Protecting 3D Graphics Content

BY DAVID KOLLER AND MARC LEVOY

To prevent the theft of 3D models, these methods defend the high-resolution geometric detail of their physical shape, while still allowing interactive display and manipulation.

The digital rights management problem of protecting data from theft and misuse has been addressed for many information types, including software code, digital images, and audio files. Few technological solutions are designed specifically to protect interactive 3D graphics content.

Demand for ways to protect 3D graphical models is significant and growing. Contemporary 3D digitization technologies allow the efficient creation of accurate 3D models of many physical objects. For example, our Stanford Digital Michelangelo Project [3] has developed a high-resolution digital

Computer rendering of Michelangelo's *David* made from a laser-scanned 3D model of the statue containing eight million polygons, each 2.0 mm in size. The marble veining and reflectance are artificial. (Stanford Digital Michelangelo Project, rendering by Henrik Wann Jensen)

archive of 10 of Michelangelo's large statues, including the *David* (see the sidebar "Generating a Replica of Michelangelo's *David*"). These statues represent the artistic patrimony of Italy's cultural institutions, and our contract with the Italian authorities permits distribution of the 3D models only to established scholars for noncommercial use. Though everyone involved would like the models to be available for any constructive purpose, the digital 3D model of the *David* would quickly be pirated if it were distributed without protection; simulated marble replicas would be manufactured outside the provisions of the parties authorizing creation of the model.

Digital archives of archaeological artifacts are another example of cultural heritage 3D models that could require piracy protection. Curators of such artifact collections increasingly turn to 3D digitization as a way to preserve and widen scholarly use of their holdings, but they often want strict control over the manner of that use of the 3D data and to guard against theft. An example of such a collection is our Stanford Digital Forma Urbis Project (formaurbis.stanford.edu) we've undertaken with Italian archaeological officials to digitize more than a thousand marble fragments of an ancient Roman map and make them publically available through a Web-based database—provided the 3D models have adequate protection.

Other application areas (such as entertainment and online commerce) could also require protection for 3D graphics content. Valuable 3D animated character models developed for use in motion pictures and 3D body scans of high-profile actors may be repurposed for widespread use in video games and promotional materials. Content developers might be reluctant to distribute the 3D models in interactive applications without control over piracy and reuse. A number of Internet application developers have reported that their clients are unwilling to pursue online 3D graphics projects due to the inability to prevent theft of 3D content.

Prior technical research in intellectual property protection for 3D data has concentrated on 3D digital watermarking techniques. These steganographic approaches have sought to embed hidden information into 3D graphical models, with varying degrees of robustness to attacks aimed at disabling the watermarks by altering 3D shape or data representation. Many of the most successful 3D watermarking schemes are based on spread-spectrum frequency domain transformations, embedding watermarks at multiple scales by introducing controlled perturbations into the coordinates of the 3D model vertices [4]. Complementary technologies search collections of 3D models, examining them for the presence of digital watermarks in an effort to detect piracy.

For the digital representations of valuable 3D objects (such as cultural heritage artifacts), it is not sufficient to detect piracy after the fact; piracy must be prevented. The computing industry has experimented with a number of techniques for preventing unauthorized use of digital data, including physical dongles, software access keys, node-locked licensing schemes, copy-prevention software, obfuscation, and encryption with embedded keys. Most are either broken or bypassed by determined attackers, causing undue inconvenience and expense for nonmalicious users. High-profile data and software are particularly susceptible to attackers.

Fortunately, 3D graphics data differs from most other forms of digital media in that the presentation format—2D images—is fundamentally different from the underlying representation—3D geometry. 3D graphics data is usually displayed as a projection onto a 2D display device, resulting in a large information loss for single views. This property supports an optimistic view that protected 3D graphics systems can still be useful to users, without making the 3D data as vulnerable to piracy as other types of digital content.

Here, we address the problem of preventing the theft of 3D models, while still allowing for their interactive display and manipulation. Our goal is to provide a solution for maintainers of large collections of high-resolution static 3D models (such as the cultural heritage artifacts we are digitizing). The methods we are developing aim to protect both the physical shape of the 3D models and their particular geometric representation (such as 3D mesh vertex coordinates, surface normals, and connectivity information). We accept that the coarse shape of visible objects is easily reproduced regardless of protection efforts, so we concentrate on defending the high-resolution geometric detail of 3D models. This detailed geometry is usually the most expensive to model or measure (perhaps requiring special access and advanced 3D digitizing technology) and is often the most valuable in exhibiting fidelity to the original object.

Protection Techniques

Figure 1 outlines an abstraction of the 3D graphics pipeline, identifying some of the methods an attacker might use to attempt to recover 3D geometry data in a computer graphics system. To counter

Generating a Replica of Michelangelo's *David*

Demonstrating recent improvements in digitizing the shape of physical objects, a team of 30 faculty, staff, and students from Stanford University and the University of Washington spent the 1998–99 academic year digitizing the sculptures and architecture of Michelangelo, including the well-known sculpture of *David*. Our main scanning device was a laser triangulation rangefinder mounted on a motorized gantry (image a in the figure), using it to digitize the marble statues with a spatial resolution of 0.25mm, dense enough to capture Michelangelo's chisel marks. After scanning, our range data-processing pipeline aligned the scans taken from different gantry positions, combined them into a unified surface mesh, and automatically filled any small holes that could not be seen by the scanner [2]. Scanning the *David* took three weeks, with 400 individually aimed scans resulting in two billion polygons of data. Reduced-resolution versions of the resulting 3D model have been used to make computer renderings (b) and for a variety of scientific and scholarly studies.

A simplified 1.25-million polygon model of the *David* was the basis for replicas we have manufactured in collaboration with Gentle Giant Studios (www.gentlegiantstudios. com/). We used a thermojet wax printer to make a master (c), following with a latex molding and casting procedure. The final replicas (d), standing 15 inches tall, are made of plastic resin, though other materials (such as marble dust in a binder) can be used.

Though the economic value of digital representations of 2D artwork is uncertain due to the proliferation of photographic replicas (many unlicensed), the situation for 3D artwork may be different. Until recently, few famous sculptures had been digitized, and of those that had been scanned, even fewer have been used to manufacture replicas for retail sale. Although replicas of famous statues like the *David* abound, they are usually based on handmade models and are of relatively poor fidelity (e). Thus, the potential economic value of digital representations of 3D artifacts is great, as evidenced by the vigorous market for replicas of statuary conducted through mail-order catalogs.

Our replica is not for sale to the public, though eventually it will be. Meanwhile, our 3D models can be studied using the ScanView protected rendering system described here. The particular model of *David* embedded in ScanView is the same one used to generate the replica shown in the figure, with the exception of some small, inconspicuous modifications we made to each model, effectively watermarking them to identify illegal copies. C

Producing a replica of Michelangelo's *David*: (a) laser scanner positioned in front of the statue in the Galleria dell'Accademia in Florence; (b) computer rendering of 3D model; (c) replica master under construction (Gentle Giant Studios); (d) replica produced from scanned data; and (e) inexpensive replica ($100) purchased from a street vendor in Florence.

these attacks, we have considered several possible approaches for sharing and rendering protected 3D graphics. One involves bypassing the graphics processing unit (GPU) driver and hardware, and using software-only graphics rendering for at least a portion of the data. Software rendering keeps control of the rendering process in the hands of the viewing application programmer, allowing for specialized data encryption or obfuscation techniques to be used to protect the data in the early stages of the pipeline, trading off display performance. Another involves the introduction by the viewing application of subtle deformations in the geometry of the model before passing the 3D vertex data to the graphics driver; attackers would have difficulty reconstructing the full 3D model due to the distortions.

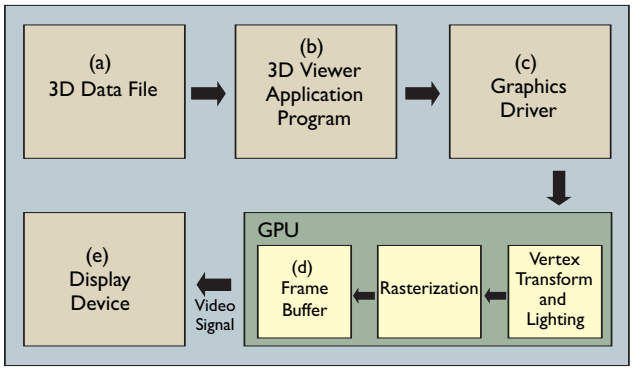

Figure 1. Abstracted graphics pipeline showing possible attack locations (a–e) for recovering 3D graphics data from an interactive viewing program: (a) 3D model file reverse engineering. Users with full access to 3D model data files can reverse engineer obfuscated or encrypted formats; (b) 3D viewer application tampering. Hackers use program tracing, memory dumping, and other techniques to obtain access to data being used by application programs; (c) Graphics driver tampering. 3D data passes through graphics driver software on its way to the graphics hardware where the drivers are vulnerable to tampering or replacement by attackers trying to capture streams of 3D data; (d) Reconstruction from the frame buffer. Sophisticated attackers can access rendered images from the graphics memory, using 3D computer vision techniques to reconstruct the original model; and (e) Reconstruction from the final image display. Irregardless of any system protections in the pipeline, the final video images output from a graphics system are vulnerable to capture and reconstruction.

The drawback of such protection techniques is that they all eventually rely on "security through obfuscation," which is unsound from a computer-security point of view. Any 3D graphics-protection technique that makes the actual 3D data available to potential attackers in software can be broken [5], as any attacker with enough time and resources will be able to reverse engineer the protections on the data. It is possible that future "trusted computing" platforms for general-purpose computers will make software tampering difficult or impossible, but few such systems are deployed today.

Other approaches to protecting 3D graphics include hardware GPU decryption and image-based rendering. If the 3D models were encrypted using public-key encryption when they are created, then custom GPUs could accept encrypted data and perform on-chip decryption and rendering. This technology would provide robust protection for the 3D data, but such GPUs do not exist today, and it will be years before it is widely available in personal computers. Image-based graphics data representations (such as light fields [2]) are densely sampled data structures that do not explicitly include a geometric description for 3D shape yet are still amenable to interactive and accurate display. However, distributing 3D models as image-based light fields at the high sampling resolutions required would involve huge, unwieldy file sizes and not allow for geometric operations (such as surface measurements performed by archaeologists) on the data.

A final scheme for securing 3D graphics is to retain the 3D model data on a secure server—controlled by the content owner—and pass only 2D-rendered images of the models back to user-client requests. The 3D geometry is thus safe from all types of graphics pipeline attacks (except reconstruction from images), though the server itself is still vulnerable to direct attack.

Remote Rendering to Protect 3D Models

We have implemented such a remote rendering system with a client-server architecture to provide controlled, protected access to collections of 3D graphics models (see Figure 2). Users employ a special 3D client viewer program to interactively view the protected 3D content. The program includes low-resolution, decimated versions of the 3D models that can be interactively rotated, zoomed, and illuminated by the user in real time. When the user stops manipulating a low-resolution model, detected by a "mouse up" event, the client program queries the remote rendering server via the network for a matching image rendered from the high-resolution model data, replacing the low-resolution rendering seen by the user (see Figure 3). On computer networks with reasonably low latencies, the user has the impression of manipulating a high-resolution version of the model. In typical use involving cultural heritage artifacts, we use models with approximately 10,000 polygons for the low-resolution version, whereas the server-side models often contain tens of millions of polygons. Low-resolution model complexities are of little value to potential thieves yet still

provide enough clues for the user to navigate and manipulate.

The remote rendering server receives rendering requests from users' client programs, renders corresponding images, and passes them back to the clients. It is implemented as a module running under the Apache 2.0 HTTP server, communicating with client programs using the standard HTTP protocol and taking advantage of the access-protection and monitoring tools built into the Web server software. As render requests are received from clients, the server checks their validity and dispatches valid requests to a GPU for OpenGL hardware-accelerated rendering. The rendered images are read back from the frame buffer, compressed using JPEG compression, and returned to the client. The server uses level-of-detail techniques to speed the rendering of highly complex models and maintain high throughput rates. In practice, an individual server node with, say, Pentium 4 CPU and NVIDIA GeForce 4 video card can handle a maximum of eight typical client requests per second; the bottlenecks are in the rendering and readback stage (about 100 milliseconds) and in the JPEG compression step (about 25 milliseconds). Incoming request sizes are about 700B each; the images returned from our servers average 30kB per request.

The primary benefit of using a remote image-rendering system to share 3D models is that the high-resolution model geometry data is never made available to potential attackers. Only 3D reconstruction from 2D images remains as a possible attack from those outlined in Figure 1, but general 3D reconstruction from images is a challenging computer vision research problem. However, synthetic graphics renderings can be particularly susceptible to reconstruction, as the human cost of harvesting large numbers of images is low, and the attacker may be able to specify the parameters used to create the images. Moreover, synthetic images are potentially perfect, with no sensor noise or miscalibration errors.

To combat such reconstruction attacks, we've implemented a number of defenses in our rendering server system. To deter image-harvesting attacks, we perform automatic analysis of the server logs, detecting suspicious sequences or frequencies of image requests. We employ obfuscation to create hurdles for attackers by encrypting the rendering request messages sent from the client programs, as well as by encrypting the low-resolution client-side 3D models. The server imposes constraints on rendering requests, disallowing extremely close-up views of models and requiring a fixed field of view.

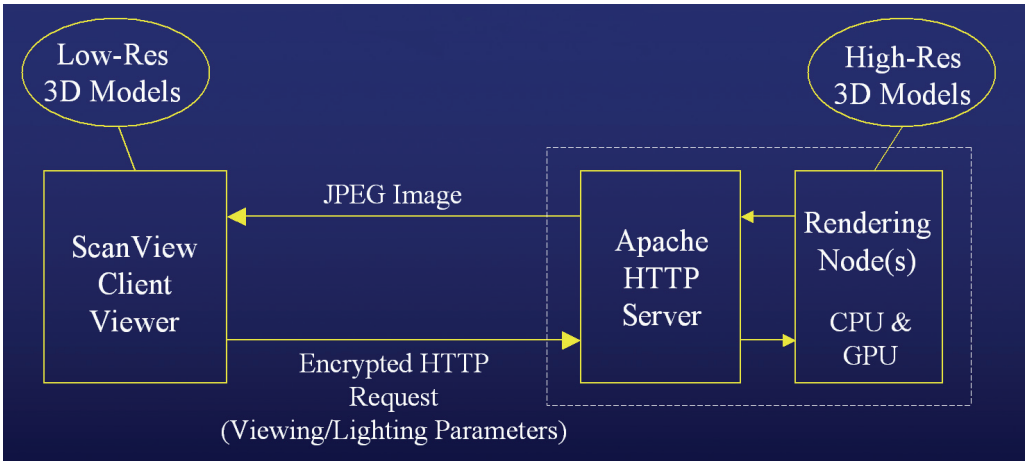

Figure 2. Remote rendering system.

Finally, we employ a number of perturbations and distortions to the images returned from the server. They are generally applied in a pseudorandomly generated fashion, so their effects are not easily modeled and reversed, and their magnitude is limited so as not to distract nonmalicious users viewing the models. Examples of such distortions include nonlinear image warps, adding high-frequency noise to images, and perturbing the lighting parameters slightly from those being requested.

We have experimentally validated the effectiveness of these defenses against a variety of traditional computer vision reconstruction techniques [1]. However, we know of no formalism for rigorously analyzing the security provided by our systems-based approach. One inevitably falls into an "arms race" between attacks and countermeasures (such as the ones we've implemented).

Results and Future Work

The protected-graphics software we developed—ScanView, available at graphics.stanford.edu/software/scanview/—has been used to share 3D models from the Digital Michelangelo Project and other collections of cultural heritage artifacts. More than 10,000 users have installed the client software on their computers and accessed the remote servers to view the 3D models. Art students, sculptors, and

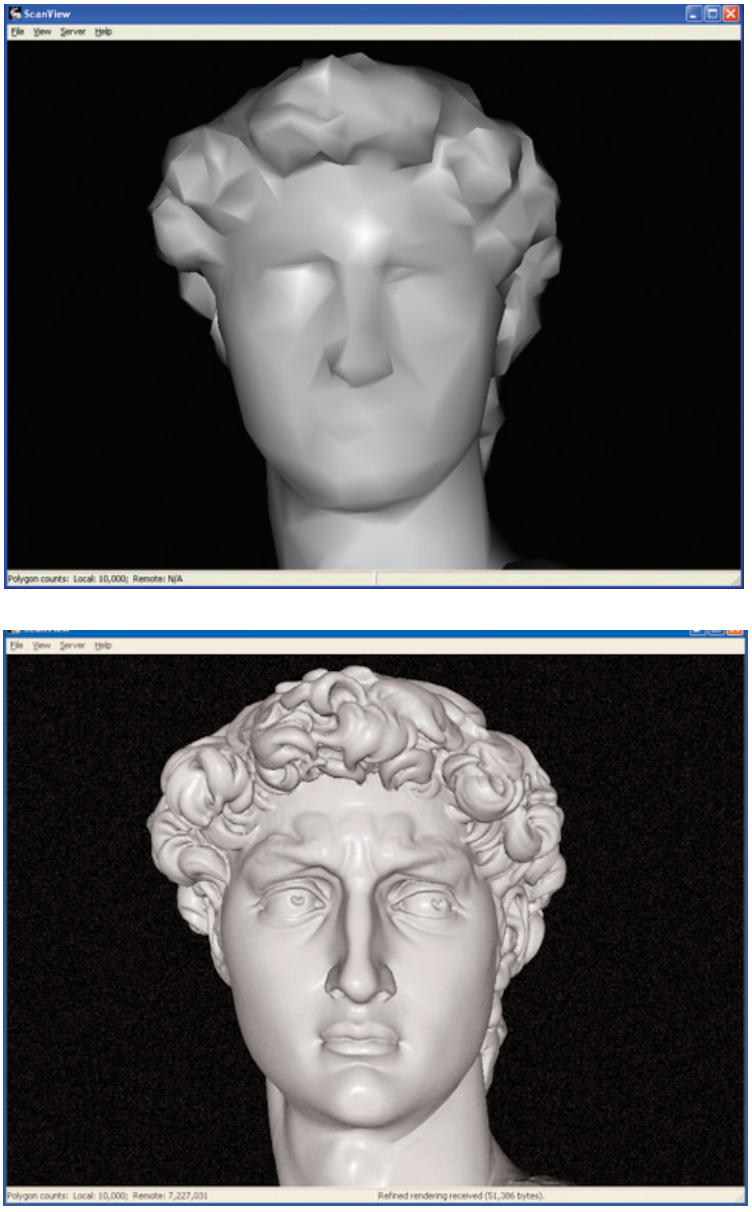

Figure 3. Client-side low-resolution (top) and server-side high-resolution (bottom) model renderings.

enthusiasts have examined the high-resolution artwork, while archaeologists have studied individual artifacts. Few of them would have qualified under the strict guidelines required to obtain unrestricted access to the models, so the protected remote rendering system has made it possible to grant whole new categories of users access to 3D graphical models for both professional scholarship and personal enjoyment.

User comment is uniformly positive. Fetching high-resolution renderings over intercontinental broadband Internet connections involves less than two seconds of latency, while fast continental connections generally experience latencies dominated by the processing time of the rendering server. Moreover, the render server architecture scales up to support an arbitrary number of requests per second, and servers can be installed at distributed locations around the world to reduce long-distance latencies.

One direction for further research is analysis of computer vision techniques that specifically address 3D reconstruction of synthetic data under antagonistic conditions—to increase our understanding of the efficacy of such attacks and the corresponding render server defenses. Another issue is how to grant users a greater degree of geometric analysis of protected 3D models without further exposing the data to theft. Scholarly users have expressed interest in measuring distances and plotting profiles of 3D objects for analytical purposes beyond the simple 3D viewing supported in the current system. Finally, there is general interest in alternative approaches to protecting 3D graphics, including specialized systems that make data security a priority while sacrificing general-purpose computing platform capabilities. A GPU-decryption scheme might, for example, be appropriate for console devices and other custom graphics systems. C

References

1. Koller, D., Turitzin, M., Levoy, M., Tarini, M., Crocia, G., Cignoni, P., and Scopigno, R. Protected interactive 3D graphics via remote rendering. *ACM Transact. Graphics 23,* 3 (Aug. 2004), 695–703.
2. Levoy, M. and Hanrahan, P. Light field rendering. In *Proceedings of ACM SIGGRAPH 1996* (New Orleans, Aug. 4–9). ACM Press, New York, 1996, 31–42.
3. Levoy, M., Pulli, K., Curless, B., Rusinkiewicz, S., Koller, D., Pereira, L., Ginzton, M., Anderson, S., Davis, J., Ginsberg, J., Shade, J., and Fulk, D. The Digital Michelangelo Project. In *Proceedings of ACM SIGGRAPH 2000* (New Orleans, July 23–28). ACM Press, New York, 2000, 131–144; graphics.stanford.edu/projects/mich/.
4. Praun, E., Hoppe, H., and Finkelstein, A. Robust mesh watermarking. In *Proceedings of ACM SIGGRAPH 1999* (Los Angeles, Aug. 8–13). ACM Press, New York, 1999, 49–56.
5. Schneier, B. The fallacy of trusted client software. *Information Security* (Aug. 2000).

David Koller (dk@cs.stanford.edu) is a Ph.D. student in the Computer Science Department at Stanford University, Stanford, CA. **Marc Levoy** (levoy@cs.stanford.edu) is an associate professor of computer science and (jointly) electrical engineering at Stanford University, Stanford, CA.

This work has been supported in part by National Science Foundation contract IIS-0113427 and the Max Planck Center for Visual Computing and Communication.