A SUITE OF TOOLS FOR THE MANAGEMENT OF 3D SCANNED DATA

C. Rocchini, P. Cignoni, C. Montani, P. Pingi, R. Scopigno

Istituto di Scienza e Tecnologie dell'Informazione (ISTI) – Consiglio Nazionale delle Ricerche
Area della Ricerca CNR – S. Cataldo, 56100, Pisa, Italy
Email: {rocchini|cignoni|montani|pingi}@iei.pi.cnr.it , r.scopigno@cnuce.cnr.it

Abstract

In this paper we describe a 3D scanning software suite which gives support to most of the processing phases of a complex 3D scanning project. In particular, our tools support: range maps alignment, range maps merge, mesh editing and mesh simplification. The software suite has been implemented by scratch and encompasses both up-to-date solutions (alignment, editing) and some original methods (merging, simplification, and, in part, alignment). The architecture of the software suite is described and an evaluation of its use in the framework of a complex acquisition in the Cultural Heritage domain (3D scanning of a bronze statue) is reported.

KEY WORDS: 3D scanning, range maps, alignment, surface editing, merging, surface simplification.

1 Introduction

3D scanning technology evolved considerably in the last few years, both in terms of hardware devices and of algorithms for processing the raw data produced by scanning devices [1]. Many different devices exist, including both academic prototypes and commercial systems. Unfortunately, 3D scanning is not as simple as scanning images. The acquisition of a complex object cannot be performed by simply pressing a button. Recent devices based on the *reconstruction from silhouette* approach are the ones that more closely match the ideal *push a button* operative mode; on the other hand, the low accuracy supported by these systems makes them ideal only for e-commerce applications. In all other cases, a more accurate approach has to be endorsed (e.g. active scanning based on laser or structured light) and complex objects are modelled via the acquisition of a set of partially overlapping range scans. The classical pipeline which characterizes a 3D scanning session is rather complex, involving many different operations. The available software tools are rather incomplete, or in some cases they implement rather old and inefficient solutions. In particular, commercial software often gives unsatisfactory tools for the management of the range map registration phase, inefficient software for the simplification of the meshes produced and limited (or lacking) support for range maps which also hold attribute data (e.g. color). In this paper we describe a 3D scanning software suite which gives support to most of the processing phases of a complex 3D scanning project.

2 The VCG's 3D scanning tools

The pipeline of phases of any 3D scanning session is rather complex, involving many steps that can be resumed as follows:

- **acquisition planning**; decide the set of range maps (how many, view specification of each of them) to be taken for the complete acquisition of the surface of the object;
- **scanning** the artefact;
- range maps **alignment**; all the range maps have to be aligned to lie in the same space;
- range maps **merging**; build a single, non redundant mesh out of the many, partially overlapping range maps;
- mesh **editing**; improve (if possible) the quality of the reconstructed mesh;
- mesh **simplification**; reduce the huge complexity of the model obtained;
- finally, mesh **conversion**; export data to the representation scheme used by the application of interest.

A comprehensive tutorial of the techniques proposed in literature and most frequently used in applications has been recently proposed in [1].

We have designed a suite of software tools that manages most of these phases:

- *MeshAlign*: the module allows the registration of multiple range maps; by definition, range map geometry is relative to the current sensor location and has to be transformed into a unique reference space, such that the sections of the range maps which correspond to the same surface zone will be geometrically overlapping.
- *MeshMerge*: the module allows the reconstruction of a single 3D mesh out of a set of registered range maps. It adopts a new approach [13], which is characterized by a lower space complexity, higher efficiency and improved accuracy with respect to the current alternative volumetric reconstruction approaches [6].
- *MeshEdit*: the module allows to perform simple editing actions on the mesh (e.g. to fill small holes, to remove non-manifold components of dangling edges/faces, to apply smoothing filters, etc).
- *MeshSimplify*: the module supports the simplification of the [huge] meshes produced by 3D scanning devices, by removing mesh vertices in a controlled manner. The simplification follows the edge collapse approach [8] and has been implemented in an out-of-core fashion to allow the management of meshes that could be larger than the core memory of the computer used [4].

The above modules are described in detail in the following sections. The system has been defined as a suite of independent software modules to reduce the overall complexity, in terms of both *graphical user interface* (GUI) complexity and *space occupancy* (each module requires appropriate data structures, and therefore having all the modules in a single application would imply an excessive space occupancy and redundancy). A clever GUI design has a primary importance: most of the processing involved in 3D scanning management requires a complex user interaction (think for example to the case of range maps alignment or mesh editing), and therefore a single system would make the GUI excessively complex. Moreover, designing the system in terms of a set of stand-alone modules beside the improvement in space occupancy and GUI usability makes the software design of the whole suite much easier allowing us to build a more robust software system in less man-time.

3 *MeshAlign*

This module allows to the user to align all the acquired range maps, which by definition represent the distance of the surface sampled points from the sensor location. Many different locations of the scanner are needed to get a complete coverage of the object surface. This means that all these range maps have

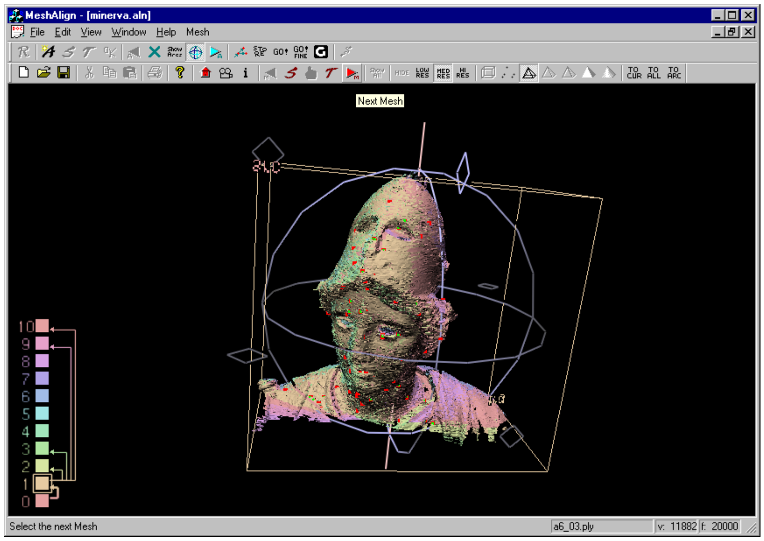

Figure 1: The graphic user interface of the *MeshAlign* module.

a different coordinate system and we have to move them in a common reference system. This process, called alignment, is partially manual and partially automatic. The user has to find a raw registration between two maps and the system uses this approximation to compute a very accurate registration of the two meshes. This pairwise registration process (repeated on all pairs of adjacent and overlapping range maps) is then used to automatically build a global registration of all the meshes, and this last alignment is enforced among all the maps in order to move everything in a unique reference system.

This software module requires the more complex interface (together with *MeshEdit* module) because easy to use tools supporting an interactive and complex manipulation of the meshes are required.

The registration module follows the approach proposed by K. Pulli [10], which is based on a variation of the Iterated Closest Point algorithm [2, 3]. In particular, our implementation is improved by the adoption of a multiple level of detail representation of the range maps which allows to improve the performance and accuracy of the process. It has to be considered that the range maps produced by current scanning devices can hold a resolution of 1000*1000 samples, or more. Therefore, the manual alignment of a pair of range maps can require the interactive manipulation of around 4M triangles. The availability of multiple level of detail [11] is therefore a handy resource to allow smooth and efficient management of the interactive visualization and manipulation on low cost platforms (PCs). Moreover, note also that a standard registration task forces to manage tens, or even hundreds of different range maps. Obviously, the low-resolution approximated range maps are used only in the visualization process and in the first steps of the alignment; in the further steps the system automatically switches to the high resolution maps in order to get the maximum precision.

An image of the interface of the *MeshAlign* module is shown in Figure 1. Figure 2 presents the results of the approximated alignment provided by the user (on the left), and the following high quality alignment automatically computed by the *MeshAlign* module (on the right).

4 *MeshMerge*

Once all the range maps have been aligned together in a common Carthesian space, a unique single mesh is obtained by an automatic merging tool based on a new algorithm, *Marching Intersections*, described in [13]. The algorithm locates the intersections between the range images and a reference grid (chosen by the user), merges nearby intersections relative to overlapped range map sections taking into account the quality of the range data, and finally reconstructs the merged surface. A 2D example is presented in

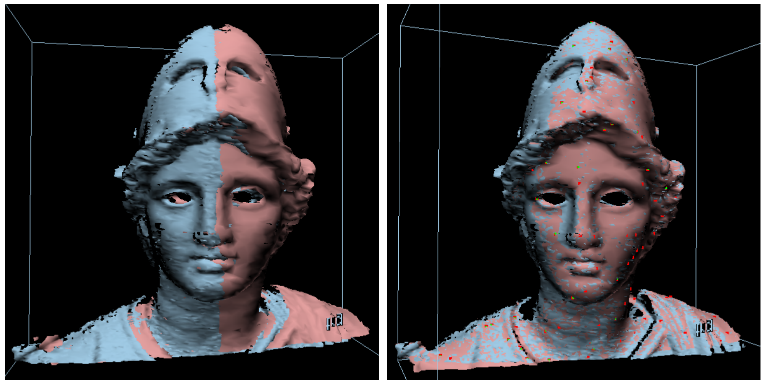

Figure 2: A comparison between the approximated alignment provided by the user (on the left) and the hi quality alignment computed automatically by the *MeshAlign* module (on the right).

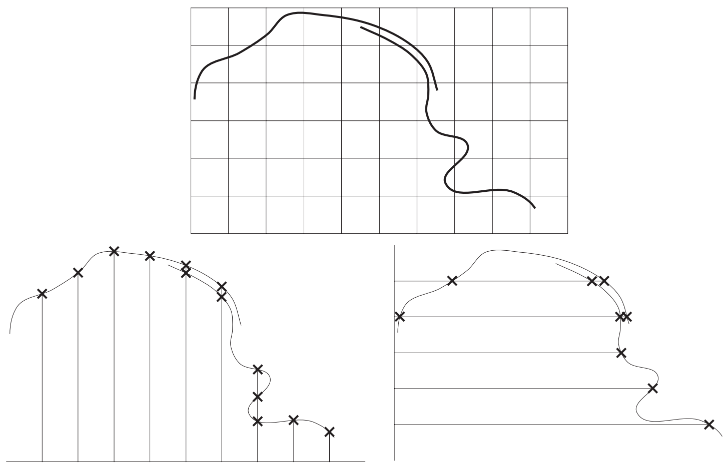

Figure 3: A 2D example of the merging process. Given two sample range curves (top), the intersections with the regular grid are found and collected in the two dynamic data structures (vertical and horizontal intersections); the merged representation is then reconstructed from these discrete set of intersections.

Figure 3. The size of the reference grid is specified by the user and influences the quality of the merged result: the smaller the grid spacing, the more accurate is the reconstructed mesh, but also the more complex is the mesh (no. of triangles). During this phase, the small holes (due to parts of the input model invisible to the acquisition sensor) are filled in an automatic manner. The *MeshMerge* module can correct automatically some small topological anomalies due to acquisition noise or the small differences among multiple scans of the same surface region.

This module take also into account the possible presence of color information in the scans and it coherently fuses them together in the resulting mesh.

The algorithm used for the mesh fusion [13] is characterized by a lower space complexity, higher efficiency and improved accuracy with respect to mostly used alternative volumetric reconstruction approaches [6].

The system can also manage the fusion of a number of meshes so large that the resulting mesh is much larger than the available RAM memory. *MeshMerge* subdivides the space occupied by the whole mesh into a given number of sub-blocks and iterates the fusion algorithm onto each sub-block separately; the boundary of the subblocks are guaranteed to be identical so the joining of resulting sub-meshes is

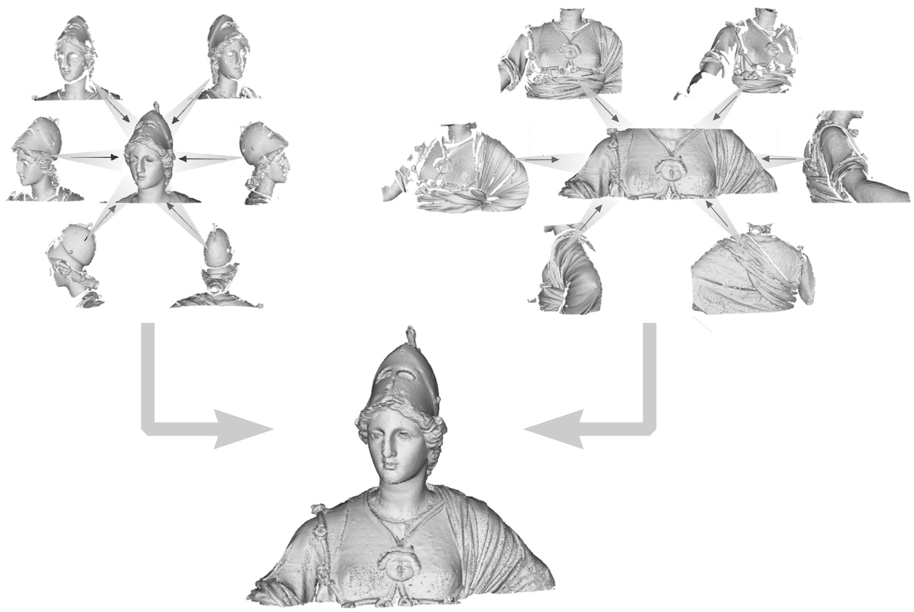

Figure 4: The range image fusion algorithm is run separately for each sub-portion of the space containing all the range images and then the resulting meshes are joined together.

trivial.

Figure 4 shows the effects of the fusion of a small set of range images; the space containing all the range images is subdivided and the fusion algorithm is run separately for each sub-block, then the two resulting mesh are joined together.

5 *MeshEdit*

This module is aimed to allow all semiautomatic mesh-editing tasks that often are necessary to *clean-up* and/or improve the results of all the various steps. In particular we have find essential the following functionalities:

- **Hole Filling**. Small holes, the one composed by a few edges, can be safely filled in an automatic manner. Larger ones need often some human intervention, usually in judging what automatic fill strategy gives the best results.
- **Small components removal**. The presence of noise in the scanning process can produce a lot of small spurious unconnected components during mesh integration and merging phase. All these disconnected components can be automatically removed from the mesh, improving overall mesh quality.
- **Topology Enforcing**. Producing an output meshs with a clean topology is a must for many applications. Non two-manifold situations can be produced during mesh simplification or mesh editing, or can be due to geometric robustness and limited arithmetic accuracy of the geometric processing codes. Many of these problems can be automatically removed.
- **Selective Smoothing**.
- **Mesh Conversion**.

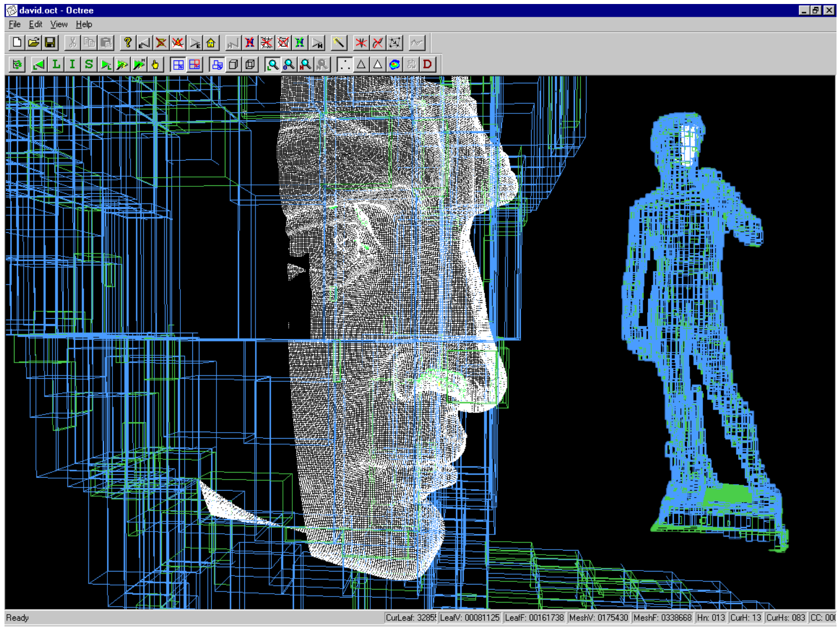

Figure 5: The graphic user interface of the simplification system.

6 *MeshSimplify*

The result of the integration of a large number of scans can be composed by millions of triangles. An example is the high complexity of the meshes produced by the Digital Michelangelo Project [9]. Obviously, processing, rendering, transmission and archival of these meshes are not simple tasks. Mesh simplification and LOD management are a rather mature technology [11, 7, 5] that in many cases can efficiently manage complex data. But only few available algorithms can manage meshes characterized by a huge size: RAM size is often a severe bottleneck because, currently, most simplification systems requires that the whole mesh is loaded in memory before the processing, requiring therefore a really huge quantity of memory. Our system is characterized by the adoption of an highly innovative out of core data structure described in [4] that allows the simplification and the interactive visualization/inspection of really huge meshes.

The simplification technique used is an incremental edge-collapse algorithm based on the one presented in [8] that makes use of *error quadrics* to measure the error introduced during simplification.

The innovative data structure used for *out-of-core* mesh management [13] is based on an octree decomposition of the mesh; it is characterized by a unique indexing of the vertexes of the mesh and maintains explicit representation of the mesh topology. This representation scheme is fairly general; we plan a future extension to use it for the implementation of all the algorithms supported by the *MeshEdit* module.

Figure 5 shows a snapshot of the user interface of the *MeshSimplify* module.

7 Results and Conclusions

Our 3D scanning tools have been used at the Restoration Laboratory of he Soprintendenza Archeologica Toscana in the framework of the restoration of the Minerva, a bronze statue of the Museo Archeologico at Florence. As is usual in 3D scanning, complex objects are modeled via the acquisition of a set of partially overlapping range scans. The whole Minerva statue (170 cm.high) has been scanned in October 2000 using an active optical scanner developed at CNR [12]. The statue was acquired by taking 146 range maps. After range map reconstruction performed by the scanner device, the software suite here presented

Figure 6: A digital model of the Minerva, simplified down to 1,034,029 triangles.

has been used to register and merge all the range maps in a single triangulated mesh. A complete digital model of he Minerva has been produced, reconstructed from the range maps using the *MeshMerge* module approach using a grid spacing of 0.57 mm. The final mesh reconstructed is rather big: more than 26 millions triangles. To improve usability, the original mesh was simplified using the *MeshSimplify* module. Figure 6 show a simplified models of the statue (around 1M faces).

Acknowledgements

This project has been developed in cooperation with the Centro di Restauro (Restoration Laboratory) of the Soprintendenza Archeologica Toscana in Florence and with the Istituto Nazionale di Ottica. In particular we would like to thank M. Cygielman, R. Giachetti, M. Giovanelli, G. Gori, L. Pezzati, R. Fontana and E. Pampaloni for the fruitful collaboration. We acknowledge the financial support of the Progetto Finalizzato "Beni Culturali" of the Italian National Research Council and the RIS+ Project of the Tuscany Regional Government.

References

[1] F. Bernardini and H. E. Rushmeier. 3D Model Acquisition. In *Eurographics 2000, State of the Art Reports Proceedings*, pages 41–62. Eurographics Association, August 24–25 2000.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 14(2):239–258, February 1992.

[3] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *International Journal of Image and Vision Computing*, 10(3):145–155, April 1992.

[4] P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno. External memory management and simplification of huge meshes. Technical Report B4-01, I.E.I. – C.N.R., Pisa, Italy, Gennaio 2001.

[5] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. *Computers And Graphics*, 22(1):37–54, 1998.

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Comp. Graph. Proc., Annual Conf. Series (Siggraph '96)*, pages 303–312. ACM Press, 1996.

[7] M. Garland. Multiresolution modeling: Survey & future opportunities. In *EUROGRAPHICS'99, State of the Art Report (STAR)*. Eurographics Association, Aire-la-Ville (CH), 1999.

[8] M. Garland and P.S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 209–216. Addison Wesley, August 1997.

[9] 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 *Comp. Graph. Proc., Annual Conf. Series (Siggraph '00)*, pages 131–144. ACM SIGGRAPH, Addison Wesley, July 24-28 2000.

[10] K. Pulli. Multiview registration for large datasets. In *Proc 2nd Int.l Conf. on 3D Digital Imaging and Modeling*, pages 160–168. IEEE, 1999.

[11] E. Puppo and R. Scopigno. Simplification, LOD, and Multiresolution - Principles and Applications. In *EUROGRAPHICS'97 Tutorial Notes (ISSN 1017-4656)*. Eurographics Association, Aire-la-Ville (CH), 1997 (PS97 TN4).

[12] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno. A low cost optical 3D scanner. Technical Report IEI-2001-TR-09, I.E.I. - C.N.R., Pisa, Italy, February 2001.

[13] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. The Marching Intersections algorithm for merging range images. Technical Report B4-61-00, I.E.I. - C.N.R., Pisa, Italy, June 2000.