyer, Gilles Debunne, Mathieu Desbrun, and Alan H. Barr. Interactive animation of cloth-like objects for virtual reality. *The Journal of Visualization and Computer Animation*, 12:1–12, May 2001.

[5] Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. A fast finite element solution for cloth modelling. *Proceedings of Pacific Graphics*, pages 244–251, 2003.

[6] B. Lafleur, N. Magnenat-Thalmann, and D. Thalmann. Cloth animation with self-collision detection. In *Proc. of the IFIP conference on Modelling in Computer Graphics, SIGGRAPH '91*, pages 179–187, 1991.

[7] Pascal Volino and Nadia Magnenat-Thalmann. *Virtual Clothing, Theory and Practice*. Springer, 2000.

[8] P. Volino and N. Magnenat-Thalmann. Accurate collision response on polygonal meshes. In *Proc. of Computer Animation Conference*, pages 179–188, 2000.

[9] Robert Bridson, S. Marino, and Ronald Fedkiw. Simulation of clothing with folds and wrinkles. In *Proc. ACM/Eurographics Symposium on Computer Animation*, pages 28–36, 2003.

[10] David Baraff, Andrew Witkin, and Michael Kass. Untangling cloth. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003)*, 22(3):862–870, 2003.

[11] Stefan Kimmerle, Matthieu Nesme, and François Faure. Hierarchy accelerated stochastic collision detection. In *Vision, Modeling, and Visualization 2004,* 2004.

[12] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *ACM Transactions on Graphics (SIGGRAPH 2002)*, volume 21, 2002.

[13] John Funge, Xiaoyuan Tu, and Demetri Terzopoulos. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 29–38. ACM Press/Addison-Wesley Publishing Co., 1999.

[14] Arnulph Fuhrmann, Clemens Groß, Volker Luckas, and Andreas Weber. Interaction-free dressing of virtual humans. *Computers & Graphics*, 27(1):71–82, January 2003.

[15] Web Ontology Language (OWL), http://www.w3.org/2004/owl/, 2004.

[16] Hyewon Seo and Nadia Magnenat-Thalmann. An automatic modeling of human bodies from sizing parameters. In *Proceedings of the 2003 Symposium on Interactive 3D graphics*, pages 19–26. ACM Press, 2003.

[17] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. *Transactions on Graphics (ACM SIGGRAPH 2003)*, 22(3):587–594, 2003.

[18] L. Dekker, I. Douros, B. Buxton, and P. Treleaven. Building symbolic information for 3d human body modeling from range data. In *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling*. IEEE Computer Society, 1999.

[19] Frédéric Cordier and Nadia Magnenat-Thalmann. Real-time animation of dressed virtual humans. In *EUROGRAPHICS 2002*, volume 21, 2002.

[20] M. Teschner, S. Kimmerle, B. Heidelberger, G. Zachmann, L. Raghupathi, A. Fuhrmann, M.-P. Cani, F. Faure, N. Magnenat-Thalmann, W. Straßer, , and P. Volino. Collision detection for deformable objects. In *Eurographics 2004: State of the Art Report*, August 2004.

[21] Arnulph Fuhrmann, Gerrit Sobotka, and Clemens Groß. Distance fields for rapid collision detection in physically based modeling. In *Proceedings of GraphiCon 2003*, pages 58–65, September 2003.

[22] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, second edition, 1999.