

Made-to-Measure Technologies for an Online Clothing Store

Frederic Cordier, Hyewon Seo, and Nadia Magnenat-Thalmann
MiraLab, University of Geneva, Switzerland

The Internet is a compelling channel for selling garments. Several recent initiatives by companies such as Nordstrom, Macy's, and Lands End focus on made-to-measure manufacturing and shopping via the Internet. Current Web technologies fuel these initiatives by providing an exciting and aesthetically pleasing interface to the general public.

However, until now, such Web applications have supported only basic functions such as viewing apparel items in 2D or 3D, combining different items together, and mixing and matching colors and textures (and sometimes using a mannequin adjusted to the shopper's proportions). The most common problems customers experience when they try on the clothes are poor fit, an unpleasant feeling while wearing the item, and surprise at the garment's color. As a result, high product return rates persist, and most consumers are still either hesitant to purchase garments online or are unsatisfied with their online shopping experience.¹

Here, we present a Web application that provides more powerful access to and manipulation of garments to facilitate garment design, pattern derivation, and sizing. (You can visit the virtual Try-On application at <http://virtual-try-on.miralab.unige.ch>.) We apply 3D graphics technology to help create and simulate the virtual store. In this article, we discuss various relevant research problems, the creation of the shopper's body and garments, simulation of body and garment movement, and online sizing.

Our system supports many efficient and interactive operations such as automatically adjusting the 3D mannequin according to the shopper's body measurement, selecting different garment items, online fitting and resizing of the garment to the mannequin, and simulating garment movement in real time. We ultimately hope to develop and integrate several key technologies

into a distributed, interactive virtual clothing store where customers can choose garments, use 3D mannequins that are adjusted to their body measurements, and receive assistance during their online purchase.

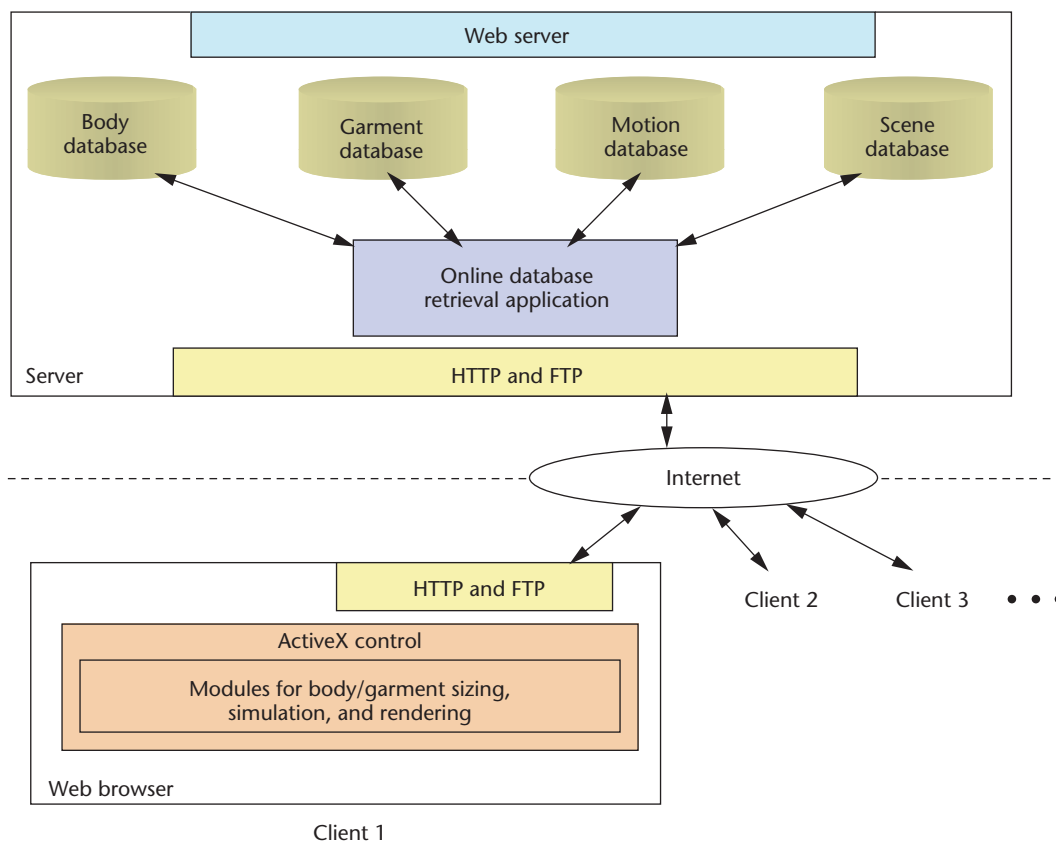
System overview

An online clothing store designed as a Web application requires flexible manipulation, fast transmission, and efficient storage of the display content. In particular, because of the relatively huge database of garments and the simulation of complex graphical objects such as skin and cloth needed for such an application, the most critical limitation is the real-time performance constraint. When we simulate garment movement using a physics-based model, the results must be prerecorded to display the animated garment at an interactive rate because it's impossible to achieve real-time performance.²

Although this approach simplifies the online computation, it requires us to transfer, for each frame, the position data of each vertex constituting the visual representation of a garment, increasing the response time of the client application. In addition, it's time-consuming to simulate each 3D garment item in the database. Typically, it takes about 4 to 12 hours to simulate the garment movement of a 1-minute sequence depending on the 3D geometry's complexity. In our application, we choose a better alternative: we calculate the garment simulation on the fly while keeping the response time interactive.

Human body modeling and simulation is another key technology enabling automated garment sizing and size selection. Since the advent of 3D image capture technology, we've seen much interest in applying this technology to taking human-body measurements. Today, some systems are optimized either for extracting accurate measurements from parts of the body, or for realistic visualization in various fields (including e-commerce applications). Cyberware Inc.'s DigiSize, for instance, was developed in a joint government project to improve and automate fitting and issuing military clothing (see <http://www.cyberware.com/products> for more information).

We present a Web application that provides more powerful access to and manipulation of clothing to facilitate clothing design, pattern derivation, and sizing.



1 Web application architecture of our online clothing store.

Despite recent efforts devoted to using 3D body scanners, limiting factors such as the inability to automatically integrate body scan data to application software exist. One important feature of our online clothing store is its ability to build a 3D mannequin according to the body measurements the user inputs. This led us to consider another body modeling scheme. We perform the online construction of the 3D mannequin that satisfies the given measurements, avoiding scanners and downloading large data models. Despite the apparent difficulty in building the body geometry from the limited amount of information within the real-time constraint, we show how this process is feasible through our robust modeling technique for practical applications. Moreover, with our approach, we can interactively modify the measurements and body motion.

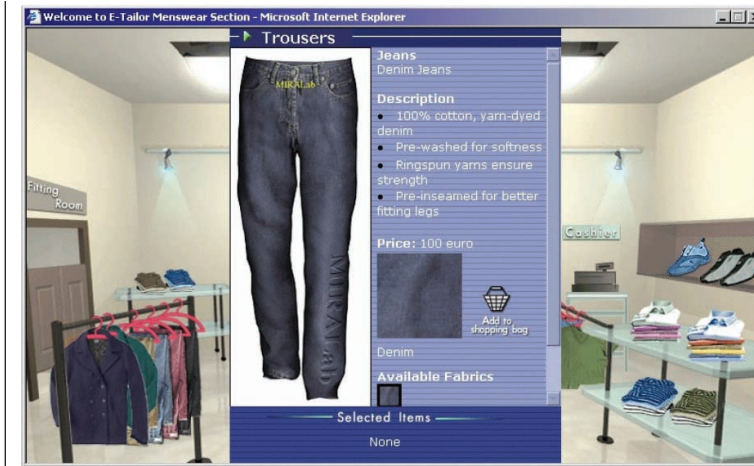
Figure 1 shows the system architecture. Our system has a minimal response time because a major part of the content to be manipulated is generated on the client side rather than on the server. Computing the body and cloth animations on the server and sending this data through the Internet would generate too much traffic and reduce the application's response time. Thus, our solution is to move the body and garment sizing as well as the cloth and skin animation to the client side, avoiding the download of large precalculated models.

The server

The online clothing store Web server consists of the following databases and an online database retrieval application module for retrieving data:

- *Body database.* This database contains two 3D mannequins for each gender, which we refer to as *generic models*, plus certain statistical information that's collected from existing models through 3D shape capture technologies. This is essentially the information needed to derive new body geometries from measurement inputs by the body- and garment-sizing module.
- *Garment database.* We created a set of 3D garment models for the generic models and categorized them. They're available to the user in the garment catalog pages (see Figure 2, next page). Ideally, the online clothing shop will feature many different garments. Moreover, the online store will make it easy to update the database to keep the coherence between the clothes shown on the Web application and the clothes available for sale. Therefore, this garment database is located on the server side. Upon user selection, the system downloads the chosen 3D garment model to the client. These garments are saved into a Virtual Reality Modeling Language (VRML) format.
- *Motion database.* The animation database contains samples of body motion data. Like the garment data, the system can download the selected motion data upon request. We obtain the motion data through the prerecording of a real person's movement using an optical motion capture system.
- *Scene database.* Graphical elements that compose the background scene are stored as VRML files in the scene database.

2 The catalog window.



body models, and what data we need for the runtime body and garment sizing.

Use of measurements

The application uses apparel designers' eight primary measurements (listed in Table 1). We assume that the users are aware of their own measurements or the measurements have been retrieved from 3D scanners or tapes.

Generic models

Instead of constructing a new geometry for each model, we assume that the model's topology is

known a priori and shared by all resulting models. The idea behind this is to exploit the objects' common structure. Aside from making the problem simpler and the modeling faster, deforming the existing model to obtain a new one makes it easier to immediately animate the newly generated model.

The generic model consists of a standard human skeleton structure (see the Humanoid Animation Working Group, or H-Anim, specification at <http://www.h-anim.org>) and a textured skin surface of approximately 6,000 vertices and 10,000 triangles. There are two models, one for each gender. The generic models' preparation consists of three successive procedures:

1. We create and properly adjust an H-Anim skeleton to the body mesh using the Discreet 3ds max tool.
2. We then calculate the skin attachment data using BonesPro (see <http://www.digimation.com>).
3. Using our in-house software, we segment the body mesh and export the whole body structure into the H-Anim format.

Table 1. Measurement definitions.

Body Measurement	Definition
Stature	Vertical distance between the crown of the head and the ground
Crotch length	Vertical distance between the crotch level at center of body and the ground
Arm length	The distance from the armpit and shoulder line intersection over the elbow to the far end of the prominent wrist bone in line with the small finger
Neck girth	The girth of the neck base
Chest/bust girth	Maximum circumference of the trunk measured at bust/chest height
Underbust girth	Horizontal girth of the body immediately below the breasts
Waist girth	Horizontal girth at waist height
Hip girth	Horizontal girth of the trunk measured at hip height

The client

Our Web client performs program execution after the executables are downloaded to the client as an ActiveX control. There are two main modules:

- The body and garment sizing module provides functionalities to deform the 3D mannequin from the customer's input body size and resize the selected garment accordingly.
- Once the garments and the body have been resized, the real-time garment simulation module animates the dressed mannequin. This approach drastically reduces the amount of data transferred from the server to the client.

Body database

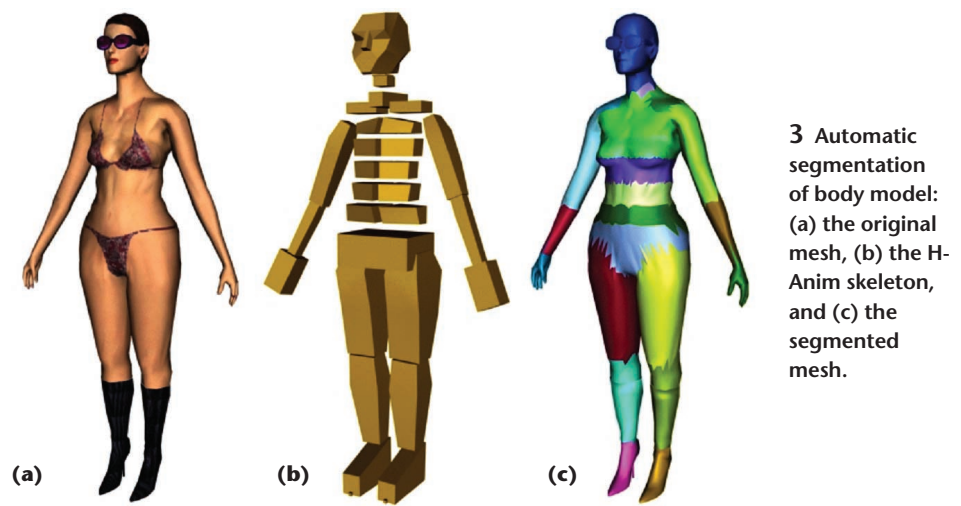
Our measurement-driven body modeling approach primarily consists of a preprocessing phase and a runtime phase. The body database stores the preprocessing results that are fed into the runtime module. Here we discuss the preprocessing phase by describing what measurements we used, how we prepare the generic

The skeleton hierarchy. The H-Anim standard specifies a common way of representing humanoids in VRML97. The human body consists of a number of segments (such as the forearm, hand, and foot), which are connected to each other by joints (such as the elbow, wrist, and ankle). The full H-Anim hierarchy contains 94 skeleton joints and 12 skin segments including the head, hands, and feet. In our application, we chose the H-Anim Level of Articulation (LOA) 2 skeleton as Figure 3b shows.

Segmentation and exportation. The skeleton-driven deformation,³ a classical method for the basic skin deformation, is the most widely used technique in 3D character skin deformation. A prerequisite of a successful skeletal-driven deformation is an appropriate attachment of the skin to the underlying skeleton. For each vertex of the mesh, the attachment data contain the vertex's influencing bones and corresponding weights. To say that a vertex is "weighted" with respect to a bone means that the vertex will move as the bone is

transformed to stay aligned with it. In our current implementation, we do this through BonesPro.

Once we've defined the attachment, we need to segment the skin mesh and locate each of the segments into the skeleton hierarchy as a proper child node of the corresponding joint. We decompose the mesh into a collection of adjacent triangles that share the joint of the highest weight, forming segment nodes of the corresponding joint nodes (Figure 3c). Now we can export the model into a H-Anim-compliant human body. We export the attachment data as XML data in the VRML file.



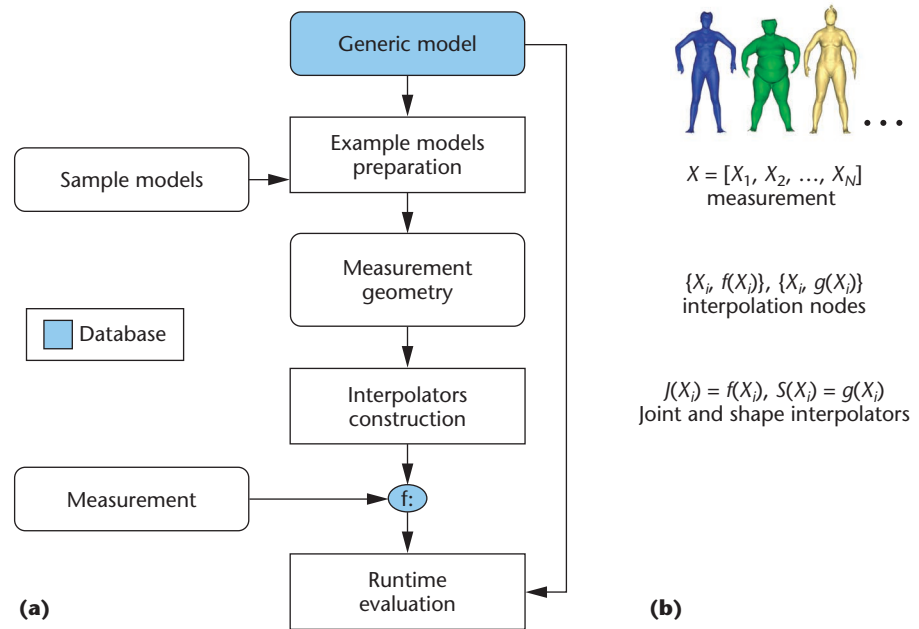
3 Automatic segmentation of body model: (a) the original mesh, (b) the H-Anim skeleton, and (c) the segmented mesh.

Example-based approach.

Mere geometric consistency of measurements doesn't guarantee a reasonable body shape appearance. Arguably, the captured body geometry of real people provides the best available resource to model and estimate correlations between measurements and the shape. We exploit correlations among measurements and between measurements and the body shape. This permits a robust modeling of body geometry even when only a limited amount of information is available.

Figure 4 shows the body modeling process. It involves two preprocessing tasks. First, we establish topological equivalence among these data sets (example models hereafter) so that a complete mapping can be found for each vertex in the mesh and the skeletal structure. Based on the prepared samples and their measurements, the second task implements interpolators for evaluating deformations necessary to obtain new models in the continuous range in the measurement space. The online execution of the body- and garment-sizing module uses the interpolator function to deform the generic model on the fly through multiple blending of examples.

Example model preparation. The initial models or examples rely on existing models or 3D shape-capture technology. In addition, we assume the correspondence is established among the models through a preprocessing phase. Each model must include the same number of vertices and connectivity in the mesh as well as the same joint hierarchy. To accomplish this, we fit the generic model onto each example. The detail description of our method to obtain correspondence among example models is beyond the scope of this article. Hav-

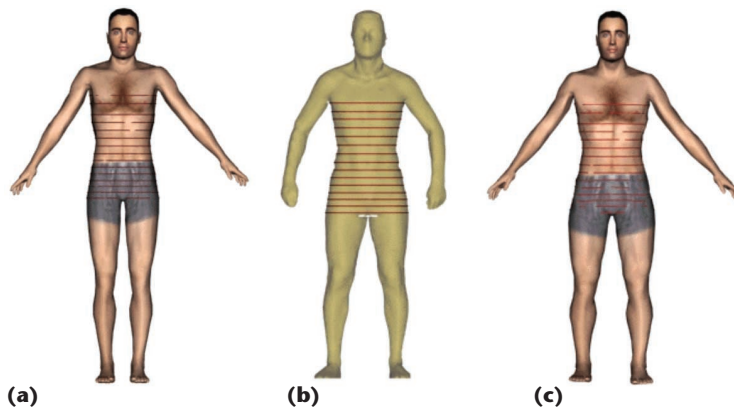


4 (a) Body modeling scheme, and (b) interpolator function development.

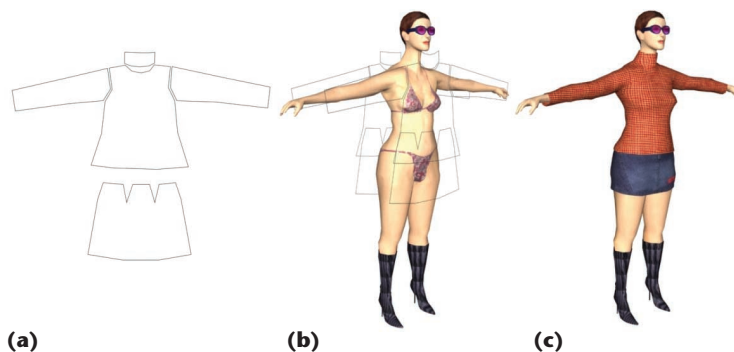
ing a predetermined topology has both theoretical and practical merits. First, it simplifies the problem by allowing a vector representation of the body geometry. Second, the skin attachment data can be reused so that we can immediately animate all resulting models. Finally, we easily resolve different postures of examples by taking only translation and scale parameters from the examples.

We used 14 models (six male and eight female) from the 3D scanner in our application. After establishing the correspondence, we measured the models following the definitions in Table 1 to obtain eight primary measurements. The measured values of each model, together with its joint and shape parameters (see the "Interpolators construction" section), serve as input to the interpolators.

We consider the body geometry as having two distinct entities, namely rigid and elastic components of the



5 Fitting a generic model to an example. (a) The generic model. (b) An example model. (c) After the fitting.



6 Creating garments. The system (a) constructs a garment pattern and (b) places it on a 3D virtual body. (c) Then, it computes the initial shape of the garment through a collision response.

body deformation. The joint parameters of a person, which will determine the global proportion of the physique, represent the rigid component. Then, we use shape parameters to express the elastic component, which, when added, depicts the body's detail shape. We define the joint and shape parameters as follows:

- *Joint parameters.* The joint parameters are essentially each joint's degree of freedom (DOF). A joint has scale (s_x, s_y, s_z) and translation (t_x, t_y, t_z) , leading to a vector of $6N$ dimension, where N is the number of joints.
- *Shape parameters.* As Figure 5 shows, we use a set of contours to represent the detail shape of the body geometry, particularly on the torso. Note that many of the primary measurements are based on contours. The elastic deviation of the contour vertex through the fitting process from its initial position represents the deformation's elastic component.

For a compact representation of both parameters, we adopt the principal component analysis (PCA) approach, one of the common techniques to reduce the dimensionality of data in a statistical manner. See Press et al.⁴ for an extensive discussion of this technique.

Interpolator construction. Once we prepare the example models, we then build interpolators that will

be used to evaluate deformations necessary to obtain new models by the body- and garment-sizing module. We consider two interpolators—joint and shape. Given a particular measurement set, joint interpolators are responsible for the body deformation's rigid component, guiding the skeleton-driven deformation to obtain the appropriate proportion of each body part. The shape interpolators deal with the elastic deformation, adding detail shape on the body. Dealing with high-dimensional data and a relatively small (less than 10) number of examples, a scattered interpolation problem best describes our problem. As a result, we implemented joint and shape interpolators as functions of the eight primary measurements by using the radial basis function (RBF).

Interpolator exportation.

Once the system has generated the interpolators, we can quickly explore the measurement space to evaluate the deformation's rigid and elastic components, allowing runtime generation of a customized 3D body model. Implemented as an RBF, each interpolator is described by its interpolation nodes and the nodes' corresponding weights. To speed up the runtime evaluation, we implemented the basis function by using a lookup table.

Garment database

We calculated a set of garments for the generic model, which reside in the garment database. For creating clothes usable in the online clothing store, designers must draw the patterns, preprocess the garments, and export them into VRML. We developed several authoring tools to assist designers in their work.

Design of garments

Designers create garments using our in-house software.³ This tool assists designers in drawing 2D patterns and defining seam lines on the borders of the garment patterns, referring to the polygon edges that are to be joined during the garment construction process as Figure 6a shows. The system tessellates the patterns into a triangular mesh and places them around the 3D virtual body (see Figure 6b). Next, the system computes the initial shape of the garment through a collision response, as illustrated in Figure 6c. The body model's shape guides the cloth's surface as a result of the collision response. Because the body- and garment-sizing module handles the garment sizing online, each garment item needs only one simulation for the generic model.

Garment preprocessing

Simulating garments in real time requires drastically simplifying the simulation process, possibly at the expense of mechanical and geometrical accuracy. Our approach⁵ is based on a hybrid method where the cloth is segmented into various sections and different algorithms are applied. When observing a garment worn on a moving character, we notice that the movement of the garment can be classified into several categories depending on how the garment is placed—that is, whether it sticks to, or flows on, the body surface. For instance, a tight pair of trousers will mainly follow the movement of the legs, while a skirt will flow around the legs. Thus, we segment the cloth into three layers that we define as follows:

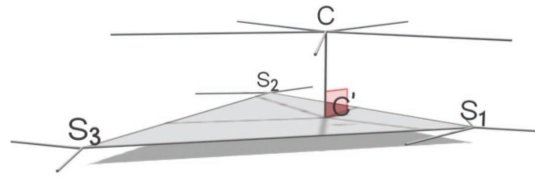
- **Layer 1: Stretch clothes.** Garment regions that stick to the body with a constant offset. In this case, the cloth exactly follows the underlying skin surface's movement.
- **Layer 2: Loose clothes.** Garment regions that move within a certain distance to the body surface are placed in another category. The best examples are shirtsleeves. In this case, we assume that the cloth surface always collides with the same skin surface and its movement is mainly perpendicular to the body surface.
- **Layer 3: Floating cloth.** Garment regions that flow around the body. The cloth movement doesn't exactly follow the body movement, and collisions aren't predictable. For example, for a long skirt, the left side of the skirt may collide with the right leg during animation.

These three categories are animated using three different cloth layers. The idea behind the proposed method is to avoid the heavy calculation of physical deformation and collision detection wherever possible. The main interest of our approach is to preprocess the target cloth and body model so that they can be computed efficiently during runtime. The garments are divided with our in-house software into a set of segments and the associated simulation method is defined for each.

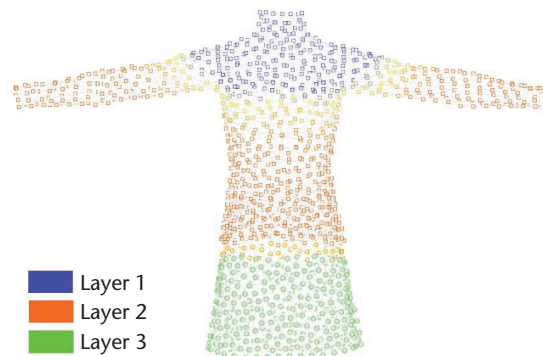
Computing cloth attachment data. Defining attachment data is one step of the preprocessing stage. As we stated in the previous section, the cloth deformation method uses the underlying skin's shape. We use the underlying skin deformation for the simulation of the first two layers of the cloth. Each garment mesh vertex is associated to the closest triangle, edge, or vertex of the skin mesh.

In Figure 7, the garment vertex C is in collision with the skin triangle $S_1S_2S_3$. We define C' as the closest vertex to C located on the triangle $S_1S_2S_3$. We then define the barycentric coordinates of C' with S_1 , S_2 , and S_3 . These barycentric coordinates are later used for cloth simulation and sizing. They provide an easy way to compute the garments' "resting" shape by using the location of the skin vertices.

Segmentation. With the garment in its resting shape on the initial body, we use the distance between



7 Mapping of attachment information. Here the garment vertex C is in collision with the skin triangle $S_1S_2S_3$.



8 Segmentation of garments. In layers 1 and 2, cloth vertices are located closely to the skin surface while in layer 3, the vertices don't collide with any skin surface.

the garment and the skin surface to determine which category the cloth triangles belong. Associated with each segment are distances from the skin surface that are used to determine the category. Each segment falls into one of three categories: tight, loose, and floating clothes. Cloth vertices that are located closely to the skin surface belong to the first or second layer. Cloth vertices that don't collide with any skin surface belong to the third layer as Figure 8 shows.

Exportation. Once segmented, we can export 3D cloth models and the associated data into VRML for use in the online clothing store. These models are located on the server, allowing easy update and maintenance.

Motion database

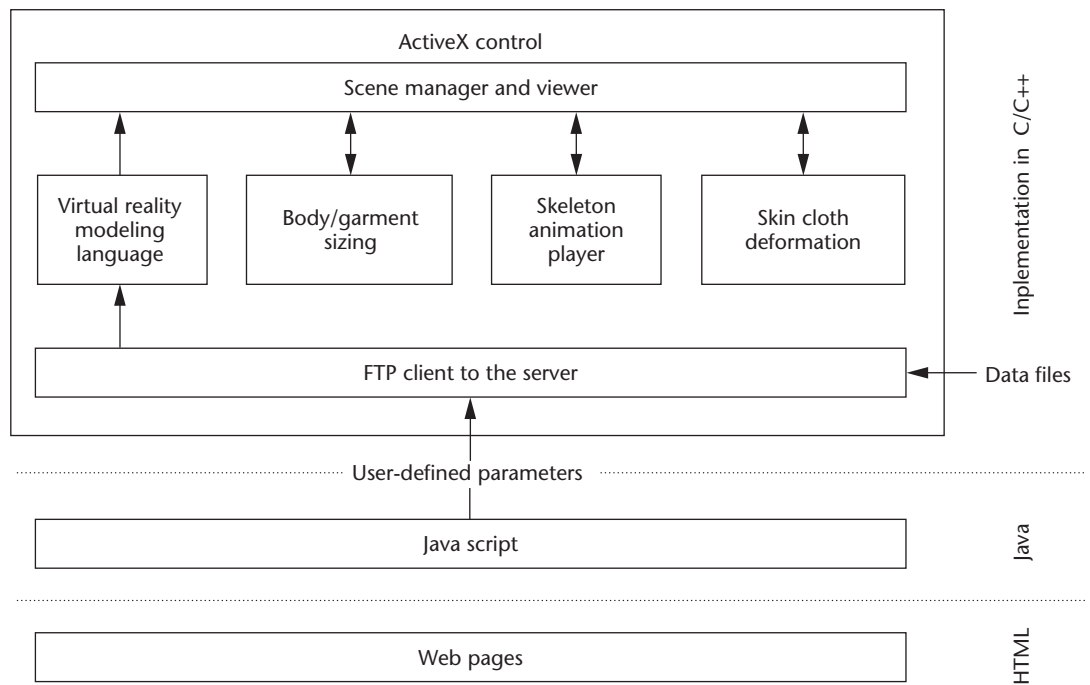
Commercially available motion capture systems offer a relatively easy way of recording a human performer's movement. Our system converts the animation data so that it's immediately applicable to H-Anim models. The converter first computes the correspondence between the two skeleton hierarchies (one from the Vicon [<http://www.vicon.com>] motion capture system and the other from H-Anim) and performs the transformation on each joint angle for each frame of animation to resolve the difference in their stand-by posture.

The converted animation data is exported in VRML format using the Interpolator node and can be applied to the H-Anim body model in a frame-by-frame basis. The database consists of several typical motions that people do when trying on clothes, such as walking and turning.

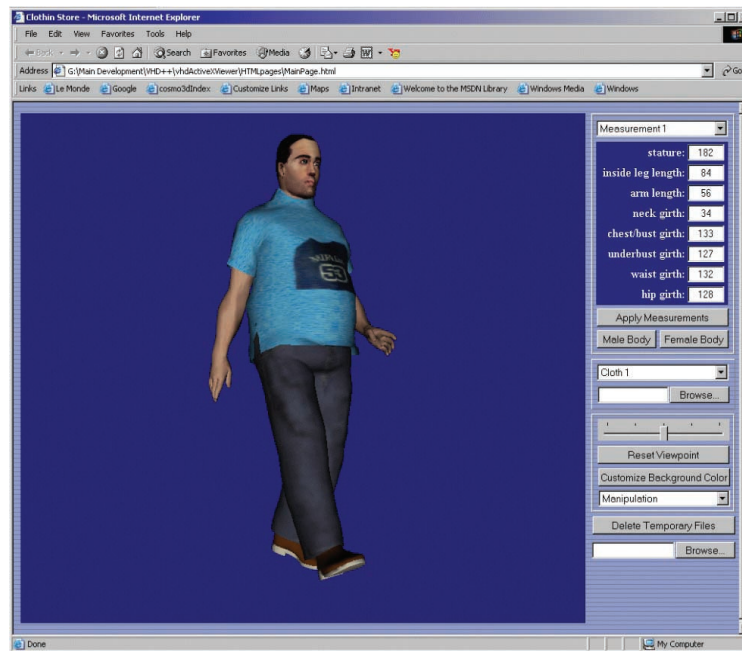
Integrating the client

The client application isn't only involved in the visualization of garments, but it also calculates the cloth and

9 Overview of the client architecture.



10 Screen shot of an ActiveX viewer with the HTML and JavaScript user interface.



body deformation. As Figure 9 shows, the client architecture uses three different layers for the implementation: C++, JavaScript, and HTML.

The ActiveX control

We developed modules for real-time animation and visualization in C++ and integrated them into an ActiveX control. ActiveX controls are components that can be embedded within Web browsers like Microsoft Internet Explorer. Because they can be written in any language, they offer the best performance for time-critical applications. Another advantage of ActiveX controls is that

their installation on the client machine is transparent to the user. They're automatically downloaded and installed at the first access to the Web page.

Our ActiveX control consists of several modules: the FTP client for downloading data from the server, the VRML loader that's in charge of loading the data into the scene manager, the skeleton animation player for animating the mannequin skeleton, the body and cloth sizing module for fitting the body and clothes to the user measurements, the skin and cloth deformation module for real-time animation, and the 3D viewer for visualization of the scene.

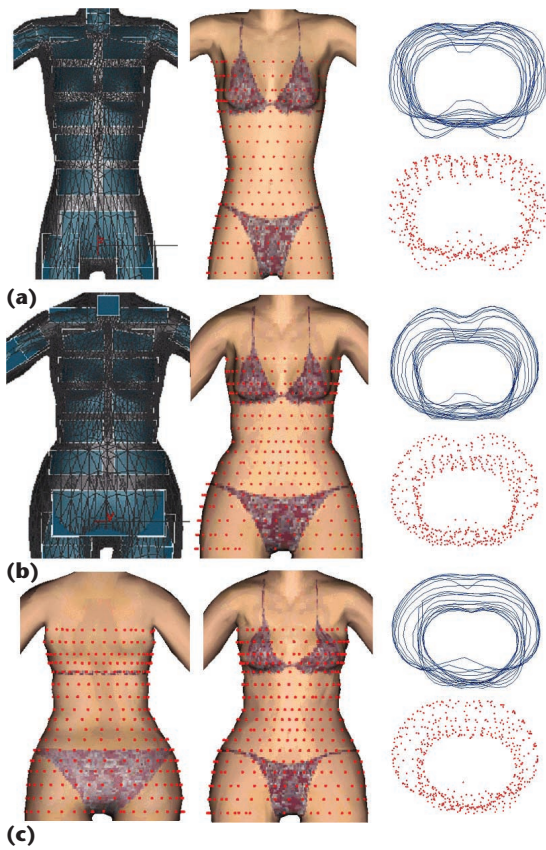
Javascript and HTML pages

We used JavaScript and HTML to implement the graphical user interface. They provide a good solution for user interaction. JavaScript

allows complex functionalities such as keeping track of the user choices, managing widgets, and sending the user defined parameters to the ActiveX control. Figure 10 shows a Web page where the user enters the body measurements and visualizes the 3D mannequin animation.

Body- and garment-sizing module

The body- and garment-sizing module's main task is to manage the proper sizing of the 3D mannequin in a VRML scene. As we described, the module uses the generic model and the interpolators to evaluate the joint and shape parameters to deform the generic model. It



11 Skin deformation through interpolation. (a) The initial reference model. (b) After the skeletal deformation via joint interpolation. (c) After adding detailed shape via shape interpolation.

first deforms the body model by applying the evaluated parameters from the interpolators as a function of the measurement input. The garments are then deformed accordingly so that they fit the deformed body.

Body sizing

Given the measurement values or a specific location in the measurement space chosen by the application at runtime, the interpolator evaluates the necessary deformation by efficiently blending the examples with known measurements to produce an interpolated shape.

Contour warping. Given the new set of measurements, the system determines the location in the measurement space. Using this vector as input, the system evaluates the joint parameters from the joint interpolators for each joint in the skeleton. Figure 11b shows the model after the joint parameters are applied to the model shown in Figure 11a. Similarly, the shape interpolators determine the shape of the primary and auxiliary contours consecutively and deform them by adding the evaluated vertex displacements as Figure 11c shows.

Mesh deformation. To compute a skin surface that smoothly interpolates between deformed contours, we again use the scattered data interpolation technique.

For each point p_i on the skin mesh, the final location is given by

$$p'_i = p_i + W(p_i) \quad (1)$$

where $W = (s_x, s_y, s_z)$ denotes the deformation function and $W(p_i) = (\Delta p_{ix}, \Delta p_{iy}, \Delta p_{iz})$. For n points p_1, p_2, \dots, p_n of the contour set, we have $W(p_i)$ from the shape interpolators. Then we can find the coefficients of the Gaussian function interpolants s_x, s_y, s_z by solving a linear system of $3n$ equations:

$$\begin{aligned} \sum s_x(p_i) &= \sum \omega_i^x \cdot \Phi(\|p - p_i\|) \\ \sum s_y(p_i) &= \sum \omega_i^y \cdot \Phi(\|p - p_i\|) \\ \sum s_z(p_i) &= \sum \omega_i^z \cdot \Phi(\|p - p_i\|) \end{aligned} \quad (2)$$

Once the interpolators have been calculated, we deform the mesh by displacing all points on the mesh according to the resulting values.

Refinement. Some joints are directly related to the input (stature, the crotch length, and the arm length) measurements, and the evaluated joint parameters from the interpolators need to be adjusted accordingly. The refinement phase of our approach measures the model from the interpolators and adjusts the skeleton to ensure the user-specified measurement constraints on the model.

Garment sizing

After the body model is properly deformed from the given measurements, the garment-sizing module modifies the garments so that they fit this new body. As we explained earlier, the cloth mesh is segmented into three layers—two layers where the cloth vertices are attached to the body surface and one layer where the garments move freely. Thanks to the cloth attachment data (Figure 7), we can easily compute the position of vertices belonging to the first two layers from the position of the underlying skin vertices weighted with the barycentric coordinates. Cloth vertices on these two layers follow the skin surface, even in the case where the skin is deformed by the body-sizing module. For the sizing of garments belonging to the third layer such as skirts or dresses, the diameter and the length of legs are calculated in the body deformation module. The length and the width of the garments are scaled accordingly to the new measurements of the legs. Figure 12 (next page) shows the body and garment sizing results.

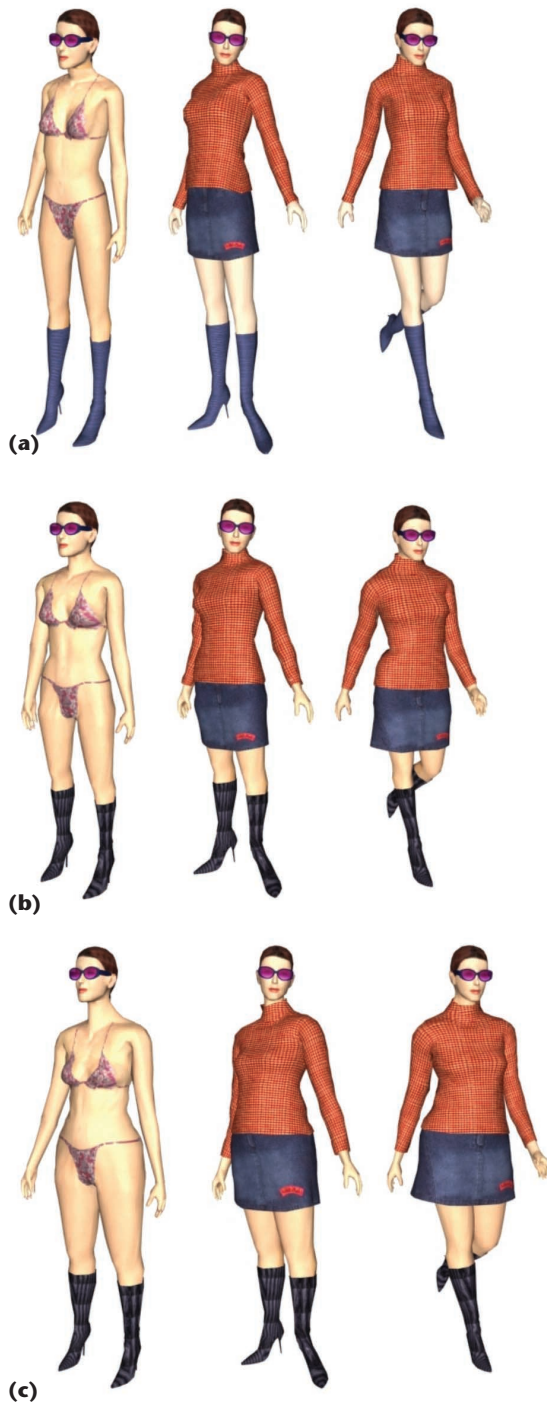
Skin and garment simulation module

After generating the dressed body according to the user measurements, the motion data applied to the H-Anim skeleton drives the skin and cloth deformation. This section addresses the computation of the real-time simulation of bodies and clothes according to the animation of the underlying skeleton.

Joint-driven deformation of skin

We use the skeletal driven deformation technique, introduced in the section “Segmentation and exporta-

12 Simulation of bodies and garments for different sizes: (a) small, (b) medium, and (c) large.



tion” for simulating the skin deformation of the virtual human. At each frame of the animation, we calculate the vertices’ position using the weight values and the transformation matrix of the joints. We also use these weight values to compute the mesh surface’s normal. The movement of vertices that belong to a single joint isn’t calculated but automatically moved as they’re attached to the joint. Duplicated vertices that lie on the boundaries have the same position because they share the same attachment information. Thus, the boundaries among segments aren’t visible. The result is a segmented body that appears as a seamless body in the render-



13 A sequence of real-time cloth animation.

ing view port. This method combines the speed of the deformation of segmented bodies with the visual quality of seamless bodies.

Garment simulation

We previously⁵ proposed techniques for a real-time clothing simulation (see Figure 13). Garment vertices are animated with three different methods, depending on which layer they belong to that’s defined during the preprocessing stage.

Layer 1. Tight clothes in layer 1 follow the deformation of the underlying skin. These deformations are calculated with a geometric-based method thanks to the mapping of the attachment data of the skin to the garment surface.

Layer 2. For layer 2, which consists of loose clothes, the relative movements of clothes to the skin remain relatively small, keeping a certain distance from the skin surface. Consider the movement of a sleeve in relation to the arm—for a certain region of the garment, the collision area falls within a fixed region of the skin surface during simulation. With this in mind, the scope of the collision detection can be severely limited. We assume that the movement of the garment largely depends on the underlying skin and yet it shouldn’t follow the skin surface rigidly. It’s necessary to simulate the local displacement of the garment from the skin surface.

We’ve developed two different methods, one for cloth deformation on the limbs (trousers and sleeves), the other one for the deformation of cloth on the trunk. Cloth vertices on the limbs are enclosed in half spheres that are attached to the skin surface. Vertices inside these spheres are displaced with the equation of the rigid body motion. A function defines the diameter of the spheres depending on the relative position of the cloth vertex to the skin surface’s normal. We animate cloth vertices located on the trunk with a rough mesh and animate the rough mesh with a physics-based method. The cloth mesh is deformed with the free-form deformation (FFD) method using the position of the vertices on the rough mesh.

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

Results and discussion

Figure 14 shows the main graphical user interface (GUI). After user authentication, the system establishes communication, and the user downloads the ActiveX control if it hasn't already been installed on the machine. At this point, initialization of the control is executed, including the download of the data for the body- and garment-sizing module. The visitor to the online virtual store follows several successive steps:

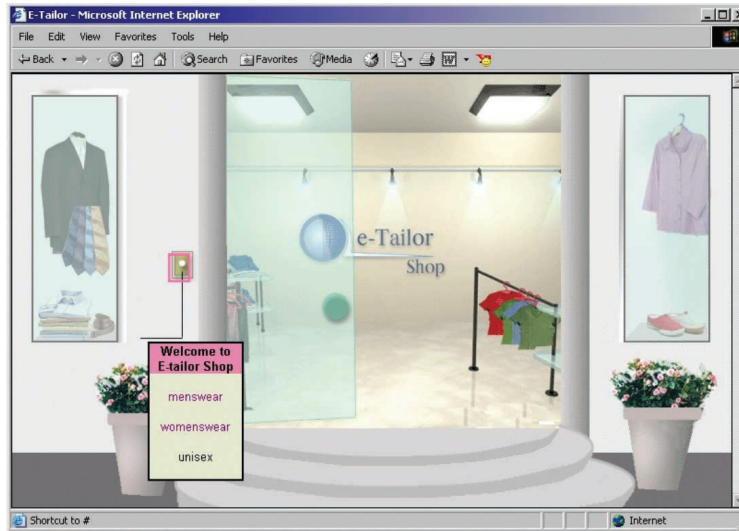
1. After users select the garment item through the 2D garment catalog, their selection is passed to the ActiveX control through JavaScript functions.
2. The VRML loader downloads the requested garments and motion data and loads them into the scene.
3. Users continue to the body measurements page containing a number of fields that are adjustable. After the users enter their measurements, the body and garments are resized accordingly.
4. Finally, a trigger action starts the execution of the real-time simulation. While the skeleton animation player updates the angle values of the mannequin's skeleton, the skin and clothes are deformed accordingly.

The time needed for the simulation is obviously linked to the complexity of the body and the garment model. Table 2 shows the average size of the different data to be downloaded for a simulation sequence.

The largest amount of data to be downloaded is the body database. Fortunately, this database needs to be downloaded only once at the initialization stage of the ActiveX control. Users need to download other data every time they change the garments and/or motion data. The frame rate of the real-time animation of the mannequin in the ActiveX control is about 20 to 30 frames per second. See Cordier and Magnenat-Thalmann⁵ for a further discussion of the performance of the body and garment deformation module.

Future directions

In our work, we've solved various research problems to build a framework for an online clothing store. The process of preparing various data components has matured, making it feasible to efficiently handle the creation of graphical contents, which are immediately usable for interactive



14 The online clothing store front page.

Table 2. Average size of downloaded data.

Data	Average size in Mbytes
Body data with generic textured model (8,000 polygons)	2.0
Motion data (duration 10 seconds)	0.2
Garments with textures (4,000 polygons)	1
Scene with textures (1,000 polygons)	1

visualization on the Internet. The bodies and garments created with our approach correspond to real body measurements at the critical locations defined for the garment design. In particular, we've developed methods for efficiently handling the online body modeling, garment selection, fitting, and simulation and integrated them into Web components, optimized for execution on the Internet. We also set up an interactive and realistic virtual clothing store, where visitors can choose among many different types of garments and visualize them on animated mannequins based on their measurements.

Nevertheless, there's still room for improvement. The sizing works only for the garment and the body, so the simulation of differently sized garments on the same body remains as future work. We also plan to enhance the input parameters to model the 3D mannequin so that it allows the best possible correspondence to a particular individual. ■

Acknowledgments

We'd like to thank the Athens Technology Center from the E-Tailor project (see <http://www.atc.gr/e-tailor>) for providing some of the Web pages of the online clothing store. Special thanks goes to Dimitris Protopsaltou, Yi Wu, and Georgios Papagiannakis, who put a lot of effort into the interface design and system integration. We're also grateful to Chris Joslin for proofreading this article, Marlene Arevalo and Christiane Luble for providing reference models, and Tom Molet for his assistance with the motion capture.

The European Research Project E-Tailor (Information Society Technologies project IST-1999-10549) supported this research.

References

1. B. Beck, "Key Strategic Issues in Online Apparel Retailing—The Need For An Online Fitting Solution," Techexchange.com, http://www.techexchange.com/thelibrary/online_fit.html, 30 June 2001.
2. D. Protopsaltou et al., "A Body and Garment Creation Method for an Internet-Based Virtual Fitting Room," *Advances in Modeling, Animation and Rendering*, J. Vince and R. Earnshaw, eds., Springer-Verlag, 2002, pp. 105-122.
3. P. Volino and N. Magnenat-Thalmann, *Virtual Clothing—Theory and Practice*, Springer-Verlag Book Series, 2000.
4. W.H. Press et al., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.
5. F. Cordier and N. Magnenat-Thalmann, "Real-Time Animation of Dressed Virtual Humans," *Proc. Eurographics 2002*, vol. 21, no. 3, Blackwell Publishers, 2002, pp. 102-110.



Frederic Cordier is a PhD candidate at the University of Geneva's MiraLab. His current research interests include real-time animation of virtual humans, particularly real-time clothes, skin deformation, and interaction among clothes and skin.

His research takes place in the framework of several European projects: *Virtual*, *E-Tailor*, *Just*, and *Cahrisma*. He studied at University Lyon-2 in Lyon and received an MS in computer graphics.



Hyewon Seo is a research assistant and PhD candidate at the University of Geneva's MiraLab. Her research interests are in virtual human modeling, face and body animation, real-time graphics for virtual reality, and optimization. She obtained her BS and MS in computer science from Korea Advanced Institute of Science and Technology. She is a student member of the IEEE.



Nadia Magnenat-Thalmann has pioneered research into virtual humans over the last 20 years. In 1989, she founded MiraLab at the University of Geneva. She obtained several bachelor and master degrees in various disciplines and a PhD in quantum physics from the University of Geneva.

Readers may contact the authors at MiraLab, University of Geneva, 24, rue du General Dufour, CH-1211 Geneva, Switzerland, email {cordier, seo, thalmann}@miralab.unige.ch.

For further information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.

IEEE Computer Graphics AND APPLICATIONS

Editorial Calendar

January/February Web Graphics

With the popularity of the Internet, we're seeing a migration of traditional applications to run on the Web environment and a growing demand for more powerful Web-based applications. Fused by the increasing availability and dramatic reduction in the cost of 3D graphics accelerators, a new direction of research, called Web Graphics, is emerging. This includes developing graphics applications as well as tools to support these applications in the Web environment.

March/April Graphics Applications for Grid Computing

Grid computing allows access to distributed computing resources with the same ease as electrical power. In recent years, graphics application tools that take advantage of distributed computing, or grid environments, have emerged. New methodologies and techniques that harness resources for graphics applications are critical for the success of grid environments.

May/June Advances in Computer Graphics

This issue covers an array of advances in computer graphics such as organic textures, lighting, and approximation of surfaces. Also, you'll find out about new developments in virtual reality, novel approaches in visualization, and innovative CG applications. The range of topics highlights the usefulness of computer graphics for everyone.

<http://computer.org/cga>

July/August Nonphotorealistic Rendering

Nonphotorealistic rendering (NPR) investigates alternatives that leverage techniques developed over centuries by artists and illustrators to depict the world. One goal of this research is to broaden the achievable range of image styles and thereby embrace new applications. Additionally, NPR has the potential to open a new line of attack on one of the central problems of 3D computer graphics today: content creation.

September/October Perceptual Multimodal Interfaces

This issue focuses on recent advances in methods, techniques, applications, and evaluations of multimodal interaction. Learn how researchers' cross-disciplinary approaches helped develop multimodal interfaces from interaction-centered prototypes to user-oriented and application-tailored solutions. This issue also explores the notion of moving toward transparent user interfaces.

November/December 3D Reconstruction and Visualization

Models based on 3D data will ultimately include everything associated with the environment, such as trees, shrubs, lampposts, sidewalks, streets, and so on. The main mode of exploration for this massive collection will be through interactive visualization. Ultimately, you should be able to fly continuously from overviews of a large city to centimeter-size details on the side of any building. Smoothly joining these different scales may require integrating rendering techniques in new ways.