Eurographics Italian Chapter Conference (2006)
G. Gallo and S. Battiato and F. Stanco (Editors)

Easy Access to Huge 3D Models of Works of Art

M. Callieri, F. Ponchio, P. Cignoni, R. Scopigno

Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche†

Abstract

Automatic shape acquisition technologies evolved rapidly in recent years, and huge mass of 3D data can be easily produced. The high accuracy of range scanning technology makes the Cultural Heritage domain one of the ideal fields of use of these devices. Given this particular application domain, two issues arise: how to visualize at interactive rates these complex data on commodity computers (both locally and on web), and how to improve the ease of use of the visualization tools (as potential users are often not expert with interactive graphics).
We present a new visualization system designed to support easy implementation of multimedia kiosk for museums or expositions, which has also been extended to web-based usage. The system allows naive users to inspect a large complex 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 by adopting a continuous *level-of-detail (LOD) representation, where on-line automatic selection of the best-fit level of detail (according to the current view frustum) is coupled with visibility culling and ready-to-render representation of the geometry.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Picture/Image Generation]: Digitizing and scanning.

1. Introduction

Interactive rendering of complex 3D models is crucial for many applications, such as architectural design, graphics simulation and scientific visualization. These systems need to provide the user with a realistic/accurate visual representation of the scene and real-time navigation/interaction. Cultural Heritage (CH) is a specific field of application of 3D graphics, where a basic issue in the design of visualization tools is ease of use: forecasted users usually possess limited skills in managing 3D graphics technology (museum curators, art historians, restorers, ordinary visitors of a museum, etc.). The design of the GUI and its overall usability 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 accurate, realistic-looking models that we are dealing with usually contain much more graphics primitives than the interactive capabilities of most graphics workstations, since our focus is how to process and interactively visualize the huge meshes which are produced with 3D scanning technologies [BR02].

The work described in this paper is two-fold: first, we present the interface of a new visualization tool, specifically oriented to the inspection of complex 3D representations of works of art, enriched by interactive links to standard descriptive multimedia information. Second, we have designed its internal architecture by taking into account two contrasting constrains: the system should support huge meshes inspection, and real-time behaviour should be ensured for both local or remote access to the 3D data. Therefore, our goals were:

- **Graphics workstation tailoring and efficient rendering**: gain maximal performances from inexpensive platforms (PCs with state of the art graphics boards), to ensure universal user community; support interactive frame

† Area della Ricerca CNR, Via Moruzzi 1, 56126 Pisa, ITALY. WWW: http://vcg.isti.cnr.it/ Email: {m.callieri | f.ponchio | p.cignoni | r.scopigno@isti.cnr.it}

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

rates even on huge models, by the adoption of efficient multiresolution solutions;

- **User tailoring**: easy of use and usability should be given priority over flexibility and completeness;
- **Integration of multimedia data**: it should support the addition of links to other MM information (hot spots over the 3D model);
- **Local and Remote Access**: the system should be able to run both locally (e.g. acting as the 3D graphics component of a multimedia kiosk installed in a museum) and remotely (providing access to the data from the web);
- **Data Protection over the WEB**: high-quality 3D model cost some effort to be scanned and could be considered sensible data; therefore, 3D data should often be protected and not distributed freely.

2. Previous Work

Many previous 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 on the technologies proposed for 3D scanning and interactive visualization.

Automatic 3D reconstruction technologies have evolved significantly in the last few years [CS00]. 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 [BR02] and some algorithmic sub-tasks have been improved since this review paper (e.g. surface reconstruction from samples).

Many significant projects concerning 3D scanning and cultural heritage have been presented in the last few years (let us cite just the seminal work [LPC*00]). Most of them considered data management issues, e.g. how to render at interactive frame rates the high resolution meshes produced with 3D scanning. Efficient data management and rendering is a central issue in processing huge dataset. 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* [GH97, CMRS03] to reduce data complexity and rendering times; others adopt a *point-based rendering* approach coupled with keen heuristic for dynamic data sub-sampling [RL00, PZvBG00, BWK02, GM04].

3. Virtual Inspector

The VIRTUAL INSPECTOR browser has been designed to give a solution to the issues introduced in Section 1. VIRTUAL INSPECTOR evolved considerably from the preliminary version presented in [BCS01]. This section presents briefly its architecture and main features (described more in detail in the following sections).

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 a multiresolution approach where view-dependent variable resolution representations are extracted on the fly using a new and highly efficient approach [CGG*05]. 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 patch-based representation, where a coarse-grain 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 reconstructed on the fly at the single triangle grain, but triangle chunks are efficiently fetched from disk on demand and copied on GPU memory for maximal rendering efficiency.

VIRTUAL INSPECTOR is mainly oriented to 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 (see Figure 2).

Other important characteristics of VIRTUAL INSPECTOR are its flexibility and configurability. All main parameters of the system can be easily specified via XML tags contained in a initialization file, such as: 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-shaded per-vertex colors or BRDF rendering) and the interaction mode (e.g. model manipulation via a standard virtual trackball, via the dummy-based “point and click" interaction, or both).

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).*

Finally, VIRTUAL INSPECTOR supports the specification of *hot-spots*. Hot spots are a very handy resource to associate multimedia data (e.g. html pages) to any point or region of a 3D model. This allows to design interactive presentations where the 3D model is also a natural visual index to historical/artistic information, presented using standard HTML format and browsers.

4. Geometric Data Management

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* with a dynamic data extraction criterion; *culling unnecessary geometry*, since only the *visible* geometry should be rendered; *keeping in memory only the currently viewed portion of the model*, 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 taking decisions at the triangle/vertex primitive level. The cons is a constant CPU workload for each triangle that with current GPU evolution makes 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 is based on a new solution [CGG*05]: a *batched* multiresolution framework based on the Multi-Triangulation (MT) [Pup96]. The MT is a very general framework that encompasses a wide class of multiresolution algorithms, but, like 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 from triangles to optimized triangle 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 DAG, where each single nodes represent a batch of triangles. The per-frame workload is reduced by reconstructing at run-time an adequate puzzle of pre-assembled optimized surface patches, making it possible to employ the retained-mode rendering model instead of the less efficient direct rendering approach. The basic idea is grouping together sets of triangles (and representing them in the more GPU-efficient manner) in order to alleviate the CPU/GPU bottleneck. Since the granularity is much smaller than the one of a standard MT representation, the processor workload for multiresolution data structure management is very small and we can keep the data in a form that has a small memory footprint.

Extracting a variable resolution model means extracting a cut over the DAG, action which can be performed efficiently when the DAG size is 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. Our reconstruction algorithm selects a new 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.

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. All the tests were done with rendering window size 800x600 on a Windows machine equipped with an AMD Athlon 64, 2 GHz, 512 MB Ram, SCSI hard disk, bus AGP 8x and graphics card GeForce 6800 GT [CGG*05]. The sustained rendering rate is around 4M triangles per frame at 35 fps with less than one pixel error.

5. GUI Design

The layout of the VIRTUAL INSPECTOR tool is shown in Figure 1. The major choices of the GUI design remained nearly unchanged from the original first version of the system [BCS01]. 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; the main visual output region is the one on the left. The object portion visible at run time in the leftmost frame usually depends on the viewing parameters that the user selects by means of the rightmost window region.

A complete model of the inspected object, called *dummy*, is visualized in the rightmost frame. It is conceived as a sort of interactive and intuitive 3D map whose role is to allow an easy selection of the view specs. The user disposes of constrained rotation of the dummy (on its vertical axis), to see it from any side view. The selection of the viewing parameters is implemented according to a very simple direct manipulation approach. A mouse click on any point of the dummy (see Figure 2) updates the current view 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. It can be updated (zooming-in/zooming-out) clicking on the corresponding GUI buttons (see Figure 2).

6. Rendering modes

In terms of rendering mode, VIRTUAL INSPECTOR supports: **enhanced Phong-based** rendering mode, **BRDF-based** rendering and **protected remote rendering** over the Web.

6.1. Enhanced local 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 lighting model used in interactive graphics is not satisfactory, since the effects of cast shadows are not taken into account. Figure 3 show the major change when considering cast shadows or not.

Figure 3: *Rendering cast shadows greatly affects the resulting expression of the David's face. On the left a standard OpenGL rendering with Phong lighting, on the right a OpenGL rendering with a pre-computed diffuse lighting.*

Current graphics hardware render self-cast shadows with multiple rendering passes, but with a sensible performance penalty. Moreover even with those approaches it is hard to render soft lighting environments where lights are not point-sized and shadows are not very sharp. For the above reasons and for the sake of efficiency we have chosen to partially fix the lighting environment and to compute off-line cast shadows and diffuse shading over the surface; the resulting data is a vector field evaluated on the mesh and stored as a per-vertex color. The fine tesselation of the mesh guarantees a sufficiently good sampling and rendering of the shadow map onto the surface. Even if we use statically cast shadows, the visual results can be made less static by adding one *headlight* to the lighting environment (a common metaphor for a dynamic light source: move always the light together with the observer during the dynamic object inspection). Therefore, we have chosen an hybrid approach: we use a headlight, but we substitute the standard constant *ambient lighting* term with the cast shadows field. The latter approximates the exact lighting that each face of the mesh would receive from an anisotropic diffuse lighting environment. This diffuse lighting can be evaluated in an approximate manner by computing the solid angle of the *sky* that can be seen from each face. Larger the solid angle more light will flow onto the face.

6.2. 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. This is especially true

Figure 4: VIRTUAL INSPECTOR *showing the Minerva head, rendered on the basis of the BRDF model sampled from the real statue.*

for works of art or archeological objects (an example of the Minerva head rendered using a sampled BRDF is shown in Figure 4). 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. [LKG*03]. It has been implemented, in collaboration with MPI colleagues, in an efficient manner by exploiting the programmable features of modern programmable GPU.

6.3. Protected remote rendering

Since managing big and accurate data is a problem on the web, we designed VIRTUAL INSPECTOR to be used on the web as well. Moreover, a protected rendering approach is also useful if 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 a corresponding dedicated rendering service module. The local machine does not receive the full resolution model but only a reduced resolution model for user interaction; when a viewpoint is selected by the user, the local rendering client sends a request over the net to a rendering server (that has 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. Therefore, we follow the client-server approach presented in [KTL*04], which presents several benefits: low ram footprint on the remote client; no limitation on 3D model resolution; data protection. The only drawback of this approach is the cost of the network communication, in terms of latency and network load.

The remote server has been implemented using the same multiresolution technology outlined in Sect. 4 and therefore can manage models of arbitrary complexity. Moreover, it is possible to apply server-side changes in order to enhance rendering (by augmenting the model resolution or using more sophisticated rendering algorithms) without having to change the client. A single server can satisfy multiple clients and deal with different high resolution models; or the remote rendering server can be implemented with a rendering farm (multiple servers managed by a renderer dispatcher). The user interaction with the 3D model and the the application behavior, beside the latency introduced by the network interaction, remains identical to the one of the local rendering mode.

7. XML-based specification of input, GUI layout and rendering modes

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 with 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. With this approach, the designer of the multimedia application does not have to compile a new version of the system to obtain a new type of layout, but he can simply design a new GUI layout by simply typing a new *.nsp* file. The elements of the *.nsp* file are divided into two classes, declarations and framework elements. Declarations are a sequence of XML elements that declares all the 3D entities (models, viewers, lights) that will be displayed, while the framework is the tag which declares the current screen configuration (e.g. resolution used or other parameters).

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 (see Figures 4 and 5) with the one of the more professional Arrigo VII's installation (Figure 7), where a professional graphic designer has redesigned the layout of the application, all icons and the background graphics elements. This has been done by the easy specification of the new images and location on the screen of all icons and elements of the GUI in the XML initialization file and did not required neither programming nor recompilations. It is a task that can be easily assigned to an operator with very limited IT competence.

8. Interoperation with a web browser and other multimedia data

The specification of hot spots is extremely easy in VIRTUAL INSPECTOR, since modifications to the 3D models are not required. We provide a simple 3D browser to the person in charge of the implementation of the multimedia presentation, which allows to query the 3D coordinates of any point on the surface of the artifact (by simply clicking with the

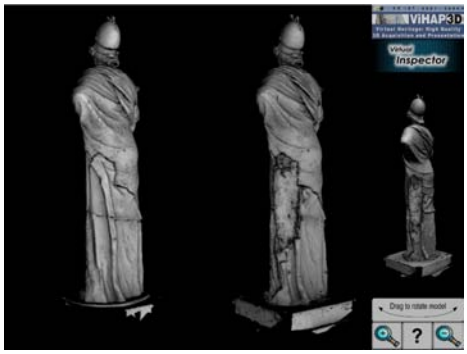

Figure 5: VIRTUAL INSPECTOR *showing two models of the Minerva statue, scanned at different times during restoration.*

mouse on the corresponding point). Then, a new hot spot is specified by introducing a new 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 an example concerning the Arrigo VII installation [BBC*04] in Figure 6).

9. Evaluation and use of the VIRTUAL INSPECTOR tool

The VIRTUAL INSPECTOR tool was originally conceived in the framework of a cooperation with the Centro di Restauro (Restoration Laboratory) of the Soprintendenza Archeologica Toscana in Florence, a cooperation aimed at the restoration of the Minerva statue and started in year 2000. This cooperation made clear the need of an easy-to-use tool which could allow the restorers to access the accurate 3D model. One basic requirement was obviously the capability to show the 3D data selectively, giving to the user the capability to access even the high resolution model interactively and without loosing accuracy and detail. The first experimentation of the VIRTUAL INSPECTOR tool were very encouraging, fulfilling the above requirements. A first release of the system was evaluated by the restorers and was considered very handy. Since then, the various version of VIRTUAL INSPECTOR have been routinely used in the scanning projects of our group

Figure 7: 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).*

10. Conclusions

We have presented the VIRTUAL INSPECTOR tool, an interactive system for the visualization of complex and accurate digital 3D models. It has been 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 proposal does not rely in the single techniques used, but rather in the design of a complex system 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 specification listed in the introduction section.

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

References

[BBC*04] BARACCHINI C., BROGI A., CALLIERI M., CAPITANI L., CIGNONI P., FASANO A., MONTANI C., NENCI C., NOVELLO R. P., PINGI P., PONCHIO F., SCOPIGNO R.: Digital

Figure 6: *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). On the right, visualization of the 3D model of Arrigo VII enthroned.*

reconstruction of the Arrigo VII funerary complex. In *VAST 2004* (Bruxelles, BE, Dec. 7-10 2004), Y. Chrysanthou K. Cain N. S., Niccolucci F., (Eds.), Eurographics, pp. 145–154.

[BCS01] BORGO R., CIGNONI P., SCOPIGNO R.: An easy to use visualization system for huge cultural heritage meshes. In *VAST 2001 Conference Proc.* (Athens, Greece, Nov. 28-30 2001), Arnold D., Chalmers A.,, Fellner D., (Eds.), ACM Siggraph, pp. 121–130.

[BR02] BERNARDINI F., RUSHMEIER H. E.: The 3D Model Acquisition Pipeline. *Computer Graphics Forum 21*, 2 (March 2002), 149–172.

[BWK02] BOTSCH M., WIRATANAYA A., KOBBELT L.: Efficient high quality rendering of point sampled geometry. In *Proceedings of the 13th Eurographics Workshop on Rendering (RENDERING TECHNIQUES-02)* (Aire-la-Ville, Switzerland, June 26–28 2002), Gibson S., Debevec P., (Eds.), Eurographics Association, pp. 53–64.

[CGG*05] CIGNONI P., GANOVELLI F., GOBBETTI E., MARTON F., PONCHIO F., SCOPIGNO R.: Batched multi triangulation. In *IEEE Visualization 2005* (2005), pp. 27–35.

[CMRS03] CIGNONI P., MONTANI C., ROCCHINI C., SCOPIGNO R.: External memory management and simplification of huge meshes. *IEEE Transactions on Visualization and Computer Graphics 9*, 4 (2003), 525–537.

[CS00] CURLESS B., SEITZ S.: 3D Photography. In *ACM SIGGRAPH 00 Course Notes, Course No. 19* (2000).

[GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings* (Aug. 1997), Annual Conference Series, Addison Wesley, pp. 209–216.

[GM04] GOBBETTI E., MARTON F.: Layered point clouds – a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models. *Computers & Graphics 28*, 6 (December 2004).

[KTL*04] KOLLER D., TURITZIN M., LEVOY M., TARINI M., CROCCIA G., CIGNONI P., SCOPIGNO R.: Protected interactive 3D graphics via remote rendering. *ACM Trans. Graph 23*, 3 (2004), 695–703.

[LKG*03] LENSCH H. P. A., KAUTZ J., GOESELE M., HEIDRICH W., SEIDEL H.-P.: Image-based reconstruction of spatial appearance and geometric detail. *ACM Transaction on Graphics 22*, 2 (Apr. 2003), 234–257.

[LPC*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINZTON M., ANDERSON S., DAVIS J., GINSBERG J., SHADE J., FULK D.: The Digital Michelangelo Project: 3D scanning of large statues. In *SIGGRAPH 2000, Computer Graphics Proceedings* (July 24-28 2000), Annual Conference Series, Addison Wesley, pp. 131–144.

[NDW93] NEIDER J., DAVIS T., WOO M.: *OpenGL Programming Guide*. Addison Wesley, 1993.

[Pup96] PUPPO E.: Variable resolution terrain surfaces. In *Proceedings Eight Canadian Conference on Computational Geometry, Ottawa, Canada* (August 12-15 1996), pp. 202–210.

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: Surface elements as rendering primitives. In *SIGGRAPH 2000, Computer Graphics Proceedings* (2000), Akeley K., (Ed.), Annual Conference Series, ACM Press - Addison Wesley Longman, pp. 335–342.

[RCM*01a] ROCCHINI C., CIGNONI P., MONTANI C., PINGI P., SCOPIGNO R.: A low cost 3D scanner based on structured light. *Computer Graphics Forum (Eurographics 2001 Conf. Issue) 20*, 3 (2001), 299–308.

[RCM*01b] ROCCHINI C., CIGNONI P., MONTANI C., PINGI P., SCOPIGNO R., FONTANA R., PEZZATI L., CYGIELMAN M., GIACHETTI R., GORI G.: 3D scanning the Minerva of Arezzo. In *ICHIM'2001 Conf. Proc., Vol.2* (2001), Politecnico di Milano, pp. 265–272.

[RL00] RUSINKIEWICZ S., LEVOY M.: QSplat: A multiresolution point rendering system for large meshes. In *Comp. Graph. Proc., Annual Conf. Series (SIGGRAPH 00)* (July 24-28 2000), ACM Press, pp. 343–352.