OUT-OF-CORE MLS RECONSTRUCTION

Valentino Fiorin,
Paolo Cignoni,
Roberto Scopigno
ISTI CNR [*]

Abstract

The accurate and robust reconstruction of a digital representation using 3D scanning technologies is an important task in many application field. An important problem in this field is the integration of multiple range maps in a single polygonal surface. In this paper we describe a robust and effective implementation of a reconstruction algorithm based on implicit representation which, thanks to out-of-core techniques, allows to efficiently reconstruct large set of range maps.

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. 3D scanning devices are usually based on optical technology (laser or structured light) and use either the triangulation approach (small and medium scale objects) or the time of flight approach (large scale objects, e.g. architectures). The scanning of complex objects is therefore performed by taking a (usually large) set of partially overlapping range scans. The classical pipeline which characterizes a 3D scanning session is rather complex, involving many different operations that usually end with the final reconstruction of a single surface from a set of well aligned, cleaned range maps. As sketched in Section 2 many different algorithms have been proposed for the task of reconstructing a surface starting from a set of possibly noisy samples. In the rest of the paper we will present a practical and robust implementation of a MLS based out-of-core framework for the construction of a triangulated surface. We should remark that we are focusing on the field of Cultural Heritage application of 3D scanning technologies so some aspect of the reconstruction, notably the management of a very large set of range-maps, the possible presence of systematic errors in the set of samples (typically due to errors in the registration process) must be taken into account. In section 4 we will present some practical results and timing of the discussed approach, comparing it with the results obtained using a standard volumetric approach based on [12].

2 Related Work

Methods for surface reconstruction aim to find a mathematical discrete description of an object surface from its sampling. The need of requiring certain guarantees on the reconstructed surface along with the necessity of reducing the computational resources needed by the algorithms for giving such a description makes this problem an active subject of research. In this context, a lot of new solutions and approaches to this problem have been sketched out in the last years. Some of them use the topological informations inside the rangemaps in order to identify the surface: in [34] the rangemaps are cleaned up by the redundant informations in the overlapping area and then are merged through a triangulation process of the border belonging to adjacent rangemaps. Ignoring the topological informations inside the rangemaps but constraining the surface to interpolate the point cloud, Bernardini et al. [9] suggest a region-growing approach based on the ball-pivoting operation.

Different solutions have been formulated starting from the Delaunay complex associated to the point cloud. The alpha-shape approach [15] represent the first work in this direction; Bajaj et al. [8] extend the initial idea with heuristics aimed to capture concave features which the initial algorithm was not able to detect. Amenta et al. [4] [5] solve the same problem with the crust algorithm, which dynamically adapt the complexity of the surface to the curvature local factor.

The volumetric methods detect the surface distance at the corners of a regular grid, building up a signed distance function from the point cloud: Hoppe et al. [18] locally approximate the surface with tangent planes, used to compute the signed distance. In order to generate an explicit description of the reconstructed surface, generally the volumetric methods are combined with some polygonalization algorithms, such as the marching cubes [26] or related solutions [25] [21] [32] [20] [19] [17].

A relatively recent idea is to describe the surface of an object through a convenient set of functions. Carr and al. [11] [10] demonstrated the suitability of this approach to real problems combining the RBF representation with a greedy algorithm. Ohtake and al. [27] partition the point cloud with an adaptive octree and represent the portion of the surface contained inside each leaf with an opportune explicit function, whose weighted combination allow to generate a description of the complete surface implied by the point cloud. Using a blending function similar to the previ-

[*] Area della Ricerca CNR, Via Moruzzi 1, 56125 Pisa, ITALY. WWW: http://vcg.isti.cnr.it/ Email: {v.fiorin | p.cignoni | r.scopigno}@isti.cnr.it

ous, Shen et al. [33] associate to each point a different function so that also the gradient of the overall implicit function is constrained near the surface. Actually both approaches share the same mathematical framework, known as *moving least squares* (MLS). This method constitute the kernel of the projection operator originally proposed by Levin [24]: this operator is able to project a point near the point cloud on a continuous surface which minimize a local error measure formulated in terms of the least squares: the set of point which project onto themselves represent a surface generally called *point set surface* (PSS). These projection based approaches have been subject of investigation in the last years by numerous researchers by virtue of their many interesting properties, first of all the ability to automatically filter out the acquisition noise inside the rangemaps. Adamson and Alexa [3] provide a definition of smooth and manifold surface starting from a point cloud and then expand their work [2] [1] in order to combine the PSS definition with rendering and ray tracing methods. Amenta e Kil [7] propose a projector operator defined on *surfels*, namely point-normal pairs, and give an efficient procedure of minimization along with a proof of the convergence of projected point onto the PSS surface. Later the same authors [6] extend the domain of the projector operator and give two definitions of PSS surface with different ratio between the computational complexity and the precision of the sharp feature description. With reference to this last aspect, Reuter and al. [31] suggest a different projector operator based on the *Enriched Reproducing Kernel Particle Approximation* (ERKPA) method, aiming to limit the smoothing out of corners and edges in the PSS surface. This modified projection operator allows to correctly reconstruct surfaces with sharp features, but limited to those volume areas manually marked before by a user. Kolluri [22] proposes a different projector operator and shows theoretically its correctness under the assumption of a uniform sampling. Day e Sun [14] give a definition of MLS surface based on the *local feature size* and provide guarantees on the quality of the reconstructed surface under the hypothesis of an adaptive sampling. Fleishman et al. [16] adapt the *forward-search* paradigm to drive the MLS operator during the surface definition process: starting from a small set of samples not containing outliers, this paradigm progressively add new samples to the set provided that these new samples verify some statistical properties which monitor the quality of the surface. By means of this framework, they are able to manage the noise inside the dataset and also to detect sharp features and outliers.

3 Our reconstruction approach

Most of the reconstruction methods based on pointset describe the surface implicitly through an projector operator capable of projecting point laying near the surface exactly on the surface implied by the point cloud. Because we are interested in the construction of triangular meshes, we employ this projector operator to generate a signed scalar distance field: combining this scalar field with a polygonalization algorithm, we are able to generate polygonal surfaces. Before describing the details of our strategy, we give a brief review of the projector operator. Because a thorough survey is clearly beyond the scope of this paper, we address the interested reader to the papers in bibliography for a more exhaustive coverage of this subject.

3.1 MLS projector operator

Surfaces based on the Moving-Least Squares (MLS) projection approach have been introduced by Levin [24, 23] and subsequently used in many different works as robust approach to define a surface starting from a point set. MLS surfaces have been used, for example, in point-based graphics by Alexa et al. for high quality rendering of point-sampled surfaces [3] or to create approximations of point-based models with low sampling density in [30].

Given a point cloud near a surface S, the aim of the reconstruction algorithm is to build an implicit representation of the surface from which could be extracted an explicit description, generally a simplicial mesh. The implicit representation used by our algorithm belongs to the extremal surfaces family: this family includes the set of the surfaces which can be described by the interaction of two distinct functions, the energy function and the vector field.

In order to make the algorithm more robust, both functions are defined on the set of the points and the set of their normals. The normals either are directly available from the rangemaps, if the point cloud has been acquired by 3D scanning or by converting a polygonal model into a point-based one, or can be obtained by a preprocessing phase of the point cloud. The kernel of our algorithm is represented by the implicit representation described in [7], whose definition will now be briefly sketch out.

The definition of the vector field n follows the intuition that its valuation in a point $\mathbf{p}$ in the space must mimic the normal at the piece of surface closest to that point. Thus the vector field can be computed from the normals in the dataset in such a way that the direction associated to a point $\mathbf{p}$ is more influenced by the nearest points in the dataset. That condition is enforced through the following weighting function:

$$\vartheta_N(\mathbf{p},\mathbf{x}_i) = \frac{e^{-\frac{\|\mathbf{p}-\mathbf{x}_i\|^2}{h^2}}}{\sum_j e^{-\frac{\|\mathbf{p}-\mathbf{x}_j\|^2}{h^2}}}, \tag{1}$$

which is a normalized gaussian weighting function based on the distance. The vector field is then defined as the weighted average of the surfel normals, i.e.:

$$n_{\mathrm{PSS}}(\mathbf{p}) = \sum_i \vec{\mathbf{n}}_i \vartheta_N(\mathbf{p},\mathbf{x}_i).$$

The energy function also is formulated in a very intuitive way as the unsigned distance from the surface in terms of the surfel position and normal. Since we would like to give more importance to the points lying along the surfel normals, the energy function formulation make use of the Mahalanobis distance, a distance measure similar to the euclidian distance but with elliptical support rather than spherical:

$$\mathrm{dist}_M(\mathbf{p},\mathbf{x}_i,\vec{\mathbf{n}}_i) = \left((\mathbf{p}-\mathbf{x}_i)\cdot\vec{\mathbf{n}}_i\right)^2 + c\left\|(\mathbf{p}-\mathbf{x}_i) - \left((\mathbf{p}-\mathbf{x}_i)\cdot\vec{\mathbf{n}}_i\right)\vec{\mathbf{n}}_i\right\|^2,$$

where k is a scale factor witch affects the ellipsis shape: in particular when $k = 1$, the Mahalanobis distance is equivalent to the euclidian distance between the point $\mathbf{p}$ and the sample $\mathbf{q}_i$, whereas when $k = 0$ it corresponds to the distance from $\mathbf{p}$ to the plane through $\mathbf{x}_i$ with normal $\vec{\mathbf{n}}_i$. The resulting energy function is:

$$e_{\mathrm{PSS}}(\mathbf{p},\vec{\mathbf{n}}) = e_{\mathrm{PSS}}(\mathbf{p}) = \sum_i \mathrm{dist}_M(\mathbf{p},\mathbf{x}_i,\vec{\mathbf{n}}_i)\ \vartheta_N(\mathbf{p},\mathbf{x}_i).$$

Finally the implicit surface is determined by the set of points where the energy function e, evaluated along the direction of the vector field n, is minima, i.e.:

$$S = \left\{\mathbf{p}\,|\,\mathbf{p} \in \operatorname*{arglocalmin}_{\mathbf{q}\in\ell_{\mathbf{p},n_{\mathrm{PSS}}(\mathbf{p})}} e_{\mathrm{PSS}}\left(\mathbf{q}, n_{\mathrm{PSS}}(\mathbf{p})\right)\right\}.$$

3.2 Efficient management of large datasets

In order to be able to process large datasets, we have built our reconstruction algorithm on a out-of-core framework. With this framework we can manage the whole dataset and efficiently solve different types of spatial queries over huge set of range maps. The kernel of this framework is an octree which distributes, indexes and maintains out-of-core a dataset composed by a set of range maps.

When a range map is imported, its point cloud is sorted according to the Lebesgue space-filling curve [28, 29] relative to the octree deepest level. This sorting technique imposes a linear ordering to the point set that agrees with the octree cell distribution and that guarantees that all the points included in a particular cell of the octree (not only leaves) are contiguous. Therefore this particular arrangement allows us to refer the set of points included in a octree cell with just three references: the range map id and two references addressing the first and the one past the final element in the sorted dataset. This way we don't have to merge the set of range maps in a single point cloud and we can store in different files on secondary storage the images of the sorted range maps: so we are able to dynamically load and unload this temporary images depending on the portion of the surface the reconstruction algorithm is working on. This framework allows to not explicitly store anything in the octree nodes so we keep the octree structure light enough to maintain the whole octree resident in memory. Moreover the choice of keeping maps separate allow the easy inclusion and exclusion of single range maps, a feature useful when assembling the correct set of maps to be merged and when choosing the right reconstruction parameters.

3.3 Generating the triangular meshes

The projector operator does not directly define a scalar field; nevertheless, starting from the result generated by the projection procedure, we can obtain a signed distance function. As said before, the projector operator allows to project a point $\mathbf{p}$ on the nearest surface along the direction defined by the vector field at $\mathbf{p}$: i.e. the application of the projector operator to a point $\mathbf{p}$ gives back a point-direction pair $(\mathbf{p}^\star,\vec{\mathbf{n}}^\star)$ which respectively represent the projection of the point $\mathbf{p}$ on the surface and the direction it has been projected along: furthermore, as the projection procedure is iterative until the detection of a stationary point, the direction $\vec{\mathbf{n}}^\star$ is the one computed when the convergence is reached. The pair $(\mathbf{p}^\star,\vec{\mathbf{n}}^\star)$ allows us to define the plane P passing through $\mathbf{p}^\star$ and having direction $\vec{\mathbf{n}}^\star$ which for construction turns out the best local approximation of the nearest surface to $\mathbf{p}$. This approximation can then be exploited in order to identify a valuation of the distance between a point $\mathbf{p}$ and the surface: that distance is approximated with the distance between the point $\mathbf{p}$ itself and its projection on the plane P generated through the results of the application of the projection procedure to it. It's convenient to point out that the direction $\vec{\mathbf{n}}^\star$ computed by the projection operator isn't a undirected direction, because it's computed as weighted average of the point normals, and then that the computed distance is necessarily a signed distance.

We use the signed distance function just defined to generate the scalar field needed by the polygonalization algorithms in order to identify the surface. We sample the signed distance function at the corner of the leaves of the octree used to index the point cloud. In order to avoid the sampling of the signed distance function at all the corner of the grid induced by the octree, we suggest to limit the sampling of the scalar field only at the corners of the voxels which probably are crossed by the surface: these voxels are identified through spatial query to the octree aimed to identify the set of cells containing at least one point of the dataset. Most probably the voxels returned by such a query will contain surface pieces: for each of these voxels the signed distance function is sampled at its corners and, if a sign change is detected, the voxel is processed by the polygonalization algorithm in order to obtain the triangle set approximating the surface.

3.4 Hole Filling

As defined above, the reconstruction algorithm searches for the surface only on those voxels containing some point of the dataset: even though this solution is essential in order to avoid the sampling on the whole scalar field, however the mesh reconstructed this way can have discontinuities and unconnected components, since it can happen that the

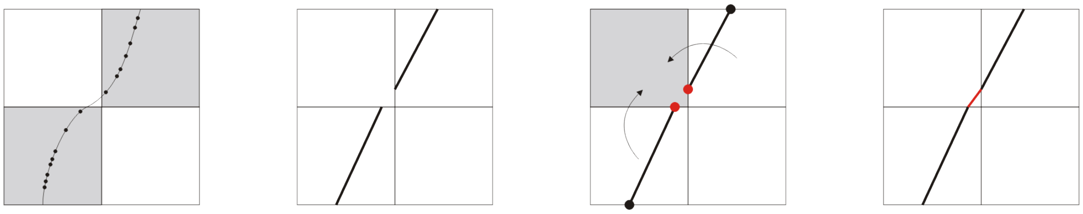

Figure 1. Due to the sample distribution as for the octree cells, if the algorithm visit those cells where is contained at least one sample point (cells in grey), then the reconstructed surface will be composed of two distinct components.

surface goes across voxels not containing any point of the dataset 1. The solution we propose to this problem is based on the possibility to encode in each voxel if it has been visited, if it has to be visited or if it don't belong to the set of voxels which will be visited. The surface reconstruction is namely performed in a fixed number of *expansion steps*, i.e. a maximum number of iterations specified by the user: during the first iteration, the algorithm analyzes only the voxels returned by the octree in response to the spatial query, while in the next steps the analysis is expanded to those voxel marked in the immediately previous iteration. The voxels which are to be visited as well as the voxel already visited are maintained in two different sets: if, performing the triangulation of the surface crossing a given voxel a intercept is computed laying on a edge shared by two voxels which not belong neither to the set of the already visited voxels nor to the set of the voxels to be visited at the current iteration, then all the voxels which share that edge are inserted into the set of voxels to be analyzed at the next step. In reference to the figure 1, during the first step, the algorithm only visits the cells containing at least one point of the dataset, i.e. the cells colored in gray: when, during this iteration, the intercept in red are computed, as these lay on a edge shared with a cell not already visited, this cell is marked in order to guarantee the reconstruction of a continuous surface.

This approach allows to build closed surfaces even for limited value of the expansion steps. Nevertheless using this extension without any precaution can lead to the generation of artefact absent in the original surface: this technique has indeed the tendency to propagate the analysis to those cells excessively distant from the point cloud. The remedy we propose is to constrain the propagation only to those cells whom distance from the point cloud isn't greater than k times the leaf diagonal, where k is another parameter chosen by the user. This simple adjustment don't preclude the algorithm ability to generate closed surfaces and in addition allows the surface to propagate in those regions difficult to sample due to the occlusions on the real object.

4 Results

Most of the dataset selected for the experiments presented in this section have been acquired with a laser triangulation scanner (Konica Minolta VI910), which produces very dense range maps with a low sampling error. These datasets, presented in Table 1, encode the digital shape of: the "Chief Warrior", a nuragic small bronze of the Bronze Age (Archeological Museum, Cagliari); the bust of the "Minerva of Arezzo", a full body bronze statue (Archeological Museum, Florence); and the bust of "Ippolita Sforza", the sole plaster cast of a Renaissance work by Francesco Laurana. In order to test the effectiveness of our algorithm in view of a more noisy dataset and to stress its out-of-core extension, we included also a dataset representing a architectural model, the "Pisa Cathedral", acquired with time-of-fly technology (Leica Geosystem HDS3000). The size and resolution of all dataset are reported in Table 1.

For testing purposes we have reconstructed all the dataset with **OM**, the reconstruction tool presented in previous sections, and with **PlyMC**, another reconstruction program developed at the Visual Computing Lab. PlyMC is an implementation of the classical volumetric approach based on discrete distance field and MC-like reconstruction. The distance field is reconstructed from range maps following the approach proposed by Curless and Levoy [12]. To fill unsampled regions, PlyMC adopts the volumetric diffusion technique proposed by Davis et al. [13].

We compare visually the output generated by the two programs in the following figures, in order to provide a comparison of the improved reconstruction accuracy provided by OM. All the test have been executed on a PC with an Intel Pentium IV and 2GB of RAM.

The algorithm performs very well in practice: all the dataset have been successfully processed and the reconstructed surfaces preserved the details of the scanned object. The ability of our algorithm to correctly reconstruct regions rich of details are exemplified in Figures 2 and 3:

Dataset	Range map no.	Samples no.	Size
Ippolita Sforza's bust	29	5 265 736	191 MB
Chief Warrior	102	9 864 132	756 MB
bust of Minerva of Arezzo	80	11 543 578	1.13 GB
Pisa Cathedral	180	154 247 414	5.6 GB

Table 1. The datasets used in our tests.

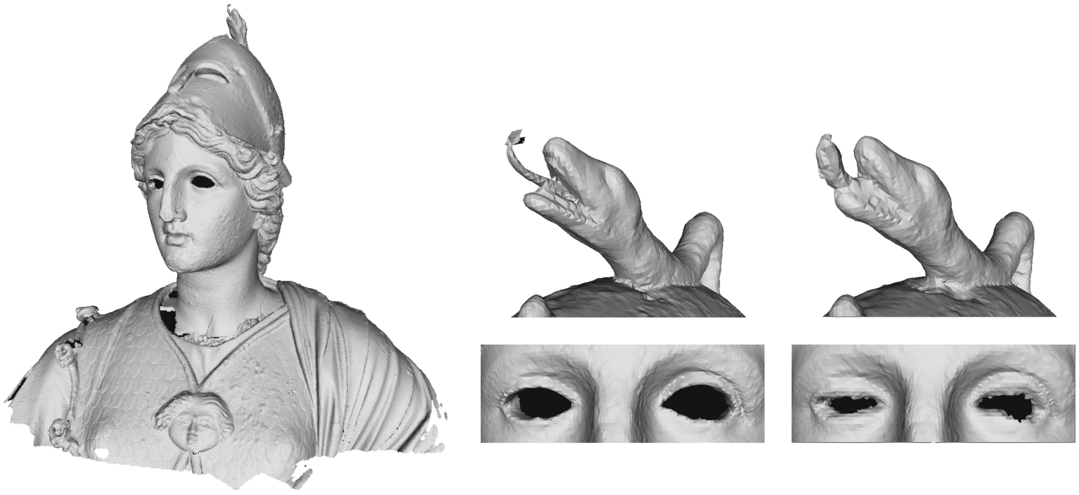

Figure 2. Reconstruction of the bust of the Minerva of Arezzo: on the left the reconstructed model and at the center and on the right the details from the helmet and the eye hole as reconstructed by OM and PlyMC respectively.

Dataset	Voxel sz	Tool	Face no.	Time (h:m:s)
Ippolita Sforza's	1.14 mm	OM	1 085 808	3:00
bust		PlyMC	1 163 249	5:42
Chief Warrior	0.46 mm	OM	2 067 672	10:00
		PlyMC	2 032 143	16:42
Bust of Minerva	1.15 mm	OM	1 015 070	7:37
of Arezzo		PlyMC	859 334	9:36
Pisa Cathedral	7 mm	OM	214 706 141	63:06:07

Table 2. The results of the reconstruction tests.

as we can see from these screenshots, the models reconstructed with the new MLS-based solution preserve better the shape and the details of the scanned object (see Figure 2, at the centre), especially in regions having a thin section or high curvature; moreover, results do not present artifacts or wrinkles where the original surface was smooth (see Figure 3 in the middle).

Running times of the MLS-based solution (OM) are usually approximately 30% shorter than the ones of the PlyMC solution (and please consider that in the PlyMC run we perform volumetric diffusion, which has a substantial impact in running times).

The test have highlighted a certain sensitivity of the MLS-based system to possible residual misalignment errors. However, when the order of magnitude of the alignment error is comparable with the sampling rate (intersampling distance on the surface), a correct surface is reconstructed by simply increasing slightly the number of points used in the projection procedure. This is the case depicted in Figure 4: the irregularities on the left figure are due to a misalignment between the range maps; at the center we show the surface correctly reconstructed by OM after doubling the number of points used during the projection procedure.

Some residual misalignment errors were perceptible also in the reconstruction of the Pisa Cathedral, a dataset acquired using a time-of-flight scanner which produces more noisy range maps than laser triangulation. Therefore, the OM reconstruction algorithm was able to process the whole dataset (more than 5 GB in input) and to produce, as Figure 6 illustrates, a very detailed model.

5 Conclusion

We have presented and tested an algorithm for the efficient out of core reconstruction of very large set of range maps. The presented technique is based on the explicit polygonalization of a MLS surface representation. The polygonalization algorithm exploits an out-of-core octree indexing scheme that allows efficient retrieval of closest point queries, the critical core of all MLS based interpolation techniques. In the results section we have shown how the presented approach allows the reconstruction of huge datasets composed by hundreds of millions of primitives and it allows to finely control the amount of smoothing and hole-filling introduced during the reconstruction process.

Acknowledgements We acknowledge the financial support of some current projects: the EU Network of Excellence IST-2002-507382 “EPOCH”, projects “Fruizione - Tecniche di supporto e modalita’ innovative” of the CNR Patrimonio Culturale Dept. and “Media Multidimensionali” of the CNR ICT Dept. But the work presented is also the outcome of several prior projects, such as projects EU IST-2001-32641 “ViHAP3D and Regione Toscana “RIS+.

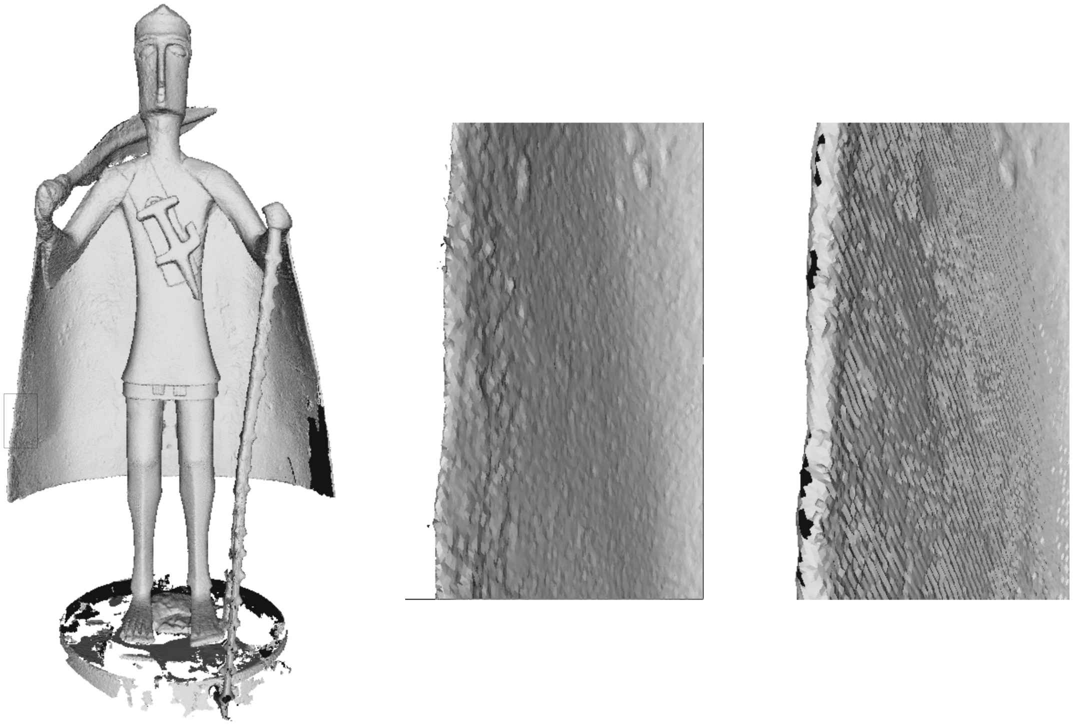

Figure 3. Reconstruction of the Chief Warrior: despite of the thinness of the cloak, both sides of the surface have been correctly reconstructed. At the center and on the right a detail from the cloak of the models reconstructed using OM and PlyMC respectively.

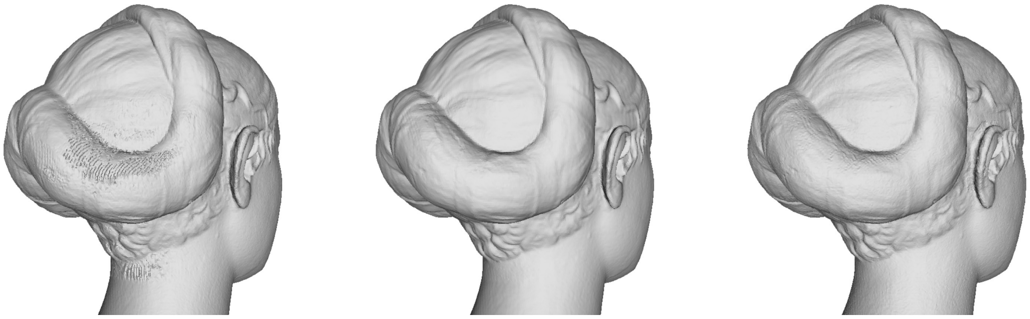

Figure 4. Reconstruction of the Ippolita Sforza's bust: the irregularities on the left are caused by some residual misalignment error; on the center the surface correctly reconstructed using different projection operator parameters. On the right the surface reconstructed using PlyMC.

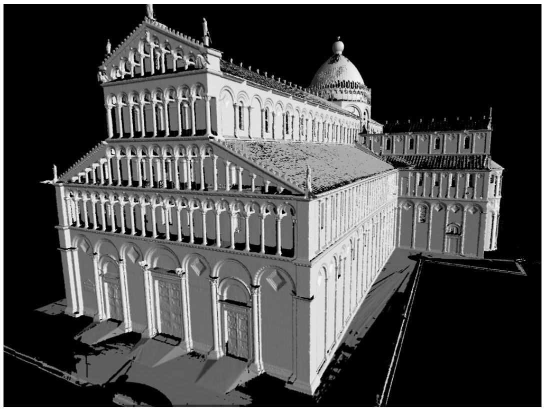

Figure 5. Reconstruction of the Pisa Cathedral: this is the largest model we have reconstructed so far using OM; once reconstructed using a cell width of 7 mm, it produces a surface encoded by 214 million of faces.

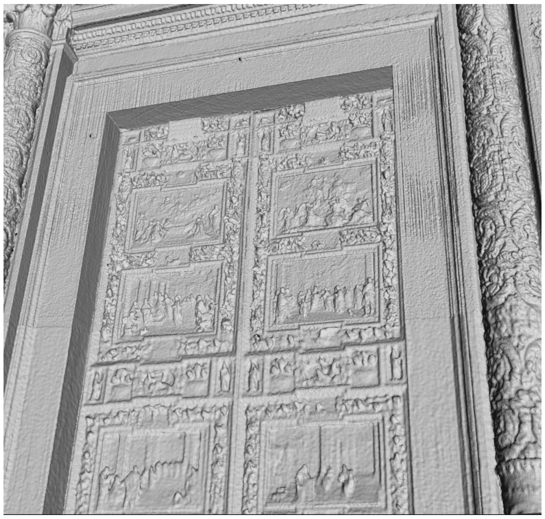

Figure 6. Detail from the reconstruction of the Pisa Cathedal: the main door.

References

[1] Anders Adamson and Marc Alexa. Approximating and intersecting surfaces from points, 2003.

[2] Anders Adamson and Marc Alexa. Ray tracing point set surfaces. In *SMI '03: Proceedings of the Shape Modeling International 2003*, page 272, Washington, DC, USA, 2003. IEEE Computer Society.

[3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, January 2003.

[4] Nina Amenta. The crust algorithm for 3D surface reconstruction from scattered points. In *Proceedings of the Conference on Computational Geometry (SCG '99)*, pages 423–424, New York, N.Y., June 13–16 1999. ACM Press.

[5] Nina Amenta and Marshall W. Bern. Surface reconstruction by voronoi filtering. In *Symposium on Computational Geometry*, pages 39–48, 1998.

[6] Nina Amenta and Yong J Kil. The domain of a point set surfaces. *Eurographics Symposium on Point-based Graphics*, 1(1):139–147, June 2004.

[7] Nina Amenta and Yong Joo Kil. Defining point-set surfaces. *ACM Trans. Graph.*, 23(3):264–270, 2004.

[8] Chandrajit L. Bajaj, Fausto Bernardini, and Guoliang Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. *Computer Graphics*, 29(Annual Conference Series):109–118, 1995.

[9] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, Oct.-Dec. 1999.

[10] J. C. Carr, R. K. Beatson, B. C. McCallum, W. R. Fright, T. J. McLennan, and T. J. Mitchell. Smooth surface reconstruction from noisy range data. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 119–ff, New York, NY, USA, 2003. ACM Press.

[11] Jonathan C. Carr, Richard K. Beatson, Jon B. Cherrie, Tim J. Mitchell, W. Richard Fright, Bruce C. McCallum, and Tim R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *SIGGRAPH 2001, Computer Graphics Proceedings*, Annual Conference Series, pages 67–76. ACM Press / ACM SIGGRAPH, 2001.

[12] 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.

[13] J. Davis, S. Marshner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *First Int. Symp. on 3D Data Processing, Visualization and Transmission (3DPVT'02)*, pages 428–438. IEEE Comp. Soc., 2002.

[14] Tamal K. Dey and Jian Sun. . an adaptive mls surface for reconstruction with guarantees. In *Symposium on Geometry Processing*, pages 43–52, 2005.

[15] Herbert Edelsbrunner and Ernst P. Mücke. Three-Dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, January 1994. ISSN 0730-0301.

[16] Shachar Fleishman, Daniel Cohen-Or, and Cl&#225;udio T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, 2005.

[17] Chien-Chang Ho, Fu-Che Wu, Bing-Yu Chen, Yung-Yu Chuang, and Ming Ouhyoung. Cubical marching squares: Adaptive feature preserving surface extraction from volume data. *Computer Graphics Forum*, 24(3):537–545, August 2005. Special issue: Proceedings of EUROGRAPHICS 2005.

[18] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26(2):71–78, 1992.

[19] Scott Schaefer Joe. Dual contouring: The secret sauce.

[20] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. In *Siggraph 2002, Computer Graphics Proceedings*, pages 339–346. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2002.

[21] L.P. Kobbelt, M. Botsch, U. Schwanecke, and H.P. Seidel. Feature-sensitive surface extraction from volume data. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 57–66. ACM Press / ACM SIGGRAPH, 2001.

[22] Ravikrishna Kolluri. Provably good moving least squares. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, pages 1008–1018, August 2005.

[23] D. Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, pages 37–49, 2003.

[24] David Levin. The approximation power of moving least-squares. *Mathematics of Computation*, 67(224):1517–1531, 1998.

[25] Thomas Lewiner, Hélio Lopes, Antônio Wilson Vieira, and Geovan Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of Computer Graphics*, 8(2):1–15, 2003.

[26] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM Computer Graphics (SIGGRAPH 87 Proceedings)*, volume 21, pages 163–170, 1987.

[27] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, July 2003.

[28] V. Pascucci. Multi-resolution indexing for hierarchical out-of-core traversal of rectilinear grids, 2000.

[29] Valerio Pascucci and Randall J. Frank. Global static indexing for real-time exploration of very large regular grids. In *Supercomputing '01: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)*, pages 2–2, New York, NY, USA, 2001. ACM Press.

[30] M. Pauly, M. Gross, and L. Kobbelt. Efficient simplification of point-sampled surfaces, 2002.

[31] Patrick Reuter, Pierre Joyot, Jean Trunzler, Tamy Boubekeur, and Christophe Schlick. Surface reconstruction with enriched reproducing kernel particle approximation. In *Proceedings of the IEEE/Eurographics Symposium on Point-Based Graphics*, pages 79–87. Eurographics/IEEE Computer Society, 2005.

[32] Scott Schaefer and Joe Warren. Dual marching cubes: Primal contouring of dual grids. In *PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04)*, pages 70–76, Washington, DC, USA, 2004. IEEE Computer Society.

[33] Chen Shen, James F. O'Brien, and Jonathan R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. *ACM Trans. Graph.*, 23(3):896–904, 2004.

[34] G. Turk and M. Levoy. Zippered polygon meshes from range images. *ACM Computer Graphics*, 28(3):311–318, 1994.