Virtual Inspector: A Flexible Visualizer for Dense 3D Scanned Models

Marco Callieri, Federico Ponchio, Paolo Cignoni, and Roberto Scopigno
Institute of Information Science and Technologies, Italian National Research Council

Virtual Inspector lets novice users inspect dense 3D models at interactive frame rates on commodity PCs. The system obtains visualization efficiency without sacrificing quality by adopting a continuous level-of-detail representation. Visual Inspector's use of XML to encode the GUI's structure and behavior makes it flexible and configurable.

The rapid evolution of automatic shape acquisition technologies will make a huge amount of sampled 3D data available in the near future. The cultural heritage domain is an ideal field for applying 3D scanning technologies (see the "Projects in 3D Scanning and Cultural Heritage" sidebar). Although these technologies still present some issues both in acquiring the 3D data (see the "3D Scanning Technical Issues" sidebar) and in their latter practical use, their potential impact in the cultural heritage field is clearly significant.

Interactive inspection and rendering of complex 3D models is crucial to many applications, such as architectural design, graphics simulation, and scientific visualization. Users of these systems need realistic and accurate visual representations and real-time navigation and interaction tools. Ease of use is also an important issue in the design of visualization tools for cultural heritage applications, because most of these applications' users (such as museum curators, art historians, restorers, and museum visitors) have limited skills in managing 3D graphics technology. The GUI's design and the rendering tool's overall usability play a critical role in determining whether the tool will be just a nice toy or a useful technical instrument. Another problem is that the complexity (the huge number of graphics primitives) of accurate, realistic-looking models exceeds the interactive capabilities of most graphics workstations. We need real-time visualization of those huge meshes without sacrificing the high quality achievable with 3D scanning technologies.

Virtual Inspector, our cultural-heritage-oriented system, aims to manage some of these issues.

Virtual Inspector

We designed Virtual Inspector with the following goals:

- *Commodity graphics workstation tailoring*. The system should attain maximal performance from inexpensive platforms (PCs with mainstream graphics boards) to ensure the widest user community.
- *User tailoring*. Because most of the system's users are novices, interaction with 3D data should be as easy and natural as possible. Usability should have priority over the flexibility and completeness of the rendering features.
- *High-quality interactive rendering*. Users should be able to interactively browse highly dense 3D models without degrading performance or rendering quality. We adopted a continuous level-of-detail (LOD) approach[1] for the online selection of best-fit geometry, using for each rendering a data resolution adequate to the current visualization task.
- *Integration of multimedia data*. An exposition curator or art historian should find it easy to enrich the 3D model with related multimedia information. We therefore added a hot-spot feature (that is, interactive links to other information associated with single points over the 3D model surface) and enabled the system to operate with standard Web browsers.
- *Local and remote access*. The system should be able to run both locally (for example, on a multimedia kiosk in a museum) and remotely, allowing Web access to huge 3D models through a thin client able to run even over obsolete hardware.
- *Data protection over the Web*. Scanning high-quality 3D models, which could be considered sensible data, requires some effort. Therefore, in some cases, we need to protect the 3D data and prevent its uncontrolled distribution.

 Published by the IEEE Computer Society 0272-1716/08/$25.00 © 2008 IEEE

In addition, we took into account our experiences with our first cultural-heritage-oriented visualization tool.[2]

We designed the system architecture using a triangle-based approach to 3D data management. To support interactive presentation of massive models, Virtual Inspector adopts an out-of-core multiresolution approach. The system extracts view-dependent variable resolution representations on the fly using a highly efficient approach.[3] In the first version of Virtual Inspector, we used a discrete LOD approach. For each frame, Virtual Inspector selects the best-fit variable-resolution LOD according to the current view frustum and the requested visualization accuracy. LOD selection and rendering are efficient because we use a representation compatible with graphics processing units (GPUs). With this new data representation, we use a coarse-grained patch-based multiresolution hierarchy that, during rendering, is efficiently traversed to recover a set of ready-to-render geometry patches. During rendering, the system therefore doesn't process 3D data at the single-triangle grain, but fetches a read-to-render set of

Projects in 3D Scanning and Cultural Heritage

Researchers have presented many significant projects involving 3D scanning and cultural heritage in the past few years. Here we present a short but representative selection.

- M. Levoy et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Proc. Computer Graphics* (ACM Siggraph), Addison Wesley, 2000, pp. 131-144.
- F. Bernardini et al., "Building a Digital Model of Michelangelo's Florentine Pieta," *IEEE CG&A*, vol. 22, no. 1, 2002, pp. 59-67.
- J. Stumpfel et al., "Assembling the Sculptures of the Parthenon," *Proc. Int'l Symp. Virtual Reality, Archaeology, and Intelligent Cultural Heritage* (VAST), Eurographics, 2003, pp. 41–50.
- M. Pollefeys et al., "Image-based 3d Acquisition of Archeological Heritage and Applications," *Proc. Int'l Symp. Virtual Reality, Archaeology, and Intelligent Cultural Heritage* (VAST), ACM Siggraph, 2001, pp. 255-261.
- M. Callieri et al., "Visualization and 3D Data Processing in David Restoration," *IEEE CG&A*, vol. 24, no. 2, 2004, pp. 16-21.

3D Scanning Technical Issues

Many researchers use 3D technology to reconstruct digital 3D models of cultural heritage masterpieces or to present those models through digital media. An exhaustive description of these works goes well beyond this article's scope. Instead, we cite only some seminal papers about the technologies proposed for building the models (mostly 3D scanning) and for supporting their interactive visualization.

Automatic 3D reconstruction technologies[1] have evolved significantly in the past few years. Unfortunately, instead of a final, complete 3D model, most 3D scanning systems produce a large collection of raw data (range maps) that must be postprocessed. Bernardini and Rushmeier present an excellent overview of the postprocessing pipeline.[2] Solutions to some algorithmic subtasks have improved a lot since their review article. The most notable progress is in the areas of semiautomatic alignment[3]; surface reconstruction from samples, with many new solutions based on pointset representations[4,5]; and automatic reconstruction from uncalibrated sequences of high-resolution photographic images.[6] These software and hardware improvements make 3D scanning a viable, sufficiently fast, and affordable modeling option for cultural heritage purposes.

Many of these projects considered data-management issues, in particular how to process and render the high-resolution meshes produced with 3D scanning at interactive frame rates. Several techniques cope with this problem. Some keep a triangle-based representation and adopt geometry simplification and multiresolution representation[7] to reduce data complexity and rendering times. Others adopt a point-based rendering approach, coupled with keen heuristics for dynamic data subsampling.

Most previous articles focused only on the problem of efficient rendering, rather than presenting a complete data-visualization tool. The person in charge of setting up a museum kiosk or a virtual exposition still has a complex task. Academia has produced several well-designed algorithmic solutions, but most require significant effort to use in a standard application domain, and they're usually limited to the pure rendering of the 3D data. On the other hand, a small number of commercial tools can set up pleasant multimedia presentations that, in some cases, offer sophisticated rendering options for 3D models, such as Cult3d, Macromedia Director, and Turn Tool. But those commercial tools are designed to work with standard low-to-medium complexity 3D models and don't support the efficient rendering of dense 3D scanned meshes. When using these tools, the only viable option is to aggressively simplify the original data sets, discarding most of the sampled digital model's detail and accuracy.

References

1. N. Amenta and Y.J. Kil, "Defining Point-set Surfaces," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 264-270.
2. F. Bernardini and H.E. Rushmeier, "The 3D Model Acquisition Pipeline," *Computer Graphics Forum*, vol. 21, no. 2, 2002, pp. 149-172.
3. B. Curless and S. Seitz, "3D Photography," *ACM Siggraph Course Notes*, no. 19, ACM Press, 2000.
4. A. Fasano et al., "Automatic Registration of Range Maps," *Computer Graphics Forum*, vol. 24, no. 3, 2005, pp. 517-526.
5. Y. Ohtake et al., "Multi-level Partition of Unity Implicits," *ACM Trans. Graphics*, vol. 22, no. 3, 2003, pp. 463-470.
6. S. Rusinkiewicz and M. Levoy, "QSplat: A Multiresolution Point Rendering System for Large Meshes," *ACM Trans. Graphics*, vol. 10, no. 3, 2004, pp. 343-352.
7. T. Tuytelaars and L. Van Gool, "Matching Widely Separated Views Based on Affine Invariant Regions," *Int'l J. Computer Vision*, vol. 59, no. 1, 2004, pp. 61-85.

Figure 1. User interface of the Virtual Inspector tool, showing the *Arrigo VII* mesh.

triangles from disk and copies them on GPU memory with the maximal rendering efficiency.

Virtual Inspector is mainly designed for visualizing single works of art (sculptures, pottery, architecture, and so on) and adopts an intuitive approach to guiding the virtual manipulation and inspection of the digital replica, based on a straightforward metaphor. We provide a dummy representation (that is, a small complete view of the inspected object) of the current inspected model on a side of the screen, which we can rotate on its axis. To select a view, the user points with the mouse to the corresponding point on the dummy.

Virtual Inspector also supports the specification of 3D links and popups and smoothly interoperates with Web browsers. Hot spots on 3D models are a handy resource for associating multimedia data (such as HTML pages and videos) with selected locations of a 3D model. This lets presentation authors design interactive presentations in which the 3D model acts as a natural visual index to historical and artistic information, presented using standard HTML formats and browsers.

Finally, Virtual Inspector is flexible and configurable. The authoring user can easily specify the entire interface and its behavior via XML coding. For example, the user can specify

- the 3D models to render (single or multiple meshes),
- the system layout characteristics (that is, how the different models and GUI components will appear on the screen),
- the rendering modes (for example, standard Phong-shading of sampled geometry or bidirectional reflectance distribution function rendering),
- the interaction mode (for example, model manipulation via a standard virtual trackball, via the dummy-based point-and-click interaction, or both), and
- links to multimedia external information.

GUI design

Figure 1 shows Virtual Inspector's standard GUI layout, which effectively balances ease of use and freedom of inspection. The tool's output window consists of two main frames. The right frame lets users interactively select the desired view and GUI; the left frame displays the full 3D model using the chosen view.

The dummy is visualized in the right-hand frame of Figure 1. This dummy is conceived as an interactive and intuitive 3D map that lets users select the view specs. The dummy is always visualized as a whole. The current view doesn't affect its appearance or its level of detail (the dummy is visualized using a fixed low-resolution model). The user can see all sides of the dummy by rotating it (on its vertical axis) with a simple button-driven interface.

We implemented the viewing parameter selection (for the frame on the left) using a simple direct-manipulation approach. When a user clicks on any point on the dummy, the selected point becomes the new gaze point, and the new viewing direction is set by default to be equal to the averaged surface normal in the gaze point (see Figure 2). The viewfield is initially set to a default value (in the example in Figure 2, approximately 20 percent of the object is in the current view volume), then it defaults to the value used or set in the previous interaction. Clicking on the (zoom-in/zoom-out) GUI buttons controls the field of view value (see Figure 1).

To be more specific, we set the view direction equal to the average surface normal computed over a small mesh portion centered on the gaze point. Doing so prevents potential erratic or random values of the view direction in locations where the surface normal changes rapidly (for example, on rough surfaces or on highly convex or concave regions).

For more advanced users, skilled in standard 3D interaction techniques, a virtual trackball is also available (far-left frame in Figure 2). The trackball lets users more precisely specify rotations or pan actions around the gaze point, supporting a more accurate and versatile object manipulation.

From a technical viewpoint, the two frames present two different coordinate systems:

- The right-hand frame uses a constrained coordinate system that models the space containing the dummy.
- The left-hand frame uses the coordinate system derived by the current view specs, which the user selects by clicking on the dummy or using a virtual trackball on the mesh.

This interface design reflects most novice users' familiarity with point-and-click interfaces, whereas few are familiar with standard computer graphics interaction tools (such as the virtual trackball approach).

Geometric data management

To guarantee a satisfying interaction, the system's rendering engine must be able to provide an interactive frame rate without sacrificing the models' quality. Researchers have proposed many solutions for efficiently visualizing large, complex digital 3D models. In

Figure 2. The user selects a new gaze point (circled in red for illustration purposes) by clicking on the corresponding location on the dummy.

designing our rendering system, we face several issues, including:

- *Choosing the right resolution.* 3D scanned models at full resolution often consist of more rendering primitives than the number of pixels in a standard view. Using a dense representation (more triangles than pixels) for rendering is rarely the wisest choice, because the visual improvements over using a model with a smaller number of triangles typically wouldn't be perceptible.
- *Culling unnecessary geometry*. Given a particular view and resolution, only the visible geometry should be rendered.
- *Keeping in memory only the required data*. Digital models can consist of tens or even hundreds of millions of triangles, and rendering them at full resolution can require many gigabytes of RAM. Thus, only the currently viewed portion of the model should be loaded in main memory, possibly in a format that allows maximal rendering performance on modern GPUs.

Until recently, most view-dependent LOD methods were based on multiresolution structures that make decisions at the triangle or vertex primitive level. This approach requires a constant CPU workload for each triangle. On modern GPUs, this approach is doomed to make the CPU the bottleneck of the entire rendering process. Overcoming this bottleneck and fully exploiting current graphics hardware's capabilities requires selecting and sending batches of geometric primitives to be rendered with just a few CPU instructions.

Virtual Inspector's rendering engine adopts a batched multiresolution framework[3] based on the GPU reinterpretation of the multitriangulation (MT) approach.[4] The MT is a general framework encompassing a wide class of multiresolution algorithms. However, like most techniques proposed in the 1990s, its original goal was to minimize the number of triangles to be rendered, at the expense of CPU time. Therefore, we redesigned the MT scheme in a GPU-friendly fashion by moving the framework's granularity from single triangles to optimized surface patches and by redefining the construction and rendering algorithm to work on external memory (mandatory for managing huge scanned meshes).

Our batched multitriangulation (BMT) represents the 3D model by building a direct acyclic graph (DAG), in which each element represents a ready-to-render patch of a few thousand triangles (see Figure 3 on the next page). BMT thus reduces the DAG size by orders of magnitude with respect to standard solutions. The per-frame workload is therefore also much lower, involving only the runtime assembly of an adequate puzzle of preassembled optimized surface patches. By grouping sets of triangles (and representing them in the most GPU-efficient manner), we alleviate the CPU/GPU bottleneck. Because the granularity is much coarser than that of a standard MT representation, the CPU workload for multiresolution data structure management is two or three orders of magnitude lower.

The patches are also the basic units for arranging the data set in an out-of-core fashion. The multiresolution system continuously updates a dynamic set of the patches that will probably be needed in further rendering cycles and keeps them in memory using a nonblocking multithreaded approach.

Extracting a variable resolution model—that is, choosing the right set of patches to be rendered—means extracting a cut over the DAG (see Figure 3a), an action that can be performed efficiently when the DAG size is small, as in our case. For the sake of interactivity, the multiresolution extraction process should be able to support a constant frame rate, given the available time and memory resources. We therefore choose an extraction algorithm that can find a cut over the DAG within a budgeted amount of time and memory resources and that always

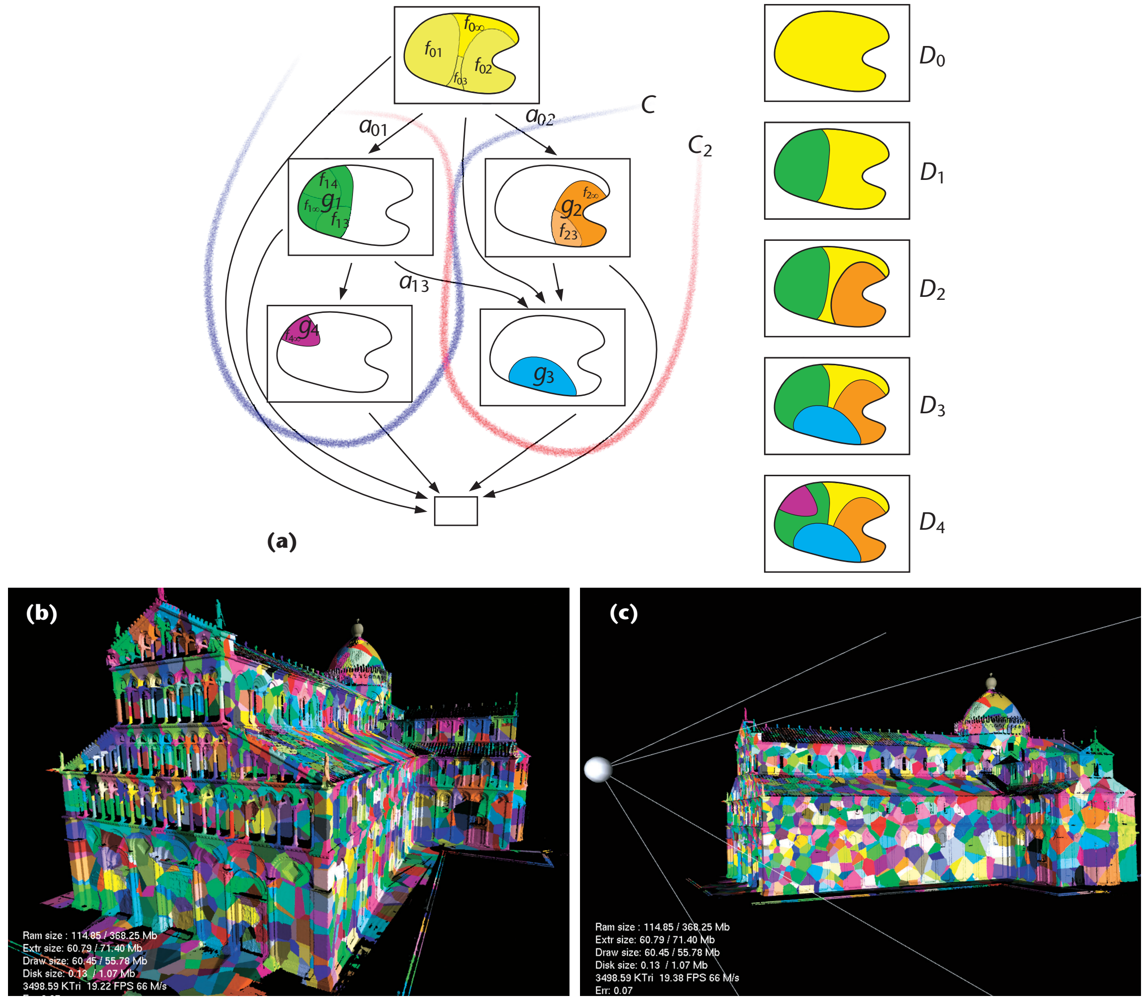


Figure 3. (a) An example of the multitriangulation direct acyclic graph showing the one-to-one correspondence between valid subsequences and valid cuts. (b-c) A single cut extracted from the Pisa Cathedral model, in which each patch is rendered with a uniform color. The image on the right (same model as the one on the left, with the small sphere indicating the viewpoint used for the multiresolution extraction) shows how the space covered by each patch increases with its distance from the viewpoint.

ends with a consistent result—in other words, it's interruptible. With this approach, eventually sacrificing some resolution, we can sustain an interactive frame rate.

We've evaluated the extraction and rendering of a dynamic continuous LOD with the BMT scheme over several inspections, rotating and abruptly zooming in and out on the model. We performed these tests using different rendering window sizes on a PC equipped with an ADM Athlon 64 clocked at 2.21 GHz, 2 Gbytes of RAM, Windows XP Professional 2002, and an Nvidia 6600GT graphics board.

The results demonstrate that the rendering speed depends mostly on the GPU's vertex processing speed, rather than the display resolution. On the selected GPU, we can render around 220 million vertices per second (on the Cathedral data set). Table 1 presents some numeric results, obtained from the Pisa Cathedral (390 million triangles) and the color-mapped double *David* (112 million triangles) data sets. We measure the accuracy threshold used in the variable LOD's real-time extraction in terms of screen pixel units (see the max error value in Table 1). This accuracy threshold is an estimate in excess of the actual error (the maximal upper bound of the error), which is usually much larger than the actual error.

Table 1. Rendering performances of the batched multitriangulation representation.

Data set	Display size	Vertices per frame (millions)	Frames per second	Max error
Pisa Cathedral	1,248 × 1,024	22.2	10	1.3 pixels
	800 × 600	22.2	10	0.8 pixels
	1,248 × 1,024	12.3	18	2.0 pixels
Double *David*	1,248 × 1,024	11.4	19	0.9 pixels
	1,600 × 1,200	11.4	19	1.2 pixels

Rendering modes

Virtual Inspector supports the interactive modification of lighting to simulate in real time the *luce radente* (grazing light) effect, which art historians and restorers often use in real inspections to enhance the visualization of small-scale surface detail.

For rendering, Virtual Inspector supports an ambient occlusion-enhanced rendering mode, rendering based on bidirectional reflectance distribution function (BRDF), and protected remote rendering over the Web.

Ambient occlusion for enhanced rendering

A common computer graphics problem is evaluating the trade-off between quality and efficiency in the rendering process. For visualizing models of real statues, the standard local lighting model used in interactive graphics is unsatisfactory. Standard OpenGL lighting doesn't account for the effects of cast shadows, which are important for perceiving shape. Moreover, artists usually account for these lighting effects when creating a statue.

Figure 4 illustrates the visual difference of cast shadows on the *David*. The only difference between the left and right OpenGL rendering is in the lighting: the left shows a standard Phong direct lighting; the right shows an approximation of a diffuse illumination lighting environment in which we properly compute cast shadow. The Phong lighting model uses a simple per-scene constant lighting term for all portions of the scene that aren't directly lit, but this approach leads to a notable flatness. Landis improved this approach by explicitly computing the accessibility value—that is, the percentage of the hemisphere (the sky) above each surface point not occluded by geometry—of each point on the surface.[5] 3D graphics practitioners use this technique, called *ambient occlusion*, in many production environments to add an approximation of the shadowing of diffuse objects lit with environment lighting. We successfully used this precomputed approach in the previous version of our visualization system.[2]

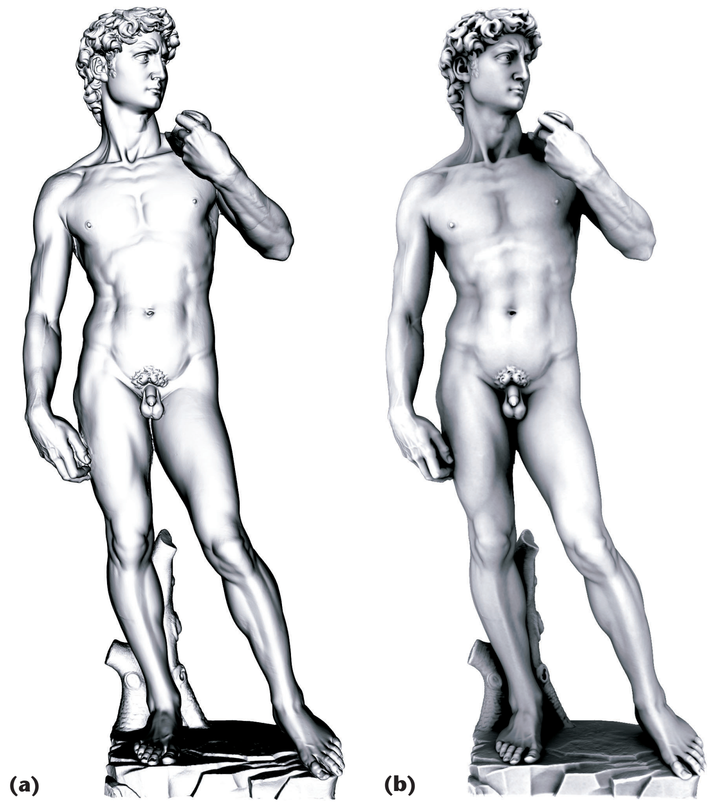


Figure 4. Rendering cast shadows affects the resulting visual presentation of sculpted surfaces. (a) A standard OpenGL rendering with Phong lighting, and (b) an OpenGL rendering with precomputed ambient occlusion lighting.

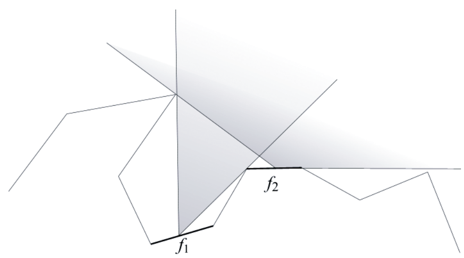


Figure 5. To better approximate the ambient lighting term, we compute the solid angle of the sky that can be seen from each face or vertex (the larger the solid angle, the more lighted the face).

Figure 5 illustrates the basic idea behind the ambient occlusion approach. Face f_1 is darker than face f_2 because it sees a smaller portion of the sky. We perform this per-face lighting computation statically in a preprocessing phase. The mesh's fine tessellation guarantees a sufficiently good sampling and rendering of the ambient occlusion term onto the surface. In Virtual Inspector, we then combine this precomputed ambient term with standard OpenGL direct lighting, letting the user interactively redirect the lighting to better perceive the inspected objects' shape.

This hybrid approach (static diffuse lighting field plus dynamic headlight) gives us a view-dependent lighting that produces shiny reflections that move dynamically over the surface depending on the current view specs.

BRDF-based rendering

BRDF describes in a precise way how light is reflected off an object's surface. To increase the rendering realism, we extended Virtual Inspector to render 3D models using accurate sampled BRDF specifications. In other words, we use standard BRDF to describe the behavior of ideal objects, whose surface consists of a single homogeneous material. Most objects, however, consist of several materials, or of a single corrupted material. This is especially true for artworks or archeological objects (see, for example, Figure 6 on the next page, which shows the Minerva head rendered using a sampled BRDF). A precise way of representing these details is to assign a different BRDF to each surface point, resulting in a spatially varying BRDF. Without these details, objects tend to look artificial and unrealistic. Virtual Inspector's spatially varying BRDF approach follows Lensch et al.'s sampling and rendering approach.[6] We implemented this approach efficiently by coding the spatially varying BRDF's parameters into textures, and exploiting the features of modern programmable GPUs

Figure 6. Virtual Inspector rendering of the Minerva head using the bidirectional reflectance distribution function (BRDF) model sampled from the real statue.

to compute in real time the resulting lighting (in collaboration with our Max Planck Institute colleagues and in the framework of the European Union Information Society Technologies' Virtual Heritage: High-Quality and Acquisition project).

Protected remote rendering

One of our goals is for Virtual Inspector to be a tool that could also be used on the Web. Large and accurate data are a problem for Web use. Moreover, the data owner could decide that the 3D mesh is valuable and shouldn't be transmitted to a generic unknown user.

To support Web-based visualization, we extended Virtual Inspector by adding a remote rendering mode and service. Instead of computing the 3D model's rendering locally, a dedicated remote server performs this task. The local machine receives only a reduced-resolution model to support user interaction. When the user selects a viewpoint, the local rendering client sends a request over the Internet to a rendering server (the only server that retains a copy of the full-resolution model). The server renders the model according to the view parameters and sends the resulting image to the client. This client-server interaction follows Koller et al.'s philosophy[7] and presents several benefits:

- It doesn't disclose the full-resolution 3D model to the final user, protecting it from improper use and data thieving.
- It dramatically reduces the amount of data necessary on the client. This is especially useful if data must be downloaded through the Internet and for casual users who will probably browse the models for a short time.
- Because a dedicated server manages and renders the full-resolution model, we can use a model much bigger than the one that could be normally processed on the local machine (because of transmission and local storage limitations).

The system's only drawback is the network communication's cost in terms of latency and network load and the need to set up a rendering farm infrastructure adequate for the forecasted access load.

We implemented the remote server using the multiresolution technology outlined earlier. This server can work with models of arbitrary resolution. The remote-rendering node is dedicated to rendering, which guarantees optimal performance. Moreover, we can apply server-side changes to enhance rendering (by augmenting model resolution or perfecting the rendering) without changing the clients. The user interaction with the 3D model and the application behavior, in addition to the small latency introduced by the network interaction, is identical to that of the local-rendering mode. A single server can respond to multiple clients and deal with different high-resolution models. We run some experiments on a server based on an Athlon 64 3500+, 2 Gbytes of RAM, 256-Mbyte Nvidia GeForce 6600, Ubuntu Linux OS, and an Internet connection running at 1.2 Mbits. We used two data sets in the experiment: the 3D model of the Pisa Cathedral (time-off-light scanning, 390 million faces) and a setup with two *David* models with color per vertex (2×50 million triangles, see Figure 10), with the following results:

- For the Cathedral, the time for an entire remote image refresh (image request sent to server + remote rendering + compression + transmission + decompression + viz onscreen) is between 750 and 2,000 milliseconds. Server-side remote rendering only requires 100 to 500 milliseconds, depending on the specific view.
- For the *David*, the entire image refresh took 150 to 650 milliseconds. For a large community of users, we could adopt a remote rendering farm composed by multiple rendering servers and managed by a render dispatcher.

XML-coded interface and behavior

We improved the configurability of the first version of Virtual Inspector by designing a simple interpreted language called NSP (from the first three consonants of Inspector). We use NSP to build up the interface, its appearance, and its behavior. The language exploits XML for all of XML's well-known advantages (availability of syntax-aware parsers, human readability, extensibility, and so on).

At startup, Virtual Inspector reads the .NSP file and configures itself according to the file's instructions. Therefore, the multimedia application's designer doesn't have to compile a new version of the system to obtain a new layout, but can simply design and distribute a new GUI layout by simply editing a new .NSP file.

The XML approach lets us easily change Virtual Inspector's look and feel. Compare, for example, the different layout and graphics of intermediate versions (see Figure 6) with the more professional *Arrigo VII*'s installation (see Figure 1), in which a professional graphic designer has redesigned the application's layout, including all icons and background graphics elements. To do this, the designer specifies in the XML initialization file the new background and icon images and the location on the screen of all GUI icons and elements. We didn't need to program or recompile Virtual Inspector. This task can easily be assigned to an operator with standard Web design competence. NSP also lets us customize the behavior of the GUI's various elements—such as change the approach used to browse the model, impose constraints, and define new mechanisms for navigating objects. A different application layout that shows the system's high degree of configurability is the double Minerva installation, in which we render two models of the Minerva statue taken at different stages of restoration in a coordinated/synchronized manner (see Figure 7).

Like the rest of the interface, the hot-spot specification is encoded in the current Virtual Inspector instance's .NSP specification, and modifications to the 3D model aren't required. We provide a simple 3D browser to the person implementing the multimedia presentation, letting that person load a 3D model and query the 3D coordinates of any point on the artifact's surface (by simply clicking on the corresponding point), generating the XML text portion to be included in the .NSP file. Then, we specify a new hot spot by adding the XML tag in the Virtual Inspector specification file. The hot-spot XML tag specifies the 3D location and the action to be triggered when clicking on the hot spot (for example, the HTML file's name and whether we want to open a multimedia page). After activation, the control passes to the HTML browser, while Virtual Inspector remains sleeping in the background, automatically regaining control of the interaction when the user closes the HTML page.

We can organize a museum installation using introductory HTML pages, which present some general artistic and historic information about the artwork. Some information might provide links to activate Virtual Inspector on single or multiple artifacts (see Figure 8 for an example using the *Arrigo VII* installation,[8] which presents a group of 13 statues).

Evaluation and use of Virtual Inspector

We conceived of the Virtual Inspector tool in the framework of a cooperation with the Restoration Laboratory of the Tuscan Archeological Superintendency (Florence, Italy) aimed at restoring the Minerva of Arezzo statue. We scanned the Minerva four times, at different stages of the restoration, starting in October 2000,[9] and produced models of up to 65 million triangles each (see Figure 7).

We planned different uses of the 3D model:

- monitoring the artifact's initial and postrestoration status (because the restorer forecast a major modification of shape and appearance);
- mapping and correlating the results of different investigation surveys (photo under visible and UV light, X-ray imaging, results of chemical analysis, and so on);
- being part of a multimedia archive that documents the complete restoration; and
- being part of a visual presentation (museum installations or multimedia CDs).

We therefore clearly needed an easy-to-use tool that would let restorers access the accurate and high-resolution digital 3D model. A basic requirement was obviously to have interactive access to the high-resolution model without compromising accuracy and detail. The first evaluations of the Virtual Inspector tool (2001-

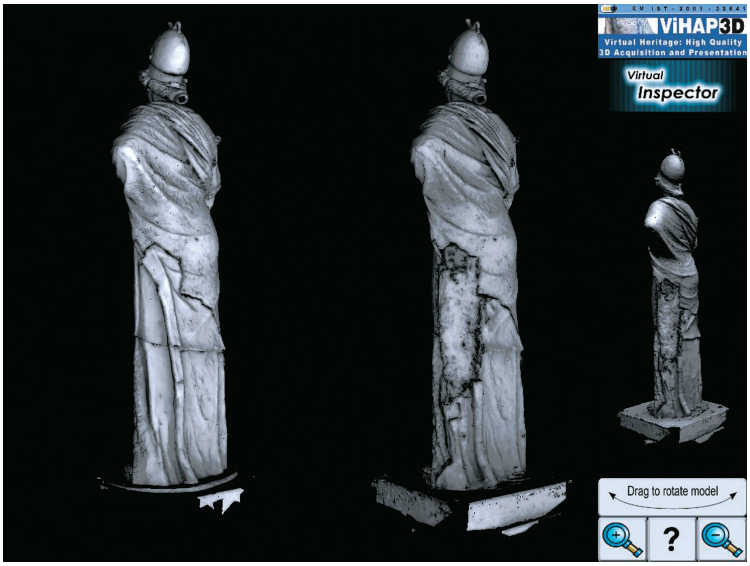


Figure 7. Virtual Inspector showing two models of the Minerva statue, scanned at different times during restoration.

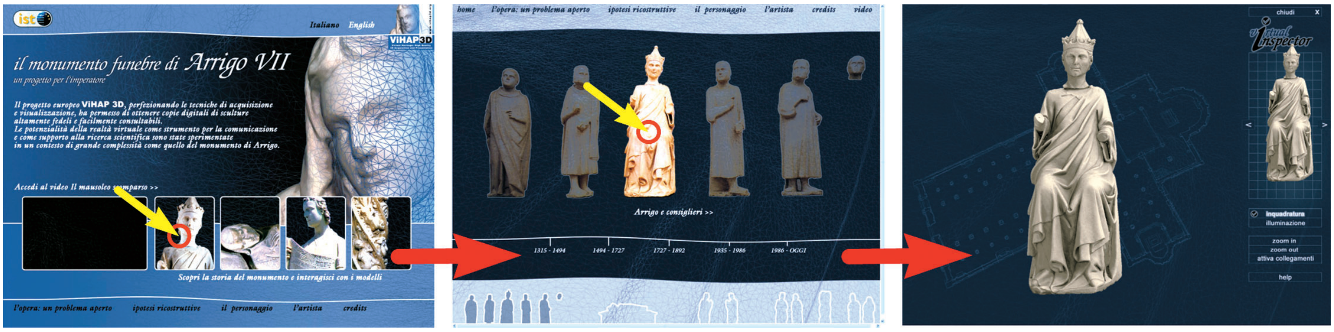


Figure 8. The initial screens of the *Arrigo VII*'s multimedia kiosk and a subindex page. To provide access to any statue of the *Arrigo VII* complex, we divided the statues into four groups (the middle image shows the index page related to the "*Arrigo VII* enthroned" and counselors' group). Users start Virtual Inspector by clicking on any of the statue icons (middle image). This starts a new session of Virtual Inspector (right-hand image).

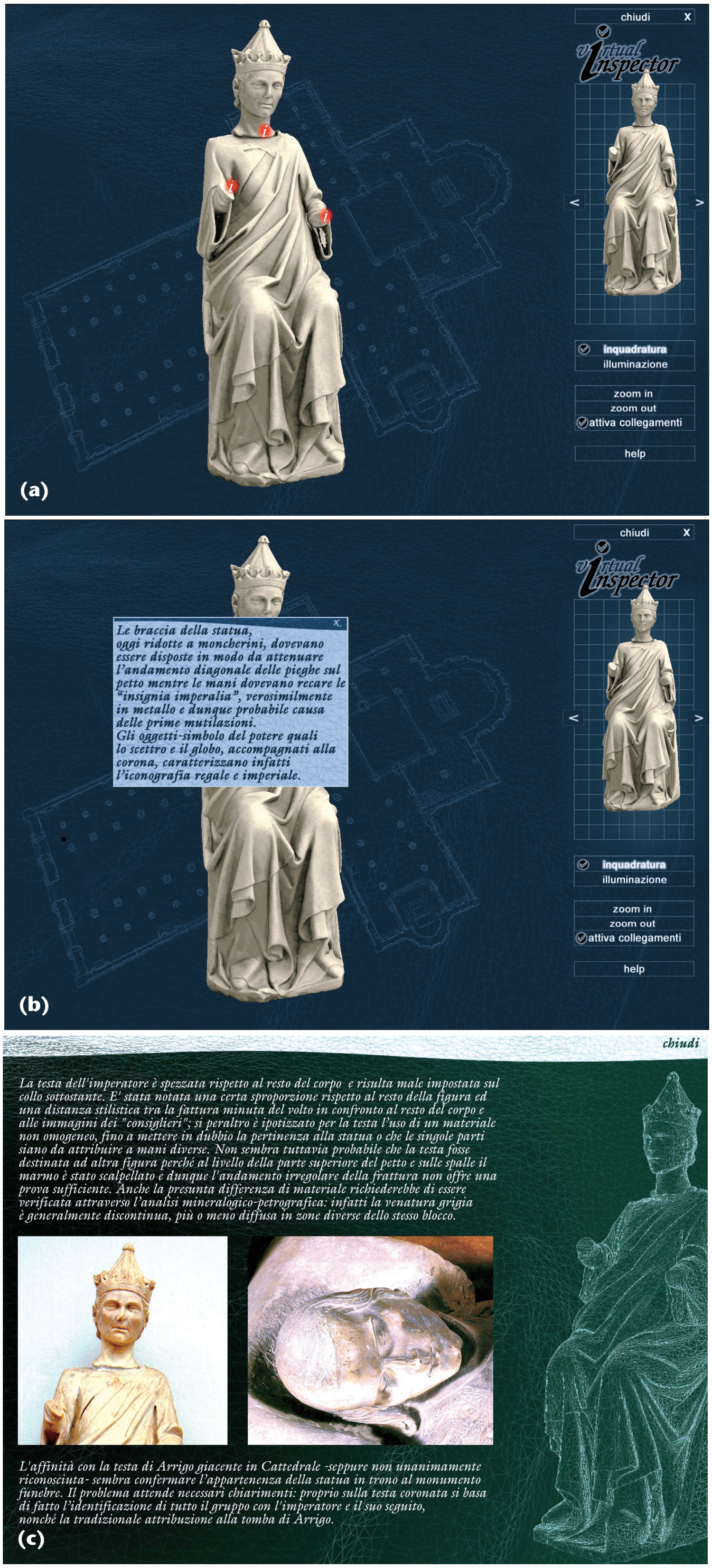


Figure 9. Views of Virtual Inspector. (a) The *Arrigo VII enthroned* statue rendered with active hot spots; (b) a popup panel, with information describing the missing hand, appears when the mouse passes over the hotspot; and (c) an example HTML page activated by clicking the hot spot on the neck.

2002) were encouraging, fulfilling our requirements. More specifically, the restorers were fascinated by the possibility of inspecting such a dense and accurate model so easily. They considered the ability to compare different digital models by referring to different time frames (see Figure 7) extremely useful for the restoration process and documentation. An interactive 3D visualizer can contribute to an artwork's inspection and analysis and can substantially improve restorers', art historians', and the ordinary public's knowledge. The Virtual Inspector system's didactic power is valuable: the capability to navigate and inspect a statue from any viewpoint with great accuracy allows new insights.

An example is the new installation that reports the condition of the Michelangelo's *David* statue before and after the restoration (2002–2004). This restoration was mainly a surface cleaning intended to remove brown crust and spots and replace plaster filling in small holes and marble cracks.[10] We've mapped two complete photographic campaigns (digital photographs depicting the status before and after the restoration, each composed of around 70 5-million-pixel images) to the *David*'s surface (using a per-vertex color mapping approach over the 56-million-triangle model). This installation allows an easy analysis of the recent restoration's effects (see Figure 10).

If we focus on art education, Virtual Inspector gives much more than the standard media used in art classes (still images and videos). Even the most complete set of 2D images can't give the same amount of information as an interactive navigation of a 3D model, and we must consider that most didactic applications don't make available a complete photographic sampling of a 3D artifact. In general, a student has just a few images depicting a given artifact. Moreover, the Virtual Inspector tool could be more informative than a personal inspection of the real artwork. For evident security reasons, museum visitors can't manipulate valuable artifacts, and many are so large that detailed inspection is impossible. The ability to closely inspect a given sculpture or architecture is a privilege that just a few art historians have. Modern 3D graphics tools can replicate a similar experience virtually for the general public.

In the last couple of years, we've successfully used Virtual Inspector in several museum exhibits and fairs. In most cases, the typically fearful attitude of nontechnical people toward 3D graphics shifted quickly to an attitude of interest, interaction, and amusement. We've run a specific quality and usability assessment of the *Arrigo VII* installation, onstage at the Museum of Pisa's Cathedral, since October 2004 (see Figure 11).

An external company, expert in evaluating museum and expositions and assessing user satisfaction, performed this end-user evaluation. They analyzed system performance by sampling museum visitors' reactions using a questionnaire and direct interviews and by analyzing the recording (via an automatic video-grabbing facility) of visitors' use of the multimedia kiosk.[11] The results were extremely positive, because most of the users were able to proficiently use the tool and wanted to see this experiment replicated for many other artworks in the museum (85 percent of the people who used the kiosk). Table 2 presents a subset of the results produced after processing the 330 questionnaires compiled by museum visitors. The complete results appear elsewhere.[11]

The *Arrigo VII* installation was successfully featured in various national and international shows, such as the 2006 Virtual Archeology Expo "Immaginare Roma Antica" in Rome and the 2007 "Interactive Salon" at the Stadsmuseum (Stockholm).

We also used Virtual Inspector in a temporary exposition at the Ferrara 2006 Restoration Fair (Italy) to demonstrate the results of the acquisition of the Cathedral of Pisa digital model, a huge architectural model obtained with time-of-flight scanning and totaling 390 million triangles.

Conclusion

We can improve the Virtual Inspector system by adding new functionality and increasing its efficiency. Few restorers would consider the simple visualization of a digital 3D replica as the ultimate goal for the use of 3D scanning technologies in cultural heritage contexts. Once we can visualize an artifact with great accuracy, the need to map other data on the 3D model will rise. We're working on a slightly extended version of the tool to support mapping and selective visualization of different sources of surface data (such as any type of 2D image). We're also developing tools to compute measures (for example, computing point-to-point lengths). Finally, we're working to more dynamically enrich the data linked to the mesh by letting users add annotations or link multimedia material to selected points of the artifact surface. Our goal is to transform Virtual Inspector into a more dynamic instrument for cultural heritage research and restoration. ■

Acknowledgments

We thank the Stanford Computer Graphics Group and Marc Levoy; the acquisitions group at MPI Saarbrucken (Hendrik Lensch, Michael Goesele, and Hans-Peter Seidel); our partners in the cultural heritage domain (Franca Falletti, director of the Galleria dell'Accademia Museum, Florence, Clara Baracchini, Cultural Heritage

Figure 10. Virtual Inspector showing two models of the *David* statue, in pre- and postrestoration status.

Figure 11. The multimedia kiosk developed for the Museum of the Cathedral at Pisa.

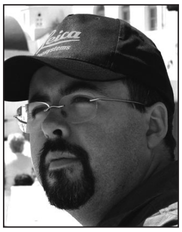
Table 2. Virtual Inspector usability results (numerical results are percentages of the 330 questionnaires returned).

Survey question	Response				
How easy is the kiosk to use?	Very easy: 26.3%	Easy: 68.7%	Complex: 1.7%	Very complex: 0.6%	No opinion: 2.8%
Is the kiosk more informative than panels?	Yes: 84.4%	No: 15.6%			
Did you use the 3D models?	Yes: 71.5%	No: 28.5%			
Are the 3D models really useful?	Yes: 92.1%	No: 7.9%			Percentage of those using 3D (71.5% of the total)
Are the 3D models easy to use?	Yes: 97.4%	No: 2.6%			Percentage of those using 3D (71.5% of the total)
Do you want similar kiosks for the other artworks?	Yes, for all artworks: 25.3%	Only for main artworks: 60.3%	No: 5.2%	No opinion: 9.2%	

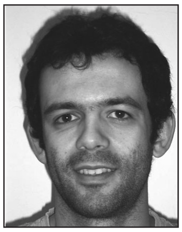
Superintendency in Pisa), and the archaeologists and restorers of the Archeological Restoration Laboratory in Florence. Several projects helped support this work, including EU IST-2001-32641 Virtual Heritage: High-Quality and Aquisition project, EU NoE IST-2002-507382 EPOCH, and the CNR Project Fruizione—Tecniche di supporto e modalitá innovative.

References

1. D. Luebke et al., *Level of Detail for 3D Graphics*, Morgan Kaufmann Series in Computer Graphics, Morgan Kaufmann, 2002.
2. R. Borgo, P. Cignoni, and R. Scopigno, "An Easy to Use Visualization System for Huge Cultural Heritage Meshes," *Proc. Virtual Reality, Archeology, and Cultural Heritage (VAST) 2001*, D. Arnold, A. Chalmers, and D. Fellner, eds., ACM Press, 2001, pp. 121-130.
3. P. Cignoni et al., "Batched Multi Triangulation," *Proc. IEEE Visualization*, IEEE CS Press, 2005, pp. 27-35.
4. E. Puppo, "Variable Resolution Terrain Surfaces," *Proc. 8th Canadian Conf. Computational Geometry*, Carleton Univ., 1996, pp. 202-210.
5. H. Landis, "Production Ready Global Illumination," *ACM Siggraph 2002 Course Notes*, ACM Press, 2002, pp. 331-338.
6. H.P.A. Lensch et al., "Image-Based Reconstruction of Spatial Appearance and Geometric Detail," *ACM Trans. Graphics*, vol. 22, no. 2, 2003, pp. 234-257.
7. D. Koller et al., "Protected Interactive 3D Graphics via Remote Rendering," *ACM Trans. Graphics*, vol. 23, no. 3, 2004, pp. 695-703.
8. C. Baracchini et al., "Digital Reconstruction of the Arrigo VII Funerary Complex," *VAST 2004*, Y. Chrysanthou et al., eds., AK Peters, 2004, pp. 145-154.
9. C. Rocchini et al., "3D Scanning the Minerva of Arezzo," *Proc. Int'l Conf. Hypermedia and Interactivity in Museums* (ICHIM), vol. 2, Archimuse Publishing, 2001, pp. 265-272.
10. S. Bracci et al., eds., *Exploring David: Diagnostic Tests and State of Conservation*, Giunti Press, 2004.
11. L. Solima, *Valutazione del caso di studio "Il monumento funebre di Arrigo VII"* [Case Study Evaluation of "The Funerary Monument of Arrigo VII"] tech. report, Megaride s.r.l., Feb. 2005 (in Italian).

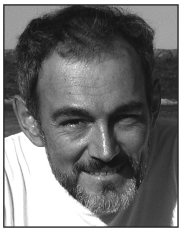


Marco Callieri *is a researcher at the Institute of Information Science and Technologies of the National Research Council in Pisa, Italy. His research interests include 3D scanning, color and appearance acquisition, and large 3D data set manipulation and rendering. He has a PhD in computer science from the University of Pisa. Contact him at m.callieri@isti.cnr.it.*



Federico Ponchio *is a research scientist at the Institute of Information Science and Technologies of the National Research Council in Pisa, Italy. His research interests include multiresolution and visualization interfaces. He has an advanced degree in mathematics (Laurea) from the University of Pisa and a diploma in mathematics from the Scuola Normale Superiore di Pisa. Contact him at f.ponchio@ isti.cnr.it.*

Paolo Cignoni *is a senior research scientist at the Institute of Information Science and Technologies of the National Research Council in Pisa, Italy. He received a PhD in computer science from the University of Pisa. His research interests include visualization and processing of huge 3D data sets, 3D scanning in the cultural heritage field, and scientific visualization. Contact him at p.cignoni@isti.cnr.it.*



Roberto Scopigno *is a research director at the Institute of Information Science and Technologies of the National Research Council in Pisa, Italy, and coleads the Visual Computing Lab. His research interests include 3D scanning, surface reconstruction, multiresolution, scientific visualization, and cultural heritage. Contact him at r.scopigno@ isti.cnr.it.*