Emerging Web Graphics Standards and Technologies

Rynson W.H. Lau and Frederick Li
City University of Hong Kong

Tosiyasu L. Kunii
Hosei University, Japan

Baining Guo and Bo Zhang
Microsoft Research Asia

Nadia Magnenat-Thalmann and Sumedha Kshirsagar
University of Geneva, Switzerland

Daniel Thalmann and Mario Gutierrez
Swiss Federal Institute of Technology (EPFL)

Migrating computer graphics to the Web poses several problems, but with new standards and technology advances, graphics applications can balance latency and bandwidth constraints with image quality.

Migrating graphics applications and techniques to the Web introduces technological challenges not present in traditional graphics. The shortage of library tools to support Web graphics application development poses a particular problem for real-time 3D Web graphics. The Internet, with its long latency and low bandwidth, can't easily handle large 3D graphics geometry. These restrictions limit the graphics applications we can develop and deliver for the Web.

To address these challenges, we need new graphics standards that let us develop library tools for portable graphics applications. These standards must run on diverse machine architectures and operating systems to cater to as many potential users as possible, and the major computer firms interested in this market must accept the standards.

As 3D Web graphics advance, human modeling and animation will be important not only for the application contents but also as a human–computer interface. Distributed virtual environments form a core part of the Web 3D engine, and research focuses on making DVEs both realistic and lightweight for the Web environment.

Web graphics standards

Graphics standards are evolving to adapt traditional graphics to the Web environment.

2D vector graphics

Although most Web images are bitmap images, vector-based graphics are becoming increasingly important. Users often convert vector images to bitmap images because Web browsers better support the latter, but this negates many advantages of vector graphics. Also, browsers typically implement vector graphics viewers as plug-ins. Without native browser support, vector graphics work poorly on the Web.

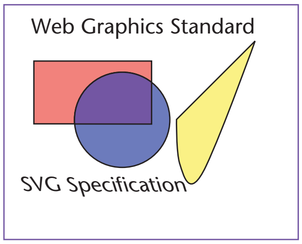
WebCGM. Two markets exist for 2D vector graphics. Industrial technical documentation uses traditional technical documents that contain standard computer graphics metafiles (CGM, an ISO standard for vector graphics; http://www.w3.org/Graphics/WebCGM). Standard CGMs don't suffice, however, for online technical documents. The WebCGM profile addresses this problem by adding constraints to improve interoperability, defining how hyperlinking works, and defining mechanisms for use in HTML.

SVG. Graphic design for advertising, clip art, business presentations, and Web interfaces needs advanced features like animation and filter effects. More importantly, the vector graphics must work well across platforms and with different devices. The scalable vector graphics standard (visit http://www.w3.org/Graphics/SVG)—a World Wide Web Consortium (W3C) recommendation designed to adapt to different output resolutions, color spaces, and available bandwidths—meets these requirements. SVG 1.0 describes 2D vector and mixed vector–raster graphics in XML. SVG is bandwidth efficient, especially with its intrinsic gzip compression support, and resizes easily to fit small Web-surfing devices such as mobile phones and PDAs.

Figure 1 shows SVG code and the resulting image, and Figure 2 shows a sample SVG application.

SVG permits tight integration of graphics and Web documents. Users can embed an SVG file or insert it as a link in any HTML file along with the related data and text, and can style SVG using Cascading Style Sheets (CSS) in the same manner as HTML. Users can also embed scripts in an SVG file to modify an existing element, its properties, and attributes within the SVG document; or create a new element and insert it to the element tree for rendering. For animation, SVG uses the Synchronized Multimedia Integration Language (http://www.w3.org/TR/2001/REC-smil-animation-20010904/). SMIL can synchronize

Published by the IEEE Computer Society



1 **An SVG sample and its rendering result.**

2 **An SVG application demonstration from Adobe. (Image courtesy of Adobe)**

3 **An X3D sample scene from OpenWorlds. (Image courtesy of OpenWorlds)**

graphics primitives with sound, text, and other multimedia elements using simple declarative tags rather than procedural programming code.

XML works well as the underlying language because of its popularity and because it provides structure and semantics. In SVG, graphics and their building blocks can have semantic meaning. This permits better manipulation of many structured graphics such as maps, illustrations, user interfaces, and animations. Developers can also leverage recently developed XML technologies, including Document Object Model (DOM), XSLT (a language for transforming XML documents into other XML documents), and XML library tools such as Active Server Pages (ASP), Java Server Pages (JSP), and Hypertext Preprocessor (PHP). SVG interoperates with all these technologies. For example, developers can create an SVG template for graphics and use server-side tools such as transformations to customize the graphics for end users, or bind data to the SVG template to generate dynamic graphics.

SVG also provides modularization, referred to as SVG 1.1. For a given application scenario, developers can bundle some SVG modules to create an SVG subset or combine them with external modules to extend SVG. The list of SVG modules in such a combination is a *profile*. Its modularization and profiling make SVG a truly scalable solution, and the mobile industry has chosen its mobile profile as the vector graphics format on third-generation mobile phones.

SVG has gained wide acceptance, and various open-source and commercial viewers and authoring tools now exist. Even server-side applications for generating SVG content have become common. We expect SVG to become part of the Web's 2D graphics rendering engine.

3D graphics

Although hardware 3D accelerators are becoming popular and accessible, 2D graphics still dominate today's Web pages. Most Web developers avoid 3D graphics because heavyweight 3D models take a long time to download and render. Many Web sites, particularly business sites, aim to create pages that download fast—Macromedia's Flash animation system, for example, succeeds because it's lightweight.

Nevertheless, 3D graphics offer much business potential, especially as PCs gain high-end 3D graphics rendering capability. Recent 3D Web graphics work has sought ways to transmit 3D content on today's bandwidth-constrained Web. These efforts include the early Virtual Reality Modeling Language, VRML97, and Extensible 3D (http://www.web3d.org/fs_specifications.htm).

X3D. The Web3D Consortium (originally the VRML Consortium) is developing X3D as the successor to VRML, the original ISO standard for Web-based 3D graphics. X3D defines interactive Web- and broadcast-based 3D content. It runs on many hardware devices and applications, and provides a universal exchange format for integrated 3D graphics and multimedia. Figure 3 shows a sample scene built using X3D.

X3D extends VRML with new features, advanced application programmer interfaces, additional data encoding formats, and a component-based architecture that permits a modular approach. Its data encoding formats include XML, VRML97, and a binary format, and its runtime architecture is separate from the data encoding. The XML data encoding gives X3D many of the advantages of SVG. It contains both declarative elements and procedural elements. Using declarative elements, X3D creates a hierarchical object tree as a scene graph, which can contain 3D graphics, 2D graphics, animation, spatialized audio, video, and user-defined objects. Using procedural elements written in scripting language, X3D changes the scene graph dynamically. A rich set of X3D application programming interfaces—specified as a set of language-independent services in the Interface Definition Language (IDL)—supports varied user interaction, navigation, and physical simulation. Third parties can have their own X3D implementations. In fact, we could consider Java3D an implementation of X3D.

MPEG recently adopted the X3D Interactive Profile for the MPEG-4 standard to provide lightweight, interactive 3D graphics. A baseline 3D feature set targets devices with limited processing power.

Java3D. Java3D is a high-level API using an object-oriented scene graph-based 3D graphics model. Programmers can use a Java class hierarchy to create and manipulate 3D geometric objects. These objects reside in a virtual universe, which is then rendered. Java3D can use either DirectX or OpenGL low-level APIs to take advantage of 3D hardware acceleration, giving users platform independence without sacrificing performance. However, this makes Java3D components heavyweight because of the native peer required to actually do the rendering. Java3D also intentionally hides rendering pipeline details from developers, making it unsuitable for some projects.

Future trends

We expect evolving Web graphics standards to share certain characteristics. They should support vector graphics manipulation to help developers create dynamic, data-driven graphics, especially with vector-based graphics and animation gaining popularity. Likewise, with so many XML-based services available on the Web, standards that support XML will prove more suitable for developing Web graphics services.

One fixed feature set won't cover all application areas, so building components and profiling will help developers provide a limited or extended specification. Thus, one standard can provide excellent reusability across different devices and application areas.

Distributed virtual environments

Virtual environments let users walk through, visualize, or interact inside a 3D virtual space. The Web lets multiple users share and interact within a VE, and we refer to such a VE as a distributed virtual environment. DVE developers face the challenge of delivering hefty VE databases to each client machine, however. Two popular approaches to this problem exist: full replication and on-demand transmission.

Full replication

With this approach, each DVE participant downloads a copy of the complete VE database from the server or a peer participant. This system is technically simple to implement, and the runtime data transmitted is usually minimal because each participant already received a copy of the complete VE during initialization. Also, the system can continue to operate even if the network occasionally disconnects.

However, this approach suffers from a long start-up time and poor scalability. A new participant can't interact in the VE until the database is completely downloaded. This download time may be unacceptably long, especially for a large VE. Also, participants must maintain up-to-date object status, but updates for objects that they're not interested in waste processing power and network bandwidth. Finally, portable devices such PDAs and mobile phones are becoming general-purpose machines for most people, but their limited storage space makes it unfeasible to store the complete VE inside them. Hence, the full-replication approach is restricted to clients with ample storage.

On-demand transmission

This approach provides each participant only a small part of the VE based on content selection criteria. For example, in a VE divided into several regions, a participant connects only to one region at a time. Hence, only the objects within the region need to be replicated. This resembles the full-replication approach but on a regional scale.

Content selection might also derive from the *area of interest*,[1] usually defined as the circular visible region surrounding the participant. As the participant moves around inside the VE, the system dynamically downloads objects inside the AOI to the client.[2] This AOI concept may also work with other media, such as audio range.

On-demand transmission delivers only required data to clients, significantly reducing their storage requirements and increasing DVE application flexibility. For example, we can construct a Global Positioning System-based 3D navigation DVE using PDAs as clients. Travelers can download a portion of a 3D map from the server to their PDAs based on their current locations to explore nearby buildings and locate their target destinations. Figure 4 shows an example VE generated by an on-demand transmission system.

Design issues

Although on-demand transmission offers advantages over full replication, it's more complex to implement and presents several design challenges.

Multiple servers. An application expecting many users might need multiple servers to maintain interactivity. However, if the users need to interact with each other, system designers must also consider how to partition the VE to minimize communication overhead.

Amount to prefetch. Because of the Internet's typically long latencies, the DVE system must prefetch potentially visible objects to ensure interactive display. Prefetching too large a VE region wastes network bandwidth, but if the prefetched region is too small, a sudden change of movement or viewing direction could generate display discontinuities.

Motion prediction. Network latencies also make real-time interaction among participants difficult, necessitating frequent roll-backs to resolve inconsistencies. Motion-prediction methods help reduce network latency effects, but most of them don't work well for predicting the viewer's motion inside a DVE. The prediction errors introduced may waste network bandwidth, particularly when used in prefetching.[3]

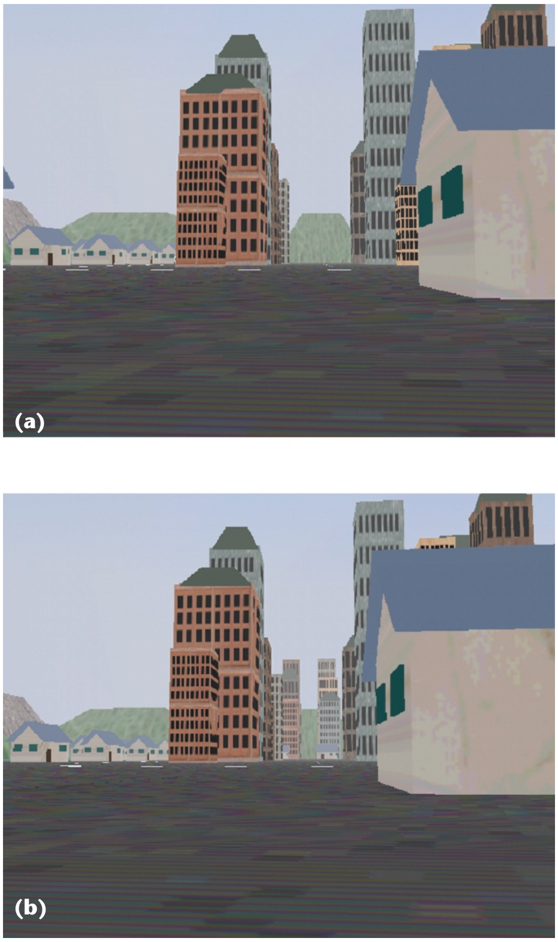
Program complexity. System interactivity requires incorporating complex techniques into the DVE. To work in a Web browser without crashing it too frequently, however, the system must be lightweight and simple. DVE designers must therefore balance simplicity and interactivity.

Research and commercial systems

Current DVE systems include both research prototypes and commercial products. Table 1 shows some of these systems and their properties.

Future research directions

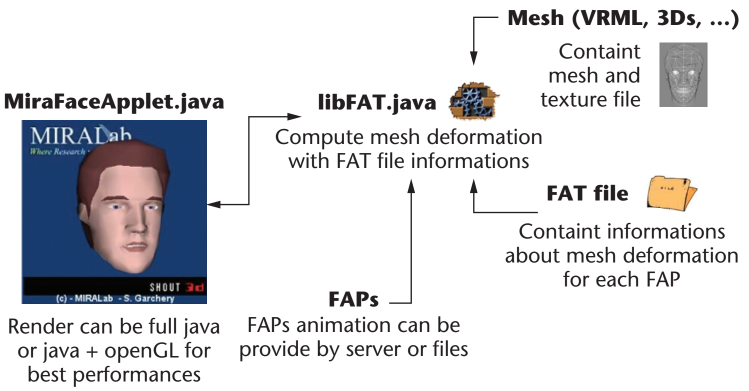
DVE research and development must address several issues. In particular, although researchers have proposed progressive model transmission techniques, existing hardware 3D graphics accelerators don't sup-



4 On-demand transmission of object models based on the viewer's area of interest, (a) before and (b) after transmission.

Table 1. Example DVE systems.

DVE Type	Full Replication	On-Demand Transmission
Research Systems	**SIMNET** (DARPA) First successful DVE implementation provided peer-to-peer, multicasting, LAN only.	**NPSNET** (Naval Postgraduate School; http://movesinstitute.org/~npsnet/index.html) Employs areas of interest (AOI) manager; distributes objects by MBone. Handles VE as a continuous spatial space and divides it into small hexagonal cells. **Massive** (Nottingham University, UK; http://www.crg.cs.nott.ac.uk/research/systems/) Based on spatial model of interaction. Generalizes AOI concept to other media, such as audio. **DIVE** (Swedish Institute of Computer Science; http://www.sics.se/dive/) Provides peer-to-peer, unicasting, LAN only. VE is divided into regions, each fully replicated when needed. **CyberWalk** (City University of Hong Kong; http://www.cs.cityu.edu.hk/~rynson/projects/dvr/) Object visibility based on object/viewer scopes, progressive object transmission, and prefetching.
Commercial Products	**Online Games** Complete game environment stored in CD-ROMs. Examples include *Quake* (idSoftware; http://www.idsoftware.com/) and *Starcraft* (Blizzard; http://www.blizzard.com/) **VRML** (http://www.vrml.org/), originally developed by SGI	**Distributed Virtual Environments** Two approaches: divide VE into regions and present each when user enters it (requires extra download time if user migrates to another region) or determine participant's AOI to handle user interaction and distribute dynamic information. Examples include Blaxxun (Blaxxun interactive; http://www.blaxxun.com/) and Community Place (Sony; http://www.sony.co.jp/en/Products/CommunityPlace/)



5 The Web-based facial animation applet.

port dynamic geometry model rendering—they assume each geometry model's vertex information to be static. Dynamic geometry models, on the other hand, require dynamic insertion or deletion of vertices from the model. Because the accelerators don't support operations such as edge split and edge collapse, the CPU must perform these operations and transfer the complete model to the accelerator every time such an operation is performed. This significantly reduces the performance gains these dynamic geometry models offer. To take advantage of dynamic geometry models, future graphics accelerators should be programmable to permit such simple operations.

Virtual humans on the Web

Lifelike human face and body animation not only improves DVEs' realism but also provides a more natural Web interface in general. However, the Web environment demands reliable, real-time performance.

Virtual faces

The past 20 years of facial modeling and animation research have focused on improving facial animation systems' believability, realism, flexibility, and applicability. Recently, the focus has shifted toward real-time techniques that maintain the animation's quality and beauty on the Web. Integrated into Web services applications, virtual human technologies can prove useful as navigation aids, salespersons, hosts, and presenters.

Web requirements. Integrating interactive animated characters on Web pages requires the following system attributes:

- *Easy installation*. In most cases, we need to install a plug-in to display a talking head. Vendors such as Cult3D, Pulse3D, Java3D, and Shout3D offer 3D plug-in APIs that developers can use to create and integrate 3D characters in Web pages. However, the ideal solution is a plugless approach.
- *Real-time interactivity*. Applications such as navigator help, a sales assistant, or an online virtual tutor require interactivity. These replay predefined animation but also let users talk to virtual characters in natural language. Ensuring speed and ease of use requires integrating Web-based artificial intelligence, speech, and graphics technologies.
- *Visual quality*. To achieve interactive speeds, Web pages often use low-resolution models and images, which hampers the application's visual quality. Designers should therefore balance visual quality with data size.
- *Fast download*. High initial download time for Web-based animated characters is acceptable, but real-time interaction should demand as little data transfer as possible.
- *Easy Web integration*. To integrate talking characters into Web pages, developers need easy-to-use tools that give quick results.

Case study. To create a Web-based virtual talking character, Garchery et al.[4] used MPEG-4 Facial Animation Table (FAT) technology and the Shout3D rendering engine for facial animation. Shout3D provides a Java API, so the visualization is plugless and the user doesn't have to install any special software in a Java-enabled browser. MPEG-4 FAT animates high-quality models with minimum real-time computation and therefore enables interactive speeds.

In developing the Web face applet, the most important and complex work is the FAT construction. The FAT specifies the morph targets for a given model's MPEG-4 facial animation parameters. The FAPs are optimum for defining reasonable-quality facial animations in most cases and easily compressed and transmitted over the network with appropriate encoding.

We've developed tools to automatically build and export FATs. For each low-level FAP, an applet applies various intensities of FAPs to the facial mesh and then compares the deformations. An animator can also specify how a character speaks (by defining the visemes) and uses facial expressions. Once the animator has designed the FAT set, the deformation set is compiled into the FAT information, which the Web-based FAT animation engine uses. After constructing the FAT, the animator can apply any FAP animation to the synthetic face and synchronize it with audio in the Web browser.

Figure 5 shows components of the Web-based facial animation applet.

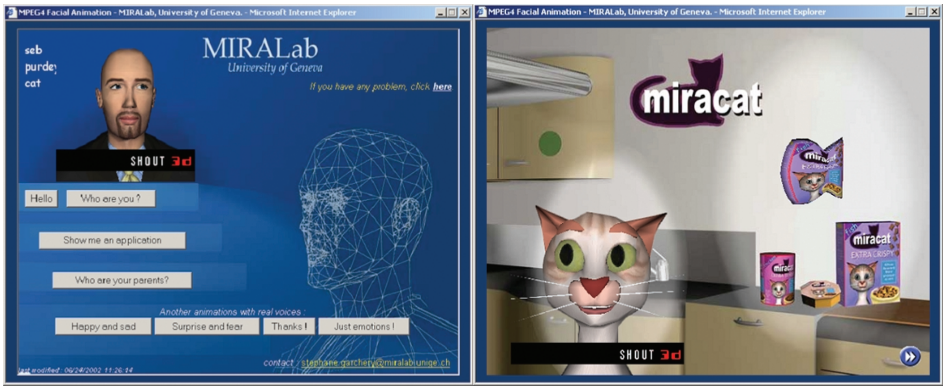
Sample applications. We used the Web face applet to play prerecording FAP stream animation and audio. Figure 6 shows sample files corresponding to different states or events. At left, a virtual presenter talks about FAT technology. At right we see a cartoon-like model that sells cat food.

You'll find these demonstrations at http://www.miralab.unige.ch, under "Research topic," subcategory "Facial animation."

Future directions. As 3D Web technology evolves, Web developers are exploring new facial animation systems. The most challenging aspect remains real-time intelligent interaction that not only engages users but also helps them with Web navigation and online shopping. Several Web sites deploy animated characters that offer text or speech interaction (see Table 2).

Virtual body

The most common virtual human implementations consist of animated faces ("talking heads") that use some variation of text-to-speech and dialog manager software to drive an "intelligent" conversation with users or present information verbally. Although they offer some interaction, their character expressions are limited to facial gestures. Limitations remain, however, in speed, visual appearance, intelligent dialog management, and interactivity.



6 **Facial animation applications on the Web.**

In particular, these talking heads can't show full body expressions and gestures. Extending the talking-head model to a full-body virtual human will let developers complement text-to-speech voice synthesis, dialog management, and facial animation with body gestures to enhance expressions and humanize virtual characters with nonverbal communication.

To enhance human–computer interaction using nonverbal communication (body gestures), we developed a multiple-platform, real-time tool, called the MPEG-4 Body Player, which produces realistic full-body animation of 3D virtual humans in Web applications. Adding body gestures to virtual human expressions gives users the impression that they're interacting with a real person inside the computer. We especially sought the capability to synthesize full-body animation instead of limiting it to face animation. Other important goals included ease of integration in a Web-based application and user-friendly installation.

General system requirements. Such an interface has several system requirements:

- *System characteristics*. The new player must execute across multiple platforms, and users shouldn't need plug-ins or additional downloads.
- *Expected performance*. The player should render 3D virtual humans at 10 frames per second or more for realistic animation.
- *Data representation and communication*. The player requires that the system send the virtual human's 3D geometric description and a set of animation parameters over the network. The system must continuously update the animation parameters to keep the virtual human "alive."

General technology. The 3D geometric description should include information to calculate anatomical deformations in real time. To enable the system to obtain the required high-quality results with a low-bitrate data transfer, developers must use an efficient data representation (the 3D content). VRML and MPEG-4 represent the two main standard representations of 3D content for Web applications. MPEG-4 adopts H-Anim,

Table 2. Web tools for animated characters.

Web Site	Interaction	Plug-in Requirement	3Dn	Application
http://www.pulse3d.com/	User types text and the character speaks it	Pulse3D	Yes	Helps Web sites integrate 3D talking characters
http://www.lifefx.com/	User types text and the character speaks it	Dedicated	Image-based character with limited head rotation	Send emails with animated characters
http://www.KurzweilAI.net/	User types question and the virtual assistant answers	Lifeex.com	Limited head rotation; user can't control 3D head's orientation	Web-based navigator, online assistant
http://www.mendel3d.com/	No interaction; replay of prerecorded animation	Mendel3D	Yes	Web news reader, weather forecast
http://www.haptek.com/	No interaction; replay of prerecorded animation	Dedicated	Yes	Web navigator, development kit

7 **The MPEG-4 face and body animation object.**

8 **Modeling the six basic human emotions: (a) joy, (b) sadness, (c) anger, (d) surprise, (e) disgust, and (f) fear.**

a VRML-based standard for virtual human representation, and provides an efficient way to animate virtual human bodies.[5]

We defined the body player as a pure Java applet that uses MPEG-4-compliant data streams to represent 3D content and the Shout3D Java engine (http://www.eyematic.com/) to render 3D scenes. This system can execute on multiple operating platforms without plug-ins or additional downloads and provides a minimum performance of 10 frames per second, depending on available computing power. The player uses the MPEG-4 face and body animation (FBA) object to describe and animate the virtual human geometry.

MPEG-4 defines two sets of parameters. One set specifies the FBA model geometry as face definition parameters (FDPs) and body deformation parameters (BDPs). These let the decoder specify shape and texture for an FBA model. The other set defines the animation of the face and body: FAPs and BAPs (face/body animation parameters). Developers apply BAPs—a set of rotation angles—to body parts to modify posture. In our player we consider the face as part of the body and don't use specific parameters for facial definition and animation. The tool's design, however, permits integration of a face animation module. Figure 7 shows a schematic view of the FBA object.

The MPEG-4 BDPs used to define the virtual human body relate directly to the VRML/Web3D H-Anim 1.1 specification. This standard divides the human body into segments (such as the forearm, hand, and foot) connected by joints (such as the elbow, wrist, and ankle). An H-Anim file contains a set of joint nodes hierarchically arranged as Figure 7 shows. Each joint node may contain other joint nodes and a segment node that describes the body part associated with that joint. Each segment is a normal VRML transform node describing the body part's 3D shape.[6]

MPEG-4 BAPs enable body movement synthesis. To animate a virtual human, the MPEG-4 Body Player accesses the H-Anim joints and alters their angles to match those defined in a BAP stream. A BAP file can contain up to 296 parameters describing the skeleton's topology. Three different angle parameters describe most joints' joint orientation: yaw, roll, and pitch. Separate segments defined by a mesh of polygons make up the BDP model. Modifying the joints' rotation angles with BAPs rotates the children segments from their original positions, creating a discontinuous surface over the rotated joint. This leads to visually unnatural animations. The set of vertices that form the segment's polygon mesh must therefore be modified to keep an anatomically realistic, continuous surface over the modified joint. MPEG-4 defines preset deformation tables containing the coordinates of the segment vertices to be deformed for each joint's key postures. The deformations of the segments on intermediate postures (not contained in the preset tables) are calculated using linear interpolation.

The player has two main input files: the MPEG-4 streams that define the virtual human geometry (BDP file) and include the anatomic deformations (BDT) information, and the body animation parameters (BAP file) containing information to alter the virtual human joints and display the animation. The system core consists of a Java package (set of classes) that includes the MPEG-4 data management and animation classes (BAPReader and HAnimBAP) and the graphics rendering classes, which are directly related to the Shout3D engine (BDPLoaderPanel and BDPLoaderApplet). The last class, BDPLoaderApplet, is the external component that must be embedded in the Web-based application for the player's use.

Case study. Researchers integrated the MPEG-4 body player in a demonstration application developed in the framework of the IST-Interface (Information Societies Technology Interface) project.[7] Among its goals, the project focuses on emotion synthesis, and this Web-based application synthesizes emotional gestures corresponding to the six basic universal emotions—joy, sadness, anger, surprise, disgust, and fear (Figure 8). The demonstration shows that intensity levels can distinguish different peoples' emotional behavior—in this example, northern, central, and southern Europeans.

Three different virtual humans perform the emotional gestures for each region.

Future directions. To realistically animate avatars in virtual environments requires synchronizing speech, facial expressions, and body gestures. The Swiss Federal Institute of Technology Virtual Reality Lab recently proposed a new high-level animation language, the Avatar Markup Language,[8] to describe avatar animation. Based on XML, AML encapsulates text-to-speech, facial animation, and body animation in a unified manner with appropriate synchronization. Now we need an agent-based language that lets developers define higher behaviors using a similar standards-based approach. Such a development will lead to powerful Web-based applications.

Topological Web graphics

Web graphics remain limited by several computer graphics technology drawbacks. For example, polygon explosions—ever-increasing amounts of graphics data originating from polygonal shape representation—hinder Web graphic interactions and communications. To improve graphics performance, we must reformulate the current polygon-based computer graphics on more scientific grounds such as the homotopy model.[9]

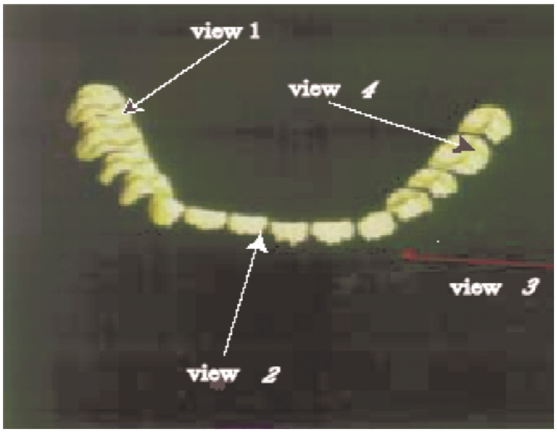
Also, users can't easily browse Web graphics because traditional projective-geometry-based graphics don't provide enough visual information. We need multiple-view graphics in addition to the traditional single-view projected images on the screen. In fact, current Web graphics drawbacks stem from a more fundamental defect: the lack of basic models to define Web graphics.

To render complex, nonlinear cyberworlds, topological Web graphics must present their diverse topological properties on graphics screens. Only mathematically verifiable tools prove valid for complex applications; hence, topological Web graphics apply certain nonlinear mathematics mostly in algebraic topology. Manifolds, homotopy, curvature, and cellular spatial structures provide versatile topological Web graphics and the tools. Computer graphics has neglected homotopy modeling, but we see it as key to modeling cyberworld changes and displaying them in Web graphics.

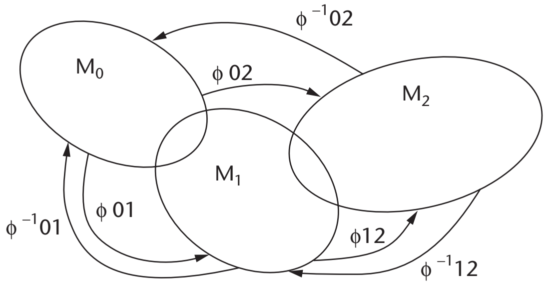
Multiple-view Web graphics

Current computer graphics rely on projective geometry, which has served as the framework for human view representation for thousands of years. Hence, a displayed image usually results from the projection of objects as seen from a single viewpoint.

Web graphics require techniques to present cyberworlds as seen from multiple viewpoints. For example, Web-based manufacturing uses tools to display automobile assembly as concurrent processes that compose automobile parts gathered from part suppliers' Web sites and to view the assembly from multiple directions simultaneously. Web manufacturing depends on rapid prototyping, which requires tools that permit viewing a prototype from various directions to confirm the design before testing.



9 Dental inspection as seen from multiple viewpoints, illustrating a manifold consisting of three charts displayed as an integrated image.



10 A multiple-view picture as a manifold of three charts.

Dental care also requires rapid prototyping. A dentist views teeth from different directions to see their general condition and conceptualize them, as Figure 9 shows.[10]

Figure 10 shows an implementation of Figure 9 as a manifold consisting of three charts—M_0, M_1, and M_2—and local coordinate transformation functions Φ_{01}, Φ^{-1}_{01}, Φ_{12}, Φ^{-1}_{12}, Φ_{02}, and Φ^{-1}_{02}. We normalized each overlap of two charts by the partition of unity, implemented by a blending function.[11]

Differential topological Web graphics

Handling graphics on the Web requires identifying necessary graphical objects quickly for real-time interactions. We need to characterize Web graphics data using unique information in small amounts. Web graphics indexing therefore becomes particularly important.

The Morse lemma and Reeb graph represent powerful tools to abstract 3D shapes' characteristics for use as indices. The Morse lemma states that any given shape is uniquely characterized by characteristic points, namely peaks, pits, and passes if they're nondegenerate. If they're degenerate, we lift peaks and lower pits by infinitely small amounts to turn them nondegenerate and then render as usual. Thus, we can establish Morse lemma-based graphics (or *Morse graphics*) as a Web graphics standard.

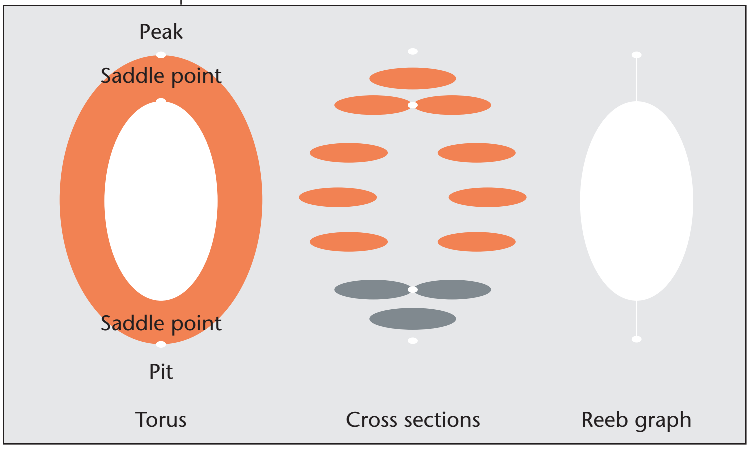
The Reeb graph charts shapes' critical points and can use a small amount of information as the indices to uniquely identify shapes. Using the torus as an example, Figure 11 shows how we characterize it by the critical points and use the Reeb graph to index it. Figure 12 presents a terrain, the characteristic points, and the Reeb graph indexing it.

Few have applied Reeb graphs to graphics, however, because of the height function dependence. We recently created height-function-independent Reeb graphs using geodesics distance and have applied them successfully to many cases.[12]

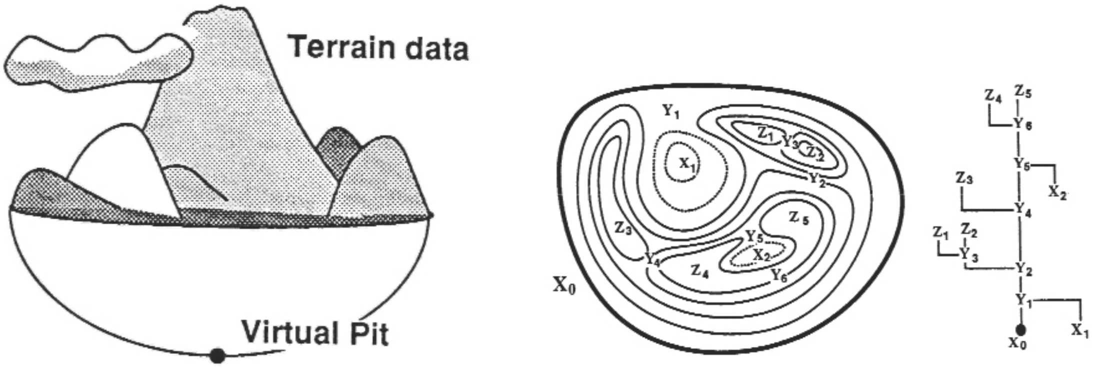
Topological graphics is a new and versatile area, and its foundations and applications both need extensive study. It clearly saves polygon explosions that are unavoidable in current computer graphics. For rapidly expanding Web graphics running over the Internet, it is crucial. We see promising directions for future studies in algebraic topological graphics for generalizing topological graphics, including cellular structure-based and quotient space-based computer graphics. ■

Acknowledgments

Support for this work came from the Swiss Federal Office for Education and Science in the framework of the European project IST-Interface and from Strategic Research Grants from City University of Hong Kong (project numbers 7001285 and 7001391).



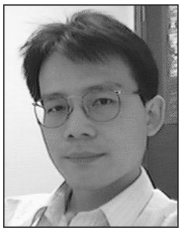
11 A torus, the critical points, and the Reeb graph.



12 A terrain, the characteristic points, and the Reeb graph.

References

1. J. Falby et al., "NPSNET: Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation," *Computers & Graphics*, vol. 17, no. 1, 1993, pp. 65-69.
2. J. Chim et al., "CyberWalk: A Web-Based Distributed Virtual Walkthrough Environment," *IEEE Trans. on Multimedia* (to appear).
3. A. Chan, R.W.H. Lau, and B. Ng, "A Hybrid Motion Prediction Method for Caching and Prefetching in Distributed Virtual Environments," *Proc. ACM VRST*, ACM Press, 2001, pp. 135-142.
4. S. Garchery and N. Magenant-Thalmann, "Designing MPEG-4 Facial Animation Tables for Web Applications," *Proc. Multimedia Modeling*, 2001, pp. 39-59.
5. C. Babski and D. Thalmann, "Real-Time Animation and Motion Capture in Web Human Director," *Proc. Web3D*, 2000, pp. 139-145.
6. Web3D Working Group on Humanoid Animation, *Specification for a Standard Humanoid*, version 1.1, Aug. 1999, http://h-anim.org/Specifications/H-Anim1.1.
7. Information Societies Technology Programme, "Project INTERFACE: Multimodal Analysis/Synthesis System for Human Interaction to Virtual and Augmented Environments," Annex 1 - Description of Work, Oct. 1999, http://www.ist-interface.org/.
8. S. Kshirsagar et al., "Avatar Markup Language," *Proc. Eurographics Workshop on Virtual Environments*, 2002, pp. 169-177.
9. T. Kunii, "Graphics with Shape Property Inheritance," *Proc. Pacific Graphics*, 1998, pp. 2-6.
10. K. Myszkowski et al., "Modeling of Human Jaw Motion in Sliding Contact," *J. Visualization and Computer Animation*, vol. 8, no. 3, July 1997, pp. 147-163.
11. A.T. Fomenko and T.L. Kunii, *Topological Modeling for Visualization*, Springer-Verlag, 1997.
12. M. Hilaga et al., "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," *Proc. ACM SIGGRAPH*, ACM Press, 2001, pp. 203-212.

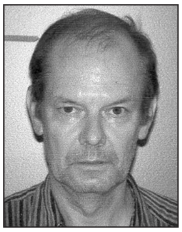


Rynson Lau *is an associate professor at the City University of Hong Kong. His research interests include computer graphics, virtual reality, and multimedia systems. He received a first-class honors degree in computer systems engineering from the University of Kent at Canterbury, England, and a PhD in computer graphics from the University of Cambridge, England. Visit his Web site at http://www.cs.cityu.edu.hk/~rynson/*

Tosiyasu L. Kunii *is a professor at Hosei University, Honorary Visiting Professor at the University of Bradford, and professor emeritus at the University of Tokyo. In 1991, he was elected an IEEE fellow for his contribution to visual computers and visual computation. He is also Golden Core Member of IEEE Computer Society with Charter Membership, and a recipient of 1998 Taylor Booth Award of IEEE Computer Society for his contribution to computer science education.*

Baining Guo *is research manager of the Internet Graphics group at Microsoft Research Asia (formerly Microsoft Research China). His main research interests are modeling and rendering. His current projects include networked games, appearance modeling, natural phenomena, texture synthesis, and image-based rendering. He received his PhD and MS from Cornell University and his BS from Beijing University.*

Nadia Magnenat-Thalmann *has been a pioneer of virtual human research for 20 years. She obtained her PhD in quantum physics at the University of Geneva and was a professor at the University of Montreal. In l989, she founded MiraLab in Geneva. She is editor in chief of* The Visual Computer.



Daniel Thalmann *is professor and director of the VR lab at the Swiss Federal Institute of Technology. His research interests include virtual humans, computer animation, networked virtual environments, and artificial life. He received his diploma in nuclear physics and a PhD in computer science from the University of Geneva. Thalmann is coeditor in chief of the* Journal of Visualization and Computer Animation.

Frederick Li *is research manager of the Web-based 3D collaborative engine project funded by the Hong Kong Government ITF Fund, Pacific Century CyberWorks (PCCW), and Sun Microsystems. His research interests include surface modeling, virtual reality, and distributed virtual environments. He received BArts and Mphil degrees from the Hong Kong Polytechnic University and a PhD from City University of Hong Kong.*

Bo Zhang *is an associate researcher at Microsoft Research, Asia. His research interests include vector graphics, game networks, and game artificial intelligence. He received a BS and PhD in computer science from the University of Science and Technology of China.*

Sumedha Kshirsagar *is a research assistant and PhD student at MiraLab, University of Geneva. Her research interests include real-time facial and expressive speech animation, emotion and personality modeling, and embodied conversational agents. She obtained a BE in instrumentation from Pune University, India, and a Master of Technology in electrical engineering from Indian Institute of Technology, Bombay.*

Mario Gutierrez *is a research assistant and PhD student at the VR lab of the Swiss Federal Institute of Technology. His research focuses on control systems for animation of autonomous virtual humans, MPEG-4 body animation, Internet and mobile applications, and computer graphics in general. He received an MS in computer sciences from Monterrey Institute of Technology, Toluca Campus, Mexico.*

Readers may contact Rynson Lau at the Dept. of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, email rynson@cs.cityu.edu.hk.