Introduction to the Special Issue

3D for the Web

Michael Macedonia

When the Web was young and we all owned 2,400-bps modems, Stefan Noll and I wrote this in IEEE CG&A:

In the 1995 report, “Virtual Reality: Scientific and Technological Challenges,” the National Research Council surveyed key challenges in VR technology and research. Years of effort on distributed interactive 3D graphics applications have clearly identified networking issues as the critical bottlenecks preventing the creation of large-scale VEs [virtual environments] as ubiquitously as home-page space on the Web. Those years have also produced important component technologies such as VRML [Virtual Reality Modeling Language] and Java. We believe the time is right to combine the network pieces with the graphics pieces.¹

What a long, strange trip this has been. VRML mutated into X3D (Extensible 3D), which in turn might become part of the overall set of HTML5 standards. JavaScript, as opposed to Java, became more robust and the common language for Web development. And new ideas for natively rendering 3D graphics in a browser, such as WebGL (Web Graphics Library), evolved to support higher-level declarative languages.

More important, the Web and interactive 3D graphics are both part of today’s information environment and often are combined in such diverse applications as online games (for example, Farmville), terrain visualization (for example, Google Earth), and distributed CAD (for example, AutoCAD 360). The Web, made ubiquitous on mobile devices, provides the basis for real-time, worldwide information distribution and has established itself as a universal application platform. It has also enabled the whole notion of user-generated content. 3D graphics and computing are now integral to all modern processors. The GPU’s wide introduction over the past 10 years has enabled high performance, even on mobile platforms, that drives big markets for 3D movies, 3D displays, and, of course, games and entertainment.

Hurdles to Overcome

However, despite the utility of interactive 3D graphics on the Web (often called Web3D), direct support for it is still minimal. There are many reasons for this. The commercial demand for 3D graphics was superseded by consumer demand for streaming services for more traditional media such as video and music. Moreover, 3D content is expensive to produce. Intellectual-property issues and lack of a 3D-digital-media rights model have also made it difficult to have open standards for Web3D. The different rendering approaches by the primary browser makers and standards bodies have pushed the Web3D market into a “wait-and-see” mode. This Tower of Babel has also led many users to consider proprietary but somewhat promising technologies such as cloud-based rendering.

Also, just as distributed 3D was gaining traction on PCs, the mobile-Internet revolution occurred. This delayed the introduction of many technologies that were hungry for bandwidth and computer performance. Bandwidth on mobile devices has been restrictive; only recently has it been made usable for 3D content, with the introduction of 4G or LTE (long-term evolution). The Internet also continues to suffer from major technical challenges such as “buffer bloat” (partly exacerbated by streaming-media applications), which introduces intolerable latency for interactive 3D applications. Finally, a key problem is mobile devices’ mundane lack of battery life. Developers currently face a trade-off between communications and graphics-rendering perfor-
mance on tablet computers—which now outsell once-dominant PCs.

Wait-and-see also has led to the lack of any successful commercial Web3D applications. However, education might provide an opening as the initial large customer for these applications, given that the government sector demands open standards and interoperability. Moreover, education opens up the possibility for the wide application of user-developed content that Web3D offers.

In This Issue
We received excellent submissions from top European research centers; their articles reflect the ferment in ideas for developing the 3D Web and overcoming past challenges. Two articles exploit GPU ubiquity and performance; the other applies current Web3D languages to a real-world education challenge.

In “Fast, Progressive Loading of Binary-Encoded Declarative-3D Web Content,” Max Limper and his colleagues deal with Web3D performance. They exploit the fact that browsers over the past 20 years have been optimized to natively receive, decompress, and render 2D images. Employing standard formats for compressed binary images, such as PNG, Limper’s team at Fraunhofer IGD uses the images as containers for meshes to eliminate CPU overhead. GPUs provide an efficient, ubiquitous, and low-power mechanism for decompression, transformation, and progressive 3D rendering of image data. This approach results in reduced bandwidth requirements when implemented using current browser capabilities. Limper and his colleagues describe how this happens in the context of using X3D and WebGL. As they point out, using binary mesh containers in a WebGL application is straightforward and leads to higher performance and capabilities such as progressive binary geometry.

The next article addresses both graphics performance and developer utility. In “XML3D and Xflow: Combining Declarative 3D for the Web with Generic Data Flows,” Felix Klein and his colleagues suggest replacing specialized content nodes found in X3D with generic data structures that optimize GPU buffers to minimize conversion overhead. They aim “to provide intuitive, easy-to-use technology that’s flexible and efficient enough for modern 3D graphics and all the accompanying high-performance processing tasks.” The authors developed Xflow, a declarative language for dataflow processing that’s appropriate for GPUs, to add flexible, high-performance processing capabilities to XML3D.

Finally, in “The LiverAnatomyExplorer: A WebGL-Based Surgical Teaching Tool,” Steven Birr and his colleagues apply powerful Web3D technologies to a practical problem—learning complex anatomical structures. Their implementation is based on HTML5, WebGL, and X3DOM, which eliminates the need for plug-ins used in similar tools. This in-depth article describes these languages’ utility for the authors’ learning tool, which they intend to be an exemplar for other developers of open-standards-based medical-education systems. They also cite their tool’s advantages for security in medical environments because it uses no plug-ins.

Despite the utility of interactive 3D graphics on the Web (often called Web3D), direct support for it is still minimal.

We hope you enjoy these articles. We’ve seen 20 years of dramatic change in computer graphics because of the Internet and the growth of the Web. These articles reflect the struggle between innovation, market competition, and the need for common standards and technology reference points. The 3D Web’s evolution isn’t over; it’s still young, and the technologies aren’t yet mature. The tablet computer became a disruptive force in computing when the technology harmonics of wireless bandwidth, low-power computing, and touch interfaces converged. Web3D might be in a similar position with modern browsers’ exploitation of the GPU, the emergence of a consensus on WebGL, and the development of novel applications such as the LiverAnatomyExplorer.

Reference

Michael Macedonia is the former chief scientist for SAIC’s simulation and modeling business operation. Previously, he was the general manager for Forterra, a 3D virtual-world startup. He is also the former chief technology officer for the US Army’s simulation technology acquisition organization and was an editor of IEEE CG&A’s Projects in VR department. Macedonia received a PhD in computer science from the US Naval Postgraduate School. Contact him at macedonia@computer.org.