CEIG'08, Barcelona, Sept. 3–5 (2008), pp. 1–6
L. Matey and J. C. Torres (Editors)

Real-Time Exploration of the Virtual Reconstruction of the Entrance of the Ripoll Monastery

Isaac Besora[2], Pere Brunet[1,2], Antoni Chica[1,2], Daniel Morales[2] and Jordi Moyés[2]

[1]Department of Software, UPC, Barcelona
[2]MOVING Research Group, UPC, Barcelona

Abstract

This paper presents the project of the virtual reconstruction and inspection of the "Portalada", the entrance of the Ripoll Monastery. In a first step, the monument of 7 x 11 meters was acquired using triangulation laser scanning technology, producing a dataset of more than 2000 range maps for a total of more than one billion triangles. After alignment and registration, a nearly complete digital model with 173M triangles and a sampling density of the order of one millimeter was produced and repaired. The paper describes the model acquisition and construction, the use of specific scalable algorithms for model repair and simplification, and then focuses on the design of a hierarchical data structure for data managing and view-dependent navigation of this huge dataset on a PC. Finally, the paper describes the setup for a usable, user-friendly and immersive system that induces a presence perception in the visitors.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications

1. Introduction

The main work of art from the Ripoll Monastery (founded in the year 879) is the entrance, which dates back to the 12th century. This entrance (also known as the "Portalada") is the the main Romanic sculpture from Catalonia. It has been defined as the "Stone Bible", since it represents many fragments of the book of the "Ripoll Bible", written during the 11th century and now in the Vatican Museum. It is a masterpiece of cultural, historical, social and scientific interest.

The romanic experts and museum curators at the Museu Nacional d'Art de Catalunya (MNAC) in Barcelona were interested in having a virtual reproduction of this significant masterpiece, to bring it to the visitors in Barcelona.

This paper presents the project of the virtual reconstruction and presentation of Portalada, which produced an interactive installation which was open to the public during the MNAC exhibition named "The Romanic Art and the Mediterranean. Catalonia, Toulouse and Pisa" from February to May 2008.

The project objectives derive from the museum requirements. The goal of the project is three-fold: having a high-fidelity virtual reproduction for the exhibition, creating a tool for the study and analysis by experts, and having a digital model that allows the archival of the present status of the Monastery entrance. Visitors of the exhibition are interacting with the virtual reconstruction in two different immersive setups (VR kiosks). They use a touch-screen for interaction. A back-projection display screen with passive stereo allows immersive navigation and zooming in different parts of the portalada.

The overall project involved a number of challenges and novel contributions that required the development of specific algorithmic tools. The requirements included the millimetric precision of the final digital model, the tight schedule (four months for the whole project) and the use of low-cost and commercial components including the computer. The project itself has produced four major contributions:

- The acquisition and model construction process which, given the very large sculpted surface (7 x 11 meters), produced a complete 173M faces high resolution model, with a sampling density of the order of one millimeter
- The derivation of specific scalable algorithms for model repair and simplification.

- The design of a hierarchical data structure for data managing and view-dependent navigation.
- The setup of a usable, user-friendly and immersive system that induces a presence perception in the visitors.

This paper focuses in the last two contributions, that are described in Sections 3 and 4. The first two contributions have been addressed in two separate papers and are briefly described in Section 2. The work is based on a number of previous works that are cited and discussed in the next Sections. The project links with a number of hot topics in computer graphics, like model acquisition in Cultural Heritage Applications, geometry processing algorithms for model repair and simplification, view-dependent visualization and algorithms for managing gigantic models.

Figure 1: *Acquisition process in Ripoll, November 2007.*

2. Acquisition and Model Generation

The portalada of the Ripoll Monastery is an extremely rich monument, with a size of 11 x 7 meters. The statues and bass-reliefs present many small details that should be preserved. The requirements were to obtain a high quality digital model (a digital copy of the portalada) with a precision of the order of one millimeter.

The initial discussions concluded on the need of a double scanning of the complete portalada: a Time of Flight scanning for overall alignment, and a number of precise, triangulation- based scans that could preserve and acquire all details. The idea was to use the Time of Flight scan results as a background low-resolution mesh, on which the different high-resolution patches would be placed.

The Time of Flight scanning was performed by using a Leica ScanStation. It took 1 day of work and 4 acquisition stations to acquire data of both the Portalada and the exterior monastery facade. The Portalada was scanned at a resolution of 0.5 cm, providing a data set of 36.2 Million points.

The triangulation-based scanning was performed in 5 days, in November 2007 by a team of 4 people using 2 Minolta Vivid 910 scanners. In order to be able to scan the upper part of the portal, a movable 7 meters tall scaffolding was used (Figure 1). The acquired data set was organized in a total of 2212 range maps and more than half a Billion 3D points, see [BBC*08].

Alignment was performed by the CNR team, [BBC*08]. A manual, out-of-core, multiscale global alignment step was used, as the standard alignment tools, both commercial and academic, are designed to work on datasets that are smaller than the one produced in this project. The Time of Flight models were used several times in the course of processing for the initial assessment of the data, to perform the initial alignment of the photographic dataset and as a geometrical reference during the alignment of the triangulation range maps. The entire model was divided into subparts, and the alignment process was composed of three different steps [FDG*05], iterated several times in order to guarantee a stable and accurate final alignment: alignment between range maps inside each subpart, alignment step of range maps of a single subpart toward the TOF model, and alignment between range maps of adjacent subparts, [BBC*08].

After generating a correct alignment for all the range maps, the next step was to create a single surface for the entire object and to associate the color information to the geometry of the model. This is usually done by registering a set of images on the model, and projecting the color values on it. In our particular case, 163 images were calibrated (Figure 2) and projected on the geometry, by associating a color value to each vertex of the model [CCCS07].

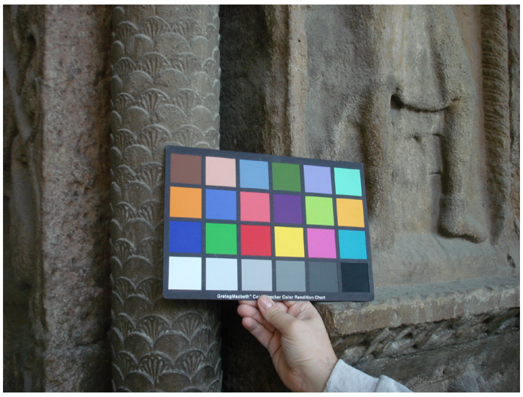

Figure 2: *Color acquisition in Ripoll: color calibration for the photos.*

The final step in the model generation was model repair. Apart from detecting and repairing small imperfections (ill-oriented triangles or spurious polygons), the model contained more than 25000 small holes to be repaired. Holes and cracks were automatically detected through a search for cycles of border edges. We decided to implement a multigrid version [BCNV08] of the fitting and smoothing loop proposed in [EBV05] and [EBV08]. The goal was to obtain a single mesh without border edges.

The final repaired mesh had 173 millions of triangles, with an average edge length of 1.4 millimeters and color per vertex.

3. Algorithms for Real-Time Visualization

Our main goal was to have real-time, interactive visualization of this gigantic mesh with 173 Mtriangles on a standard PC. The requirement was that visitors should have the possibility of freely moving along the monument and zooming in any place or any detail; the system should then react in real-time and present the digital model in the zoomed-in region, with the maximum resolution of the digital mesh.

Our approach is based on a preprocess step and on the concurrent use of a number of recent techniques:

- View-dependent, out-of-core algorithms for real-time interactive rendering of gigantic meshes. The progressive buffers in [SM05] and the far voxels algorithm in [EM05] are good references of algorithms for the interactive inspection of huge models, using textures, levels of detail and hybrid geometric/impostor representations.
- Hierarchical data structures of large granularity, in order to avoid too many changes and updates in the view-dependent front. See the algorithms in [GM04] using layered point clouds with non-deep hierarchical data structures.
- Rendering of a low-resolution mesh with high-quality textures, for distant positions of the observer. Our approach uses the results from Tarini et al. in [TCS03].
- Relief impostors for the recovery of precise details at medium distances. The work in [BD06] is a good reference for rendering geometry with relief textures, using specific fragment programs.

In our current implementation, we have observed that it is possible to obtain a high visual quality by using a three-level multiresolution model. At observer locations that are farther away than 2 meters, from the portalada, we simply render a textured base mesh which is permanently located in the GPU memory through a number of vertex buffer objects. For intermediate distances, we use a relief impostor [ABB*07] based representation in the zones that require a quality higher than the one supplied by the textured base mesh. For close-up views (distance between the observer and the monument that are lower than one meter), we render the relevant high resolution meshes.

In the preprocess step, several data structures and geometric models have been computed:

- A set of disjoint high-resolution meshes M_k. The overall mesh was divided onto 3000 small meshes $M_1, M_2, ..., M_{3000}$ of around 60000 triangles each. Every triangle of the global 173 Mtriangles mesh belongs to one and only one of the M_k meshes, while vertices are duplicated along boundaries between neighbor meshes. Obviously, the result of rendering all M_k meshes is exactly the same as rendering the huge high-resolution mesh. Managing small meshes is much more easy, and the memory overhead for duplicate vertices is very small (less than 2 percent). These small meshes allow the optimization of the performance of the CPU and GPU memory caches and of the CPU-GPU traffic.
- An octree data structure that manages the camera movements and the view dependent visualization at each frame.
- A textured base mesh, with 200 K triangles. A 200 K triangle simplification of the initial mesh was first obtained by classical vertex clustering techniques. Textures were then obtained by projecting points of the high resolution mesh onto the 3D points of the low-resolution triangles that correspond to each of their texels. Color textures and normal maps were encoded on a texture Atlas, the size of the Atlas being 550 MBytes.
- A number of relief textures for intermediate rendering, using the algorithm in [ABB*07].

Our octree supports the hierarchical data structure management of the multiresolution meshes. The octree data structure is permanently in-core during the navigation. In our present implementation, the number of leaf nodes containing parts of the mesh (surface terminal nodes) is 283617, and the number of terminal void nodes is 389945. The octree contains 96286 non-terminal (grey) nodes. The depth of the octree is 10 levels, giving a node edge size of 3.51 centimeters for a Universe size of 18 x 18 x 18 meters. Surface terminal nodes and grey nodes contain a mask pointing to the meshes M_k partially or totally contained in the node's cube. Grey nodes also represent a hierarchical linear distance field which approximates the distance from the observer to the portalada. Every octree grey node stores a 3D point, its distance value and the gradient of the distance field. During octree construction, the maximum deviation between the linear approximation of the distance and the exact distance to the portalada is computed for each node being processed. It this deviation exceeds a predefined tolerance, the node is subdivided and the distance gradient is computed for each of its son nodes. Octree nodes also contain pointers to the portalada relief impostor models.

The interactive visualization uses a view-dependent algorithm which derives from the methods in [GM04] and [EM05]. An octree traversal is performed at each frame to first detect if the movement of the observer is valid or not. If the new observer location is valid (he is not exiting the octree Universe Cube and his distance to the surface of the portalada is greater than 30 centimeters), the view-dependent front is obtained and the list of models that should be rendered is computed. The octree traversal is also used for hierarchical frustum culling and for the generation of the list of visible submeshes for the present frame. Nodes of the active octree front are computed at each frame on the basis of the size of the projection of their corresponding cube in pixel coordinates. If this size is larger than a pre-defined tolerance in pixels, the front is searched in the descendants of the node. If it isn't, the node itself is added to the list of front nodes for this frame. The projection test is very fast. It is implemented as a linear search in a precomputed array, the size of the array being the number of octree levels. This array stores the expected squared distance to the observer for front nodes of the different levels in the octree. Distances between the observer and the surface of the portalada are also computed using the hierarchical linear distance field and require a single test per octree level.

Once the list of front nodes is obtained, a list of relief impostors and/or high resolution meshes that should be rendered in this frame is computed. Two-level node masks are used to select the right set of high-resolution meshes. Each octree node at a depth greater than four, has a 11 byte mask that represents which of the high-resolution meshes M_k intersect the node, within the set of meshes M_a intersecting its ancestor at the fourth level of the tree. The mask works as a bit mask, the M_k mesh in M_a being intersected by the octree node iff the $k-th$ bit in the mask has been set to one. We have observed that the number of elements in the sets M_a is never greater than 88. When a new front node is detected, the list of high resolution meshes that should be rendered in this frame is updated by simple bit operations: look-up in a mask dictionary corresponding to M_a in order to obtain a global mask, and a bitwise or operation.

Each visible submesh inside the frustum is assigned a priority which is related to the distance in pixels between the projection of a representative mesh point and the center of the viewport. The rendering algorithm uses CPU and GPU cache storages.

A lazy CPU-GPU communication algorithm is used, where a single mesh per frame M_k is sent from the CPU to the GPU: the one having the highest priority in the CPU list, among the ones still not being sent. We use a fragment program that implements shading and shadow algorithms.

Figure 3 shows a detail of the high-resolution mesh (the Pantocrator, that cannot be inspected in the real portalada because of being located at 7 meters from the floor), and Figure 4 presents the projected imaged of an sculpture in the screen of one of the kiosks.

Figure 3: *A detail of the high-resolution mesh: the Pantocrator.*

Figure 4: *A snapshot of the interaction in the screen of the VR kiosk.*

4. User Interface for the Immersive Inspection

Visitors can interact with the digital model of the portalada through two VR kiosks. They are identical, and visitors are directed to one or to the other depending on the queues and the demand. Visitors are provided with passive stereo glasses at the entrance of the kiosks room and must return them at the exit. These two kiosks have been designed and constructed during the project, by following the specifications given by the Museum responsibles.

One of the important features of the kiosks is that they are low-cost and easy to assembly. The components of one kiosk include,

- One dual-core PC with an Nvidia 8800 GTX graphics board. This PC handles the visualization of the model. It manages the high-resolution meshes and computes the view-dependent front of the octree at every frame.
- Two DLP projectors with circular polarizing filters, fixed in a calibration frame that was designed in the framework of one of the previous projects of our group.
- A back-projection screen with a surface of 2 x 1.5 meters.
- A touch-screen for the interaction, see Figure 5.
- A second PC to manage the interface in the touch-screen. Both PCs are connected through a LAN.

All these parts are easy to obtain, on the shelf components. The touch-screen was selected instead of more advanced 3D

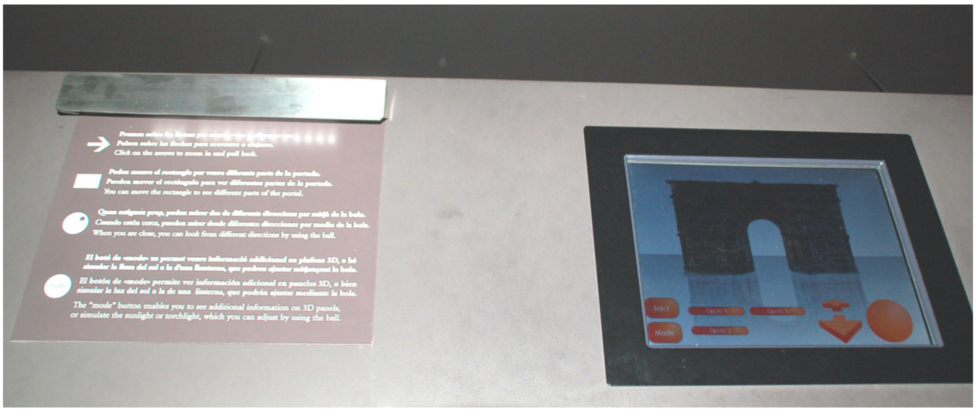

Figure 5: *User interface and instructions for the visitors.*

interaction devices because of a strong anti-vandalic requirement.

The user interface in the touch-screen (Figure 5) presents a view of the portalada and a few buttons and widgets. Besides the touch-screen, a number of brief instructions in three different languages is available. The representation of the portalada without the surrounding walls of the Monastery is intended to highlight its similarity with the Arcs of the Roman Empire. Visitors can approach by pressing on the perspective arrows, perform a "pan" operation to go to specific parts of the monument by simply moving the finger along the portalada to the place they are interested in, or rotate the camera by virtually rotating the "track ball" at the lower right corner. The interface PC sends the camera parameters to the navigation PC, which detects if this new position is valid or not. In the affirmative case, the camera position is updated and the information is sent back to the interface PC which updates the echo of the camera move. In the negative case, the camera is not updated and the visitor has the perception of having a collision with the surface of the portalada. Updating the camera basically means updating the observer and view reference points, this last one always being forced to be the intersection of the viewing direction with the surface of the monument. The echo on the touch-screen consists on the display of a semitransparent quadrilateral representing the projection of the result of intersecting the view frustum with a plane parallel to the Z-far plane and containing the view reference point. This results in a quite intuitive feedback of the part of the portalada the camera is looking at.

The user interface has two more buttons: a reset button ("inici") that moves back the camera to a general view of the portalada, and a "mode" button. The application supports three modes, that can be reached in a "circular" way by repeatedly pushing the mode button: the navigation mode, the sun illumination mode and the information mode. The navigation mode allows standard navigation as already described, the sun illumination mode simulates solar lighting while the user changes the time of the day and the day of the year using two standard sliders, and the information mode (the one shown in Figure 5) displays additional information (historical photos or texts) on request.

Figure 6 shows the Setup of one of the kiosk settings in the room of the MNAC Museum, while Figure 7 shows the interaction with the 3D model at one of the kiosks.

Figure 6: *Setup of one of the kiosks at the MNAC Museum.*

Figure 7: *Interacting with the 3D model at the VR kiosk.*

5. Conclusions

In this paper, we have presented the project of the virtual reconstruction of the entrance of the Ripoll Monastery, the so-called "portalada". The Monastery of Ripoll is located 90 Kilometers North of Barcelona, being one of the greatest and best Romanic sculptures in Spain and the main Romanic Monument in Catalonia. The project is the result of a cooperative work involving the MNAC Museum in Barcelona, the UPC group , ISTI-CNR graphics group in Pisa and the Bishop of Vic.

The project has faced a number of important challenges, from the tight schedule (the overall project has had to be completed in four months) to the huge amount of data and the quality and precision requirements.

Besides the acquisition and model repair steps, this paper focuses on the algorithms and data structures for real-time handling and visualization of the high resolution mesh and on the interaction paradigms and immersive visualization for the Museum exhibition. The main conclusion is that the high resolution model with 173 Mtriangles and color per vertex can be inspected in real-time and in stereo, using a simple PC for the interactive rendering and a set of special hierarchical view-dependent algorithms. The whole immersive setting is based on standard, low-cost components.

The system has been placed in a temporal exhibition in the MNAC Museum, and it has been open to the general public from March to May 2008. The visitors evaluation has been very positive, the consequence being that one of the VR kiosks will remain in the future as part of the permanent collection in the Museum.

6. Acknowledgements

This work has been promoted and funded by MNAC, the National Art Museum of Catalonia. It has also been partially funded by the Spanish Ministry of Science and Technology under grant TIN2007-67982-C02-01. It is a cooperative project involving the MNAC, UPC, ISTI-CNR and the Bishop of Vic.

The authors would like to thank the members of the ISTI-CNR group in Pisa for their essential participation in the acquisition and model generation phases of the project. They also would like to thank Alvar Vinacua, Carlos Andujar and Isabel Navazo for their ideas and discussions during the design of the user interface and the geometry processing and rendering algorithms. They would also like to thank Mel Slater for his suggestions on the user interface, and Silvia Ramos, Eva Monclus, Ferran Argelaguet and the rest of the UPC Moving group for their help in the usability tests and in the implementation of some of the algorithms.

References

[ABB*07] ANDUJAR C., BOO J., BRUNET P., FAIREN M., NAVAZO I., VAZQUEZ P., VINACUA A.: Omnidirectional relief impostors. *Computer Graphics Forum 26*, 3 (September 2007), 553–560.

[BBC*08] BESORA I., BRUNET P., CALLIERI M., CHICA A., CORSINI M., DELLEPIANE M., MORALES D., MOYES J., RANZUGLIA G., SCOPIGNO R.: A virtual reconstruction of the entrance of the ripoll monastery. *Submitted for publication* (2008).

[BCNV08] BRUNET P., CHICA A., NAVAZO I., VINACUA A.: Massive mesh hole repair minimizing user intervention. *Submitted for publication* (2008).

[BD06] BABOUD L., DECORET X.: Rendering geometry with relief textures. In *Proc. Graphics Interface 2006* (2006), pp. 195–201.

[CCCS07] CALLIERI M., CIGNONI P., CORSINI M., SCOPIGNO R.: Masked photo blending: mapping dense photographic dataset on dense 3d models. *Submitted for publication* (2007).

[EBV05] ESTEVE J., BRUNET P., VINACUA A.: Approximation of a variable density cloud of points by shrinking a discrete membrane. *Computer Graphics Forum 24*, 4 (December 2005), 791–808.

[EBV08] ESTEVE J., BRUNET P., VINACUA A.: Piecewise algebraic surface computation and smoothing from a discrete model. *Computer Aided Geometric Design, accepted 20 September 2007, available online* (2008).

[EM05] E.GOBETTI, MARTON F.: Far voxels: A multiresolution framework for interactive rendering of huge 3d models. *ACM Transactions on Graphics, Siggraph'05 24* (2005).

[FDG*05] FRANKEN T., DELLEPIANE M., GANOVELLI F., CIGNONI P., MONTANI C., SCOPIGNO R.: Minimizing user intervention in registering 2d images to 3d models. *The Visual Computer 21* (sep 2005), 619–628.

[GM04] GOBBETTI E., MARTON F.: Layered point clouds: a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models. *Computer and Graphics 28* (2004), 815–826.

[SM05] SANDER P. V., MITCHELL J. L.: Progressive buffers: View-dependent geometry and texture lod rendering. In *Eurographics Symposium on Geometry Processing 2005* (July 2005), pp. 129–138.

[TCS03] TARINI M., CIGNONI P., SCOPIGNO R.: Visibility based methods and assessment for detail-recovery. *IEEE Visualization 2003* (October 2003), 457–464.