Virtual Inspector: a flexible visualizer for dense 3D scanned models

M. Callieri, F. Ponchio, P. Cignoni, R. Scopigno
Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche*

June 6, 2007

Abstract

The rapid evolution of automatic shape acquisition technologies will make huge amount of sampled 3D data available in the near future. Cultural Heritage (CH) domain is one of the ideal fields of application of 3D scanned data, while some issues in the use of those data are: how to visualize at interactive rates and full quality on commodity computers; how to improve visualization ease of use; how to support the integrated visualization of a virtual 3D artwork and the multimedia data which tell its story.

We present here the VIRTUAL INSPECTOR tool. The system allows naive users to inspect a very dense 3D model at interactive frame rates on off-the-shelf PC's, presenting the 3D model and all the multimedia data that has been linked to selected points of its surface. A main goal in the design of the system was to provide the user with a very easy and natural interaction approach, based on a straightforward "point and click" metaphor. Visualization efficiency is obtained, without sacrificing quality, by adopting a state-of-the-art *continuous* level-of-detail (LOD) representation. Finally, the adoption of XML encoding of the GUI structure and behavior, makes VIRTUAL INSPECTOR a very flexible and configurable system.

1 Introduction

Interactive inspection and rendering of complex 3D models is crucial for many applications, such as architectural design, graphics simulation and scientific visualization. Users need to be provided of realistic and accurate visual representations and real-time navigation/interaction tools. A specific issue in the design of visualization tools for Cultural Heritage (CH) applications is ease of use,

---

*Area della Ricerca CNR, Via Moruzzi 1, 56125 Pisa, ITALY. WWW: `http://vcg.iei.pi.cnr.it/` Email: `{m.callieri | f.ponchio | p.cignoni | r.scopigno@isti.cnr.it`

since users usually possess limited skills in managing 3D graphics technology (e.g. museum curators, art historians, restorers, ordinary visitors of a museum, etc.). The design of the GUI and the overall usability of the rendering tool play a critical role in deciding if the tool will be just a nice toy or a really useful technical instrument. Another problem is that the complexity (the huge number of graphics primitives) of the accurate, realistic-looking models usually exceeds the interactive capabilities of most graphics workstations. Real time visualization of those huge meshes is needed, without sacrificing the high quality achievable with 3D scanning technologies.

This paper presents the design, the architecture and the evalutation of a new CH-oriented visualization system, VIRTUAL INSPECTOR, which is specifically tailored to the interactive and easy inspection of very dense 3D models. Our goals were:

- **Commodity graphics workstation tailoring**: attain maximal performances from inexpensive platforms (PCs with mainstream graphics boards), to ensure the widest user community;
- **User tailoring**: because a major fraction of the user community is composed by naive users, interaction with 3D data should be as easy and natural as possible; usability should be given priority over flexibility and completeness of the rendering features;
- **High quality interactive rendering**: user should be able to interactively browse highly dense 3D models without suffering degradation in performance or quality of the rendering. To fulfill this goal, we adopted a *continuous LOD* approach [8] for the on-line 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 an art historian should find easy to enrich the 3D model of related multimedia information. Therefore, we should go beyond pure geometric data visualization. This can be implemented by adding an hot spot feature (i.e. interactive links to other info associated to single points over the 3D model surface) and making the system able to operate with standard web browsers;
- **Local and remote access**: the system should be able to run both locally (e.g. driving a multimedia kiosk installed in a museum) and remotely (allowing a web access to huge 3D models by mean of a *thin* client able to run even over obsolete hardware);
- **Data protection over the web**: high-quality 3D model cost some effort to be scanned and could be considered sensible data; therefore, in some cases we need to protect the 3D data and to prevent its uncontrolled distribution.

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

2 Virtual Inspector

Virtual Inspector has been designed to give a solution to the issues listed in the previous section, taking also into account the experiences learned with our first CH oriented visualization tool [2]. We present here briefly the architecture and its main features of our tool.

The system architecture has been designed by choosing a triangle-based approach to 3D data management. To support interactive presentation of massive models, Virtual Inspector adopts an out-of-core multiresolution approach where view-dependent variable resolution representations[1] are extracted on the fly using a state-of-the-art highly efficient approach [4]. For each frame, the best-fit *variable resolution* LOD is selected according to the current view frustum and the requested visualization accuracy. LOD selection and rendering are very efficient since we adopt a GPU-friendly representation, where a coarse-grain patch-based multiresolution hierarchy is visited on the fly and ready-to-render geometry patches are associated to each logical node of the variable LOD produced. 3D data are therefore not processed at runtime at the grain of single triangles, but triangle chunks are efficiently fetched from disk on demand and

---

[1] Note that the first version of Virtual Inspector was based on a discrete LOD approach.

3D Scanning Technical issues Many works concern the use of 3D technology either to reconstruct digital 3D models of Cultural Heritage masterpieces or to present those models through digital media. An exhaustive description of those works goes well beyond the brief overview that we can draw in this section. We prefer to cite here only some seminal papers about the technologies proposed for building those models (mostly 3D scanning) and for supporting their interactive visualization.

Automatic 3D reconstruction technologies [A.3] have evolved significantly in the last years. Unfortunately, most 3D scanning systems do not produce a final, complete 3D model but a large collection of raw data (*range maps*) which have to be post-processed. The post-processing pipeline is presented in the excellent overview paper by Bernardini and Rushmeier [A.2]. Solutions to some algorithmic sub-tasks improved a lot since this review paper, and the most notable progress lies in the areas of: semi-automatic alignment (a recent example is [A.4]); surface reconstruction from samples, with many new solutions based on pointset representations (see [A.5, A.1]); and automatic reconstruction from uncalibrated sequences of high resolution photographic images [A.7]. All these software and hardware improvements make 3D scanning a viable, sufficiently fast and affordable modelling option for CH purposes.

Many of these projects considered data management issues, in particular how to process and render at interactive frame rates the high resolution meshes produced with 3D scanning. Several techniques have been developed to cope with this problem: some of them keep a triangle-based representation and adopt *geometry simplification* and *multiresolution representation* [8] to reduce data complexity and rendering times; others adopt a *point-based rendering* approach coupled with keen heuristic for dynamic data sub-sampling [A.6]. We should note that the majority of the previous papers focus only on the problem of efficient rendering, rather than presenting a complete data visualization tool. The task of a person in charge of setting up a museum kiosk or a virtual exposition is still very complex: we have a number of well designed algorithmic solutions coming from the academia, but most of them require significant effort to be used in a standard application domain and are usually limited to the pure rendering of the 3D data; on the other hand, we have a small number of commercial tools able to setup very pleasant multimedia presentations that in some cases offer also sophisticated rendering options for 3D models, such as Cult3d, Macromedia Director or TurnTool, just to cite a few. But those commercial tools are designed to work with standard small/medium complexity 3D models and are not able to support the efficient rendering of the very dense 3D scanned meshes. When using these tools the only viable option is to simplify the original datasets in a very aggressive manner discarding in this way most of the detail and accuracy of the sampled digital model.

A.1 N. Amenta and Y. J. Kil, Defining point-set surfaces. *ACM Transactions on Graphics*, 23(3):264–270, 2004.

A.2 F. Bernardini and H. E. Rushmeier. The 3D Model Acquisition Pipeline. *Computer Graphics Forum*, 21(2):149–172, 2002.

A.3 B. Curless and S. Seitz. 3D Photography. In *ACM SIGGRAPH'00, Course Notes No. 19*, 2000.

A.4 A. Fasano, P. Pingi, P. Cignoni, C. Montani, and R. Scopigno. Automatic registration of range maps. *Computer Graphics Forum*, 24(3):517-526, 2005.

A.5 Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. P. Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, 2003.

A.6 S. Rusinkiewicz and M. Levoy, QSplat: A Multiresolution Point Rendering System for Large Meshes. *ACM Transactions on Graphics*, 10(3):343–352, 2004.

A.7 Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision*, 59(1):61–85, 2004.

3D Scanning and Cultural heritages. Many significant projects concerning 3D scanning and cultural heritage have been presented in the last few years. We list here just a short but representative selection.

- B.1 M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D scanning of large statues. In *SIGGRAPH 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 131–144. Addison Wesley, July 24-28 2000.
- B.2 F. Bernardini, H. E. Rushmeier, I.M. Martin, J. Mittleman, and G. Taubin. Building a Digital Model of Michelangelo's Florentine Pieta'. *IEEE Comp. Graphics & Applications*, 22(1):59–67, Jan-Febr. 2002.
- B.3 J. Stumpfel, C. Tchou, T. Hawkins, P. Debevec, J. Cohen, A. Jones, and B. Emerson. Assembling the sculptures of the parthenon. In A. Chalmers D. Arnold and F. Niccolucci, editors, *VAST 2003*, pages 41–50, Bighton, UK, Nov. 5-7 2003. Eurographics.
- B.4 M. Pollefeys, L. J. Van Gool, M. Vergauwen, F. Verbiest, and J. Tops. Image-based 3d acquisition of archeological heritage and applications. In D. Arnold, A. Chalmers, and D. Fellner, editors, *VAST 2001 Conference Proc.*, pages 255–261, Athens, Greece, Nov. 28-30 2001. ACM Siggraph.
- B.5 M. Callieri, P. Cignoni, F. Ganovelli, G. Impoco, C. Montani, P. Pingi, F. Ponchio, R. Scopigno. Visualization and 3D data processing in David restoration. *IEEE Computer Graphics & Applications 24*, 2 (Mar.-Apr. 2004), pages 16–21.

copied on GPU memory with maximal rendering efficiency.

VIRTUAL INSPECTOR is mainly designed for the visualization of single works of art (sculptures, pottery, architectures, etc.), and adopts a very intuitive approach to guide the virtual manipulation and inspection of the digital replica, based on a straightforward metaphor: we provide a *dummy* representation of the current inspected model on a side of the screen, which can be rotated on its axe. To select any given view the user has just to point with the mouse the corresponding point on the *dummy* (see Figure 2).

VIRTUAL INSPECTOR supports also the specification of *3D links and popups* and smoothly inter-operates with a web browser. Hot spots on 3D models are a very handy resource to associate multimedia data (e.g. html pages and videos) to selected locations of a 3D model. This allows to design interactive presentations where the 3D model acts as a natural visual index to historical/artistic information, presented using standard HTML formats and browsers.

Finally, a very important characteristic of VIRTUAL INSPECTOR is its flexibility and configurability. The whole interface and its behavior can be easily specified by the authoring user via XML coding: which are the 3D models to be rendered (a single mesh or multiple ones), the system layout characteristics (i.e. how the different models and GUI components will be presented on the screen), the rendering modes (e.g. standard Phong-shading of sampled geometry or BRDF rendering), the interaction mode (e.g. model manipulation via a standard virtual trackball, via the dummy-based "point and click" interaction, or both), links to multimedia external information and many other aspects.

Figure 2: The new gaze point, selected by the user by a simple mouse click on the corresponding location on the dummy, is marked for illustration purposes by a red circle (see the two zoomed image fragments).

3 GUI Design

The standard GUI layout of the Virtual Inspector tool is shown in Figure 1: it represents an effective balance between ease-of-use and freedom of inspection. The output window of the tool is divided in two main frames: the one on the right is dedicated to the interactive selection of the desired view and to the GUI while the left side is devoted to the full display of the inspected 3D model with the chosen view.

A small complete view of the inspected object, called *dummy*, is visualized in the rightmost frame. This *dummy* is conceived as a sort of interactive and intuitive 3D map whose role is to allow an easy selection of the view specs. The dummy is always visualized as a whole, its appearance is not affected by the current view, neither its level of detail (the dummy is visualized using a fixed low-resolution model). The user can see all the sides of the dummy by rotating it (on its vertical axis) with a simple button-driven interface.

The selection of the viewing parameters (for the left-most frame) is implemented according to a very simple direct manipulation approach. A mouse click on any point of the dummy (see Figure 2) make that the left frame looks exactly at that point modifying the viewing parameters as follows: the point selected on the dummy surface becomes the *gaze point*, and the *view direction* is set by default to be equal to the surface normal in the gaze point. The *field of view* is set initially using a default value (an example is shown in Figure 2, where approximately 20% of the object is in the current view volume), or takes the value used/set in the previous interactive actions. The *field of view* value is controlled by clicking on the (zoom-in/zoom-out) GUI buttons (see Figure 1).

To be more specific, the *view direction* is set equal to the *average* surface normal computed over a small mesh portion centered on the gaze point. This to prevent potential erratic or random values of the view direction in locations where the surface normal changes rapidly (e.g. on rough surfaces, or on highly convex or concave regions). For each selected pixel in view space, a 3x3 or 5x5 sampling kernel is evaluated: for each of these pixels, we compute the corresponding point on the digital model, and the best fitting plane to these locations in modeling space gives the *average* surface normal.

For less than average naive users a standard *virtual trackball* is also available in the leftmost frame; the trackball allows to specify, in a more precise way, either rotations or pan actions around the gaze point, supporting a more accurate and versatile object manipulation.

Form the technical point of view, the two frames present two different coordinate systems: the rightmost frame uses a constrained coordinate system that models the space containing the dummy; the leftmost frame uses the coordinate system derived by the current view specs, selected by the user either by clicking on the dummy or by using a *virtual trackball* on the mesh rendered in the leftmost frame.

The design of this interface directly reflects the fact that most of the naive users are used to point&click interfaces, while the interaction tools which are standard in the computer graphics community (e.g. the virtual trackball approach) are not so wide spread among museum visitors. So providing a interface where viewing parameters could be specified by simple clicking was important to ensure easy of use.

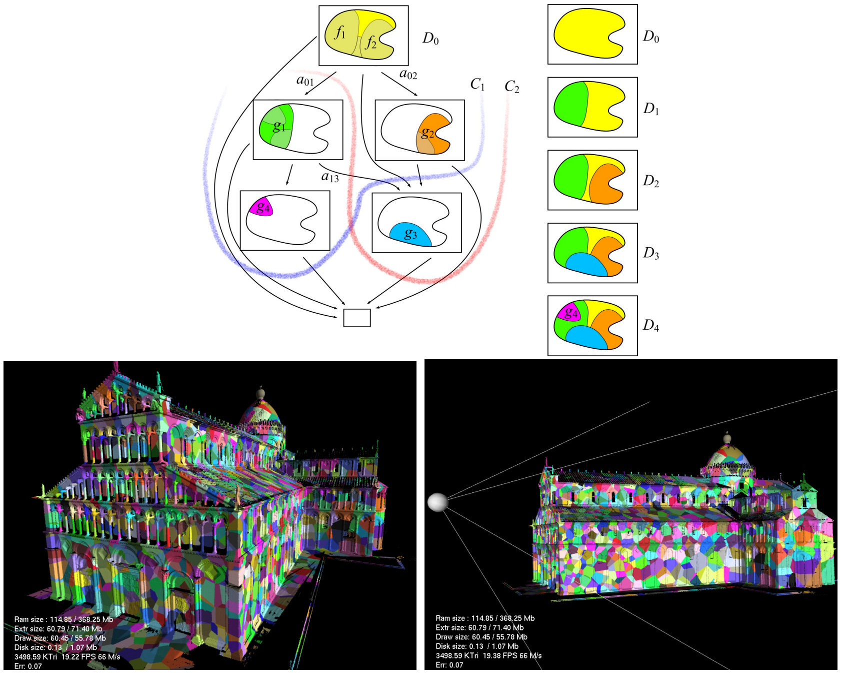
4 Geometric Data Management

To guarantee a satisfying interaction level, the rendering engine of the system has to be able to provide a interactive frame rate without sacrificing the high quality of the models. A lot of different solutions have been proposed in literature for the efficient visualization of large, complex digital 3D models. In the design of such a rendering system we must face, among the others, the following issues:

- **Choosing the right resolution**: 3D scanned models at full resolution are often composed by a number of rendering primitives much larger than the number of pixels covered by a standard view. Using a very dense representation (more triangles than pixels) is usually not the wisest choice, since the visual improvements vs. using a more compact model is usually lower than the perceptible threshold;

- **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 to be rendered can be composed by tens or even hundreds of millions triangles and



```
Ram size : 114.85 / 368.25 Mb
Extr size: 60.79 / 71.40 Mb
Draw size: 60.45 / 55.78 Mb
Disk size: 0.13  / 1.07 Mb
3498.59 KTri  19.22 FPS 66 M/s
Err: 0.07
```

```
Ram size : 114.85 / 368.25 Mb
Extr size: 60.79 / 71.40 Mb
Draw size: 60.45 / 55.78 Mb
Disk size: 0.13  / 1.07 Mb
3498.59 KTri  19.38 FPS 66 M/s
Err: 0.07
```

Figure 3: (top) An example of the MT DAG that shows the one-to-one correspondence between the valid subsequences and the valid cuts. (bottom) A single cut extracted from the Pisa Cathedral model, where each patch is rendered with a uniform color; the image on the right (same model than the one in the left, the viewpoint used for the multiresolution extraction is indicated by the small sphere) shows how the spatial exthent covered by each patch increases with the distance from the viewpoint.

> would require gigabytes of RAM to be managed at full resolution. Conversely, only the currently viewed portion of the model should be loaded in main memory, possibly in a format which allows maximal rendering performances on modern GPU's.

Until recently, the vast majority of view-dependent LOD methods were based on multiresolution structures that take decisions at the triangle/vertex primitive level. The cons of these approaches is a constant CPU workload for each triangle. On modern GPUs, this approach is doomed to make the CPU the bottleneck of the whole rendering process. To overcome this bottleneck and to fully exploit the capabilities of current graphics hardware it is therefore necessary to select and send batches of geometric primitives to be rendered with just a few CPU instructions.

Virtual Inspector rendering engine adopt a recent innovative solution, a *batched* multiresolution framework [4] based on the GPU reinterpretation of the Multi-Triangulation (MT) [9] approach. The MT is a very general framework that encompasses a wide class of multiresolution algorithms, but, like the vast majority of the techniques proposed in the 90's, it was originally designed to minimize the number of triangles to be rendered, at the expense of CPU time. Therefore, we have redesigned in a GPU-friendly fashion the MT scheme, by moving the granularity of the whole framework from single triangles to optimized surface patches, and by redefining the construction and rendering algorithm to work on external memory (a mandatory approach to manage huge scanned meshes).

Our Batched Multi-Triangulation (BMT) represents the 3D model by building a direct acyclic graph (DAG) where each single element represents a ready-to-render patch of a few thousands triangles (see Figure 3). In this way the DAG size is reduced by some orders of magnitude, with respect to standard solutions, and therefore the per-frame workload is much lower consisting just in assembling at run-time an adequate puzzle of pre-assembled optimized surface patches. By grouping together sets of triangles (and representing them in the most GPU-efficient manner) we are able to alleviate the CPU/GPU bottleneck. Since the granularity is much coarser than the one of a standard MT representation, the CPU workload for multiresolution data structure management is lowered by two or three orders of magnitude.

The patches are also the basic units to arrange the dataset in a out-of-core fashion; a dynamic set of the patches that will be probably needed in further rendering cycles is continuously updated and kept in memory with a non-blocking multi-threaded approach.

Extracting a variable resolution model, i.e. choosing the right set of patches that have to be rendered, means extracting a cut over the above cited DAG (see Figure 3 top), an action which can be performed efficiently when the DAG size is, as in our case, small. 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. For this reason we choose an extraction algorithm that is able to find a cut over the DAG within a predetermined budget of time and memory resources, always ending with a consistent result, or, in other words, it is interruptible. With this approach, eventually sacrificing on the resolution side, we are sure that a interactive frame rate is always sustained.

The extraction and rendering of a dynamic continuous LOD with the BMT scheme have been evaluated over several inspections, rotating and abruptly zooming in and out the model. Those tests were done with different rendering window sizes on a PC equipped with an ADM athlon 64 clocked at 2,21 Ghz, 2 GB RAM, Windows Xp professional 2002 and an NVIDIA 6600GT graphic board.

The results demonstrate that the rendering speed depends mostly on the vertex processing speed of the GPU, rather than on the display resolution used. We are able to render on the selected GPU around 220M vertices per second (on the Cathedral dataset). Some numeric results are presented in Table 1, obtained over the Pisa Cathedral (390M triangles) and the color-mapped double David (112M triangles, see Figure 10) datasets. The accuracy threshold used in the real-time extraction of the variable LOD is measured in terms of screen pixel

Dataset	display size	vertices per frame	fps	max error
Pisa Cathedral	1248x1024	22.2M	10	1.3 pixels
	800x600	22.2M	10	0.8 pixel
	1248x1024	12.3M	18	2. pixels
Double David	1248x1024	11.4M	19	0.9 pixels
	1600x1200	11.4M	19	1.2 pixels

Table 1: Rendering performances of the Batched Multi-Triangulation representation.

units (see the *max error* value in the table); it is an extimate in excess of the actual error (the maximal upper bound of the error), which is usually much larger than the actual error.

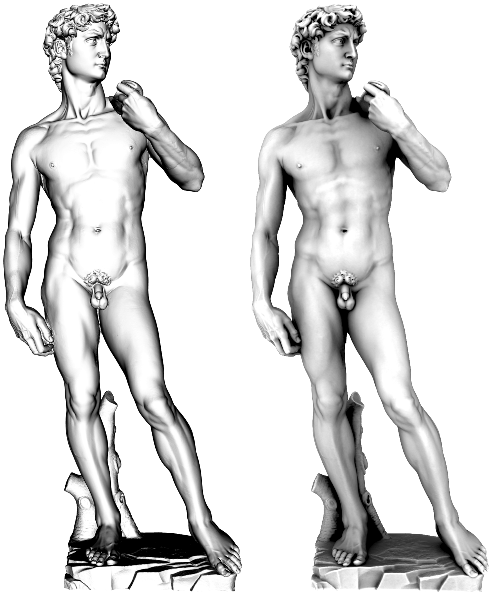


Figure 4: Rendering cast shadows greatly affects the resulting visual presentation of a sculpte surface. On the left a standard OpenGL rendering with Phong lighting, on the right a OpenGL rendering with a pre-computed ambient occlusion lighting.

5 Rendering modes

Virtual Inspector supports the interactive modification of the lighting, to simulate in real time the “luce radente” (grazing light) effect that is usually used in real inspection to enhance the visualization of small-scale surface detail.

In terms of rendering mode, Virtual Inspector supports: *ambient occlusion enhanced* rendering mode, *BRDF-based* rendering and *protected remote rendering* over the Web.

5.1 Ambient Occlusion for enhanced rendering

A common problem in computer graphics is the evaluation of the trade-off between quality and efficiency in the rendering process. In the context of the visualization of models of real statues the standard local lighting model used in interactive graphics is not satisfactory. Standard OpenGL lighting does not take into account the effects of cast shadows and these effects are very important to perceive shape; moreover, artists usually takes into account these lighting effects during the creation of a statue. Figure 4 gives a practical example on the David . The only difference between the left and right OpenGL rendering is in the lighting: on the left a standard Phong direct lighting, on the right we used an approximation of a diffuse illumination lighting where cast shadow are properly computed. The Phong lighting model uses a simple per-scene constant lighting term for all the portions of the scene which are not directly lit, but this approach leads to a notable flatness. This approach has been improved by explicitly computing for each point of the surface its accessibility value, which is the percentage of the hemisphere (the sky) above each surface point not occluded by geometry [6]. This useful technique is commonly known as *ambient occlusion* and it is used in many production environments to add an approximation of the shadowing of diffuse objects lit with environment lighting. We already successfully used this precomputed approach in the previous version of our visualization system [2].

Figure 5 illustrate the basic idea of *ambient occlusion* approach: the face f_1 is darker than face f_2 because it sees a smaller portion of the sky. This per face lighting computation is done statically in a pre-processing phase. The fine tessellation of the mesh guarantees a sufficiently good sampling and rendering of the ambient occlusion term onto the surface. This ambient term is then combined with standard OpenGL direct lighting allowing the user to interactively move the lighting direction to better perceive the shape of the inspected objects. This hybrid approach (static diffuse lighting field plus dynamic head-light) allows us to have a view-dependent lighting that produces shiny reflections which move dynamically over the surface depending on the current view specs.

5.2 BRDF-based rendering

The bi-directional reflectance distribution function (BRDF) describes how light is reflected off the surface of an object. Virtual Inspector has been extended to be able to render 3D models which comes with the specification of their BRDF. More in detail, standard BRDF are used to describe the behaviour of ideal objects whose surface is made of a single homogeneous material. Most objects however consists of several different materials, or of a single corrupted

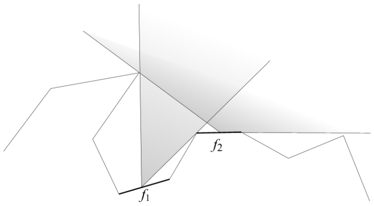


Figure 5: A better approximation of the ambient lighting term can be evaluated by computing the solid angle of the *sky* that can be seen from each face or vertex (larger the solid angle more lighted will be the face).

Figure 6: Virtual Inspector showing the Minerva head, rendered on the basis of the BRDF model sampled from the real statue.

material. This is especially true for works of art or archeological objects (an example of the Minerva head rendered using a sampled BRDF is shown in Figure 6). A very precise way to represent these details is to assign a different BRDF to each surface point which leads to a *spatially varying BRDF*. Without these details, objects tend to look artificial and unrealistic. The spatially varying BRDFs approach included in Virtual Inspector follows the sampling and rendering approach proposed by Lensch et al. [7]. It has been implemented in an efficient manner by coding the parameters of the spatially varying BRDF into textures and exploiting the programmable features of modern programmable GPU to compute in real time the resulting lighting (in collaboration with the MPI colleagues and in the framework of the EU IST “ViHAP3D” project).

5.3 Protected remote rendering

One of our goals is to make VIRTUAL INSPECTOR a tool which could be used on the web as well. Big and accurate data are a problem for web use. Moreover, the data owner could decide that the 3D mesh is valuable and has not to be transmitted to the user.

To support web-based visualization, Virtual Inspector has been extended by adding a remote rendering mode and service. Instead of computing the rendering of the 3D model locally, this task is performed by a dedicated remote server. The local machine does not receive the full resolution model but only a reduced resolution model supporting user interaction. When a viewpoint is selected by the user, the local rendering client sends a request over the net to a rendering server (which is the only one to retain a copy of the full resolution model), the server renders the model according to the view parameters and sends back the resulting image to the client. The philosophy of this client-server interaction follows the one proposed in [5] and presents various benefits:

- The full-resolution 3D model is not disclosed to the final user, protecting it from improper use or data thieving.
- The amount of data necessary on the client is drastically reduced; this is especially useful if data have to be downloaded through the net and for casual user that will probably browse the models for a short time.
- Since the full resolution model is managed and rendered by a dedicated server, it is possible to use a model much bigger than the one that could be normally processed on the local machine (due to transmission and local storage limitations).

The only drawback of this system is the cost of the network communication, in terms of latency and network load, and the need to setup a rendering farm infrastructure adequate to the forecasted access load.

The remote server has been implemented using the same multiresolution technology outlined in the previous section and it can work with models of arbitrary resolution. The remote rendering node is dedicated to the rendering task and this guarantees optimal performance. Moreover, it is possible to apply server-side changes in order to enhance rendering (by augmenting model resolution or perfecting the rendering) without changes to the clients. The user interaction with the 3D model and the application behavior, beside the small latency introduced by the network interaction, remains identical to the one 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+, 2GB RAM, nvidia GeForce 6600 256MB, Ubuntu Linux OS and an internet connection running at 1.2 Mbit. The experiment was based on the use of two datasets: the 3D model of the Pisa Cathedral (time-of-flight scanning, 390M faces) and a setup with two David models with color per vertex (2x50M triangles, see Figure 10), obtaining the following results:

- **Cathedral**: the time for an entire remote image refresh (image request sent to server + remote rendering + compression + transmission + decompression + viz on screen) is in the range 750 - 2000 milliseconds; severside remote rendering only requires 100 - 500 milliseconds, depending on the specific view.
- **David**: entire image refresh in 150 - 650 milliseconds.

In the case of a large community of users, a remote rendering farm composed by multiple rendering servers can be adopted, managed by a renderer dispatcher.

6 XML-coded Interface and Behavior

A basic innovation with respect to the first version of VIRTUAL INSPECTOR is the improved configurability of the system. This has been obtained by designing a simple interpreted language, called *NSP*, that is used to build up the interface, its appearance and its behavior. The language exploits XML for all the well known advantages of this technology (availability of syntax aware parsers, human readability, extendibility, etc.)

At startup, VIRTUAL INSPECTOR reads the *.nsp* file and configure itself opportunely accordingly to the instructions there specified. Therefore, the designer of the multimedia application does not have to compile a new version of the system to obtain a new layout, but he/she can simply design and distribute a new GUI layout by simply editing a new *.nsp* file.

The XML approach allows to change very easily the look and feel of VIRTUAL INSPECTOR. As an example, compare the different layout and graphics of intermediate versions (Figure 6) with the one of the more professional Arrigo VII's installation (Figures 1), where a professional graphic designer has redesigned the layout of the application, including all icons and background graphics elements. This has been done by the easy specification in the XML initialization file of: the new background and icon images; the location on the screen of all icons and elements of the GUI. Neither programming nor recompilations of VIRTUAL INSPECTOR was needed. It is a task that can be easily assigned to an operator with standard web-design competence. The *NSP* allows also to customize the behavior of the various elements of the GUI, changing the approach used to browse the model, imposing constraints and defining new mechanism for navigating the objects. An example of a different application layout that show the high degree of configurability of the system, is the double Minerva installation, where two models of the Minerva statue taken at different restoration progress are rendered in a coordinated/syncronized manner (Figure 7).

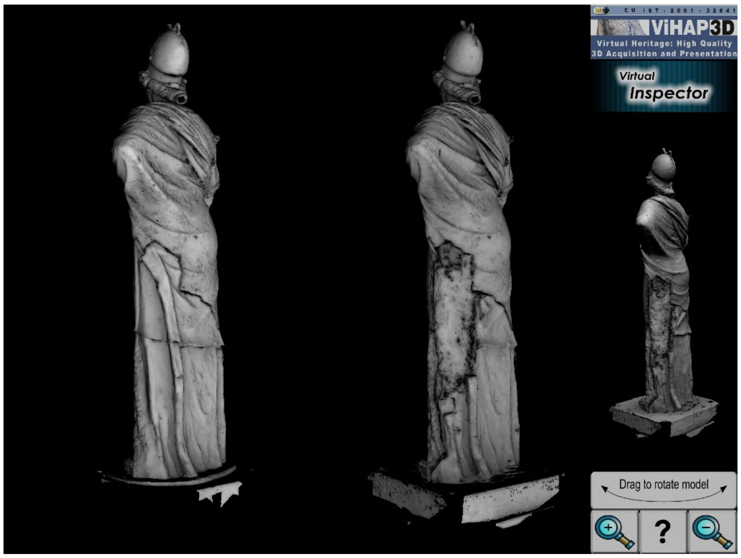
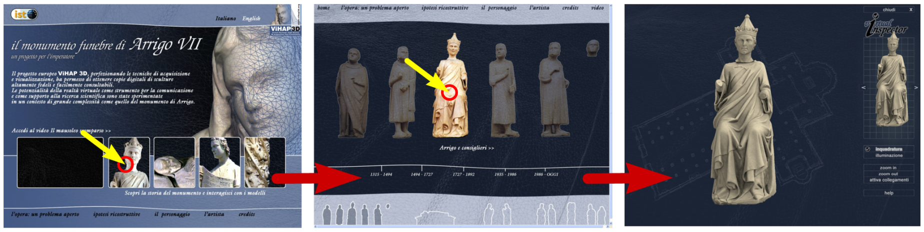


Figure 7: Virtual Inspector showing two models of the Minerva statue, scanned at different times during restoration.

6.1 Interoperation with a web browser and other multimedia data

Like the rest of the interface, the specification of *hot spots* is encoded in the *.nsp* specification of the current Virtual Inspector instance, and modifications to the 3D model are not required. We provide a simple 3D browser to the person in charge of the implementation of the multimedia presentation, which allows to load a 3D model and to query the 3D coordinates of any point on the surface of the artifact (by simply clicking with the mouse on the corresponding point), generating the xml text portion to be included in the *.nsp* file. Then, a new hot spot is specified by adding that XML tag in the Virtual Inspector specification file. The hot spot XML tag specifies basically the 3D location and the action that has to be triggered when clicking on the hot spot (e.g. the name of the html file, if we want to open a multimedia page). After activation, the control passes to the html browser, while Virtual Inspector remains sleeping in the background and regains automatically the control of the interaction whenever the html page is closed.

A museum installation can be organized with introductory HTML pages, which present some general artistic/historic information on the work of art. Some of these may provide links to activate Virtual Inspector on a single or multiple artifacts (see in Figure 8 an example concerning the Arrigo VII installation [1], which presents a group of 13 statues).



il monumento funebre di Arrigo VII

un progetto per l'imperatore

Il progetto europeo ViHAP 3D, perfezionando le tecniche di acquisizione e visualizzazione, ha permesso di ottenere copie digitali di sculture altamente fedeli e facilmente consultabili.
Le potenzialità della realtà virtuale come strumento per la comunicazione e come supporto alla ricerca scientifica sono state sperimentate in un contesto di grande complessità come quello del monumento di Arrigo.

Figure 8: The initial screen of the Arrigos VII's multimedia kiosk and one of the following sub-index pages are shown above. To provide access to any statue of the Arrigo VII complex, the statues have been divided in four groups (the second image shows the index page related to the "Arrigo VII enthroned" and counsellors' group). Virtual Inspector can be started by clicking on any of the icons of the statues presented (middle image). This starts a new session of Virtual Inspector (see rightmost image).

7 Evaluation and use of the Virtual Inspector tool

The Virtual Inspector tool was originally conceived in the framework of a cooperation with the Restoration Laboratory of the Tuscan Archeological Superintendency (Florence, Italy) aimed at the restoration of the *Minerva of Arezzo* statue. We scanned the Minerva four times, at different stages of the restoration, starting in October, 2000 [10] and produced models up to 65 M triangles each (see Figure 7). We planned different uses of the 3D model: to monitor the initial and the post-restoration status of the artifact (since a major modification of shape and appearance was forecasted); to be used to map and correlate the results of different investigation survey (photo under visible and UV light, X-ray imaging, results of chemical analysis, etc.); to be part of a multi-media archive which should document the complete restoration; to be used for visual presentation (museum installations or multimedia CDs). Therefore, it was clearly needed an easy-to-use tool which should allow the restorers to access the accurate and high resolution digital 3D model. One basic requirement was obviously to have interactive access to the high resolution model without any compromise on accuracy and detail. The first evaluations of the Virtual Inspector tool (2001-2002) were very encouraging, fulfilling the above requirements. More specifically, the restorers were fascinated by the possibility to inspect so easily such a dense and accurate model; they considered extremely useful the possibility to easily compare different digital models referring to different time frames (see Figure 7). An interactive 3D visualizer can give a fundamental contribution to the inspection and analysis of an artwork and can substantially improve the knowledge of restorers, art historians and, obviously, of the ordinary public. The didactic power of the Virtual Inspector system is valuable: the capability to navigate and inspect a statue from any viewpoint at the extreme accuracy

*Le braccia della statua,
oggi ridotte a moncherini, dovevano
essere disposte in modo da attenuare
l'andamento diagonale delle pieghe sul
petto mentre le mani dovevano recare le
"insignia imperalia", verosimilmente
in metallo e dunque probabile causa
delle prime mutilazioni.
Gli oggetti-simbolo del potere quali
lo scettro e il globo, accompagnati alla
corona, caratterizzano infatti
l'iconografia regale e imperiale.*

La testa dell'imperatore è spezzata rispetto al resto del corpo e risulta male impostata sul collo sottostante. E' stata notata una certa sproporzione rispetto al resto della figura ed una distanza stilistica tra la fattura minuta del volto in confronto al resto del corpo e alle immagini dei "consiglieri"; si peraltro è ipotizzato per la testa l'uso di un materiale non omogeneo, fino a mettere in dubbio la pertinenza alla statua o che le singole parti siano da attribuire a mani diverse. Non sembra tuttavia probabile che la testa fosse destinata ad altra figura perché al livello della parte superiore del petto e sulle spalle il marmo è stato scalpellato e dunque l'andamento irregolare della frattura non offre una prova sufficiente. Anche la presunta differenza di materiale richiederebbe di essere verificata attraverso l'analisi mineralogico-petrografica: infatti la venatura grigia è generalmente discontinua, più o meno diffusa in zone diverse dello stesso blocco.

L'affinità con la testa di Arrigo giacente in Cattedrale -seppure non unanimamente riconosciuta- sembra confermare l'appartenenza della statua in trono al monumento funebre. Il problema attende necessari chiarimenti: proprio sulla testa coronata si basa di fatto l'identificazione di tutto il gruppo con l'imperatore e il suo seguito, nonché la tradizionale attribuzione alla tomba di Arrigo.

Figure 9: Virtual Inspector: the "Arrigo VII enthroned" statue rendered with active hot spots (top); a short popup panel with a short info, describing the missing hand, appears when the mouse passes over the hotspot (middle); an example of an HTML page activated by clicking the hot spot on the neck (bottom).

provided by the range scanned model allows new insight capabilities.

An example is the new installation reporting the condition of the Michelangelo's David statue before and after the restoration (2002-2004). This restoration concerned mainly a cleaning of the surface, e.g. aimed at the removal of brown crusts and spots, and the replacement of plaster filling of small holes and marble cracks [3]. Two complete photographic campaigns (digital photographs depicting the status before and after the restoration, each one composed of around seventy 5M pixels images) have been mapped to the surface of the David (using a per-vertex color mapping approach over the 56M triangle model). This installation allows an easy analysis of the effects of this recent restoration (see Figure 10).

If we focus on art teaching, 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 cannot give the same amount of information of an interactive navigation of a 3D model, and it has to be considered that the availability of a complete photographic sampling is not the standard situation in didactic applications: in general, a student has just a few images depicting a given artifact. Moreover, the insight capability of the Virtual Inspector tool

Figure 10: Virtual Inspector showing two models of the David statue, in pre- and post-restoration status.

could be more informative than a personal inspection of the real artwork. For security reasons, valuable artifacts cannot be manipulated or can be so large in size that detailed inspection is impossible. The possibility to inspect from a scaffolding a given sculpture masterpiece or an architecture is a privileged access that just a few art historians have once in their life. A similar experience can be replicated virtually for ordinary people using modern 3D graphics tools.

Virtual Inspector has been used proficiently in several museum exhibits and fairs in the last couple of years. We experimented that in most cases the standard fearful attitude of non-technical people towards 3D graphics shifted in a few seconds to interest, interaction and amusement. Concerning the Arrigo VII installation, on stage at the Museum of Pisa's Cathedral since October 2004 (see Figure 11), we have run a specific quality and usability assessment. This evaluation was commissioned to an independent consultant, an external company expert in the evaluation of museum and expositions and in users satisfaction assessment. They analyzed the system performances by sampling the museum visitors reactions with a questionnaire, some direct interviews and the analysis of the recording (via an automatic video grabbing facility) of the performance of the visitors with the multimedia kiosk [11]. The results were extremely positive, since a very large majority of the users were able to proficiently use the tool and would like to see this experiment replicated to many other artworks shown in the museum (85% of the people who used the kiosk). A subset of the results produced after processing the 330 questionnaire compiled by museum visitors are presented in Table 12, while the complete assessment results are

Figure 11: The multimedia kiosk developed for the Museo of the Catedral at Pisa.

presented in [11].

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

Virtual Inspector has been also used in a temporary exposition at the Ferrara 2006 Restoration Fair (Italy) to show to the public the results of the acquisition of the digital model of the Cathedral of Pisa, a huge architectural model obtained with time-of-flight scanning and totalling 390M triangles.

8 Conclusions

We have presented the Virtual Inspector tool, an interactive system for the visualization of complex and accurate digital 3D models. Virtual Inspector is mainly oriented to the visualization of single works of art and adopts a very intuitive point-and-click approach to guide the virtual manipulation and inspection of the digital replica. It was designed according to the specification and needs of both restorers and art curators, and qualifies as a very handy tool for discovering the beauty and complexity of works of art. From a computer graphics point of view, the innovation degree of this paper does not rely in a single

Virtual Inspector usability results					
kiosk: easy to use?	very easy 26,3%	easy 68,7%	complex 1,7%	very complex 0,6%	no opinion 2,8%
kiosk: is it more informative than panels?	yes 84,4%	no 15,6%			
3D models: did you use them?	yes 71,5%	no 28,5%			
3D models: are really useful?	yes 92,1%	no 7,9%			% of those using 3D (71,5% of the total)
3D models: are easy to use?	yes 97,4%	no 2,6%			w.r.t. ones 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 ones 60,3%	no 5,2%	no opinion 9,2%	

Figure 12: Usability evaluation of the multimedia kiosk telling the story of the Arrigo VII's mausoleum and implemented using Virtual Inspector (numerical results are percentages of the 330 questionnaires returned).

new technique or algorithm, but rather in the design of a complex visualization tool based on state-of-the-art technologies. To our knowledge, Virtual Inspector is the only visualization tool specifically designed for CH applications which fulfills the specifications listed in the introduction section.

The Virtual Inspector system can be obviously improved, adding new functionality or increasing its efficiency. The visualization of a digital 3D replica is often not considered "the ultimate goal" by restorers. Once it is possible to visualize an artifact with that accuracy, the need of mapping other data on the 3D model rises immediately. We are now working on an slightly extended version of the tool, to support: mapping and selective visualization of different sources of data on the surface (such as any type of 2D image); tools to compute measures (e.g. computing point-to-point lengths); more dynamic enrichment of the data linked to the mesh, by giving to the users the possibility to add annotations or to link multimedia material to selected points of the artifact surface, with the goal of transforming Virtual Inspector into a more dynamic instrument for CH research and for restoration.

Acknowledgements We would like to thank: the Stanford Computer Graphics Group and Marc Levoy; the acquisition group at MPI Saarbrucken (H. Lensch, M. Goesele, H.P. Seidel); our partners in the CH domain: Franca Falletti (Director of the Galleria dell'Accademia Museum, Florence), Clara Baracchini (CH Superintendency in Pisa), and the archaeologists and restorers of the Archeological Restoration Laboratory in Florence.
We acknowledge the financial support of the projects EU IST-2001-32641 "ViHAP3D", EU NoE IST-2002-507382 "EPOCH" and the CNR Project "Fruizione

- Tecniche di supporto e modalità innovative".

References

[1] C. Baracchini, A. Brogi, M. Callieri, L. Capitani, P. Cignoni, A. Fasano, C. Montani, C. Nenci, R. P. Novello, P. Pingi, F. Ponchio, and R. Scopigno. Digital reconstruction of the Arrigo VII funerary complex. In N. Silberman Y. Chrysanthou, K. Cain and F. Niccolucci, editors, *VAST 2004*, pages 145–154, Bruxelles, BE, Dec. 7-10 2004. Eurographics.

[2] R. Borgo, P. Cignoni, and R. Scopigno. An easy to use visualization system for huge cultural heritage meshes. In D. Arnold, A. Chalmers, and D. Fellner, editors, *VAST 2001 Conference Proc.*, pages 121–130, Athens, Greece, Nov. 28-30 2001. ACM Siggraph.

[3] S. Bracci, F. Falletti, M. Matteini, and R. Scopigno, editors. *Exploring David: Diagnostic Tests and State of Conservation.* Giunti Press, 2004.

[4] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, and R. Scopigno. Batched multi triangulation. In *IEEE Visualization 2005*, pages 27–35, 2005.

[5] D. Koller, M. Turitzin, M. Levoy, M. Tarini, G. Croccia, P. Cignoni, and R. Scopigno. Protected interactive 3D graphics via remote rendering. *ACM Trans. Graph*, 23(3):695–703, 2004.

[6] Hayden Landis. Production ready global illumination. In *Siggraph 2002 Course Notes:*, pages 331–338, 2002.

[7] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transaction on Graphics*, 22(2):234–257, April 2003.

[8] D. Luebke, M. Reddy, J.D. Cohen, and A. Varshney. *Level of Detail for 3D Graphics.* The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann, 2002.

[9] E. Puppo. Variable resolution terrain surfaces. In *Proceedings Eight Canadian Conference on Computational Geometry, Ottawa, Canada*, pages 202–210, August 12-15 1996.

[10] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, R. Scopigno, R. Fontana, L. Pezzati, M. Cygielman, R. Giachetti, and G. Gori. 3D scanning the Minerva of Arezzo. In *ICHIM'2001 Conf. Proc., Vol.2*, pages 265–272. Politecnico di Milano, 2001.

[11] Ludovico Solima. Valutazione del caso di studio "Il monumento funebre di Arrigo VII" (in italian). Technical report, Megaride s.r.l., February 2005.