A Hybrid System for Three-Dimensional Objects Reconstruction from Point-Clouds Based on Ball Pivoting Algorithm and Radial Basis Functions

ASHRAF HUSSEIN
Scientific Computing Department,
Faculty of Computer and Information Sciences, Ain Shams University,
Abbassia, 11566, Cairo, Egypt.

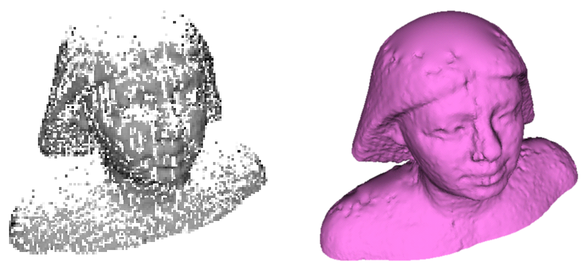
Abstract:- In this paper, a new hybrid system for 3D objects reconstruction is presented. This system is based on the well-known Ball Pivoting Algorithm (BPA) of Bernardini et al. combined with the Radial Basis Function (RBF) model. The BPA can reconstruct surfaces from very large data sets in a little execution time, using small amount of memory, and exhibits linear complexity but it produces holes in case of low sampling density besides it is not capable to fill mis-registration domains. So, the BPA is coupled with the RBF in an integrated system to gain the edge of the RBF as a powerful interpolator for generating surface vertices within severe holes and/or mis-registration domains. Performance and accuracy of the system components are investigated through reconstructing benchmark objects. The results obtained compare favorably with other eminent algorithms. Finally, the proposed system exhibited high robustness in reconstructing objects from real large clouds with non-uniform sampling and mis-registrations.

Key-Words: - Object modeling, computational geometry, surface reconstruction, and surface and object representations.

1. Introduction

There are many applications that rely on building accurate models of real-world objects such as sculptures, damaged machine parts, archaeological artifacts, and terrain. Techniques for digitizing objects include laser range finding, mechanical touch probes, and computer vision techniques such as depth from stereo. Some of these techniques can yield millions of 3D point locations on the object that is being digitized. Once these points have been collected, it is a non-trivial task to build a surface representation that is faithful to the collected data. Some of the desirable properties of a surface reconstruction method include speed, low memory overhead, faithful reproduction of sharp features, and robustness in the presence of holes and low sampling density. Figure (1) illustrates a typical example for object reconstruction problem. Few techniques are available for automatically modeling this type of data.

Mathematically, the problem can be expressed as follows, given a set of N points $P=\{p_1,\ldots,p_N\}$, which are sampled from some unknown smooth surface U, find a mesh M that approximates the surface U, i.e. the mesh and the surface are near everywhere [15]. The output is initially some mesh (mostly a triangulation) that has the same general shape of the original surface. If the generated mesh connects the original input points, then it interpolates the surface. Alternatively, the mesh may connect generated points other than those of the input set. In the latter case, the mesh is said to approximate the surface. A common classification criterion of surface reconstruction techniques is to categorize the technique as either a surface interpolation algorithm or a surface approximation one [17].



(a) (b)

Fig. 1: Example of a typical object reconstruction problem (a) Raw data of a statue with native holes and mis-registrations. (b) Reconstructed object using the present system.

One of the first surface interpolation algorithms is the one presented by Edelsbrunner and Mücke in [11], which uses the heuristic alpha-shapes. Amenta and Bern [1] used the properties of the medial axis and Voronoi diagram to introduce the Crust algorithm, which was the first guaranteed surface reconstruction algorithm. They extracted an approximated triangulated surface from the Delaunay triangulation of the input set. Amenta et al. [2] described the co-cone term and used it in their Cocone algorithm. The Cocone computes the Delaunay triangulation of the input set and then selects candidate triangles to construct the surface. Dey and Goswami [9] modified the Cocone into a new algorithm, the Tight Cocone, to solve problem instances of specific domains that require hole-free surfaces. Kolluri et al. [16] introduced a noise-resistant algorithm for watertight surfaces. The latter algorithm uses a variant of spectral graph partitioning to select triangles from the Delaunay triangulation to construct the surface. All the aforementioned algorithms can be described as Voronoi based algorithms and they generally suffer from slow performance and difficulty in handling large data. To overcome this problem, Dey et al. [10] partitioned the entire sample space into smaller clusters using octree subdivision. Then, they applied the Cocone algorithm on each of these clusters separately, and they called this algorithm the Super Cocone. An interesting interpolation non-Voronoi based algorithm is the Ball Pivoting algorithm, presented by Bernardini et al. [5]. This algorithm incrementally reconstructs surfaces by using a rolling ball to construct triangles one by one and append them to the surface.

Typical examples of surface approximation algorithms are the algorithms presented by Hoppe et al. [15] and by Curless and Levoy [8]. They used the normal orientations to estimate the tangent planes and approximate the local surface. Another surface approximation algorithm was introduced by Boissonat and Cazals [4] that is based on natural neighbors interpolation and represents the surface implicitly as a zero-set of some pseudo-distance function. Gopi et al. [13] suggested an algorithm that uses a plane projection technique where the neighbors of a sample point is computed by projecting its close points onto its tangent plane. Amenta et al. [3] suggested using the power diagram (dual of weighted Delaunay triangulation) in their Power Crust algorithm. Although the Power Crust algorithm is a Voronoi based algorithm, but -unlike the algorithms belonging to this family- it is classified as a surface approximation algorithm. Another family of surface approximation algorithms uses the implicit surface approach, for example [6], [20] and [7], which is suitable for reconstructing point clouds with local problems like holes, low sampling density, and mis-registration but not suitable for processing large clouds. To adapt these algorithms for large clouds, local support or partitioning based solutions are used (for example [18], [19] and [21]).

Still, mesh interpolation based methods are preferable for surface reconstruction than those based on mesh approximation because the former give better accuracy and have better performance from both computational time and memory usage points of view. On the other hand, mesh approximation based methods that utilize implicit surface approach have the edge of robustness in reconstructing surfaces from cloud of points natively have holes, low sampling density and mis-registration. So, gaining the advantages of the two types of reconstruction methods can be obtained by integrating methods from both of them.

In this work, a hybrid system is developed by combining two well-known methods in order to obtain a reconstruction scheme for large data sets in the presence of low sampling density and/or mis-registration domains. The Ball Pivoting algorithm [5] is used as an efficient tool for surface reconstruction while the Radial Basis Functions (RBF) [6] is used as a powerful interpolation tool.

The paper is organized as follows: section 2 provides the theoretical backgrounds of the technique. In section 3, the implementation is presented showing the practical choices to obtain an easy, efficient and robust reconstruction algorithm. Section 4 presents a study for the performance and accuracy of the present system and the application of reconstructing some realistic data that have native holes and mis-registration domains, and in section 5 the work is concluded.

2. Theoretical backgrounds

2.1. Ball pivoting algorithm

The approach of Bernardini, Mittleman and Rushmeier [5] imitates the ball eraser used in α-shapes. Assume that the sampled data set P is dense enough that a ρ-ball, a ball of radius ρ, can not pass through the surface without touching sample points. The algorithm starts by placing ρ-ball in contact with three sample points. Keeping contact with two of these initial points, the ball is "pivoting" until it touches another point. The pivoting operation is depicted in Figure (2).

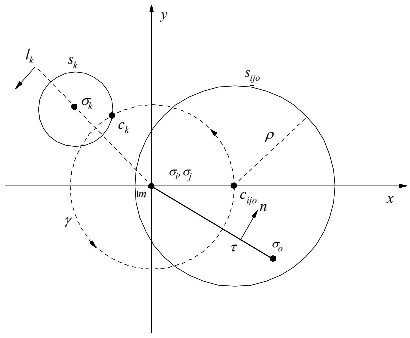


Fig. 2: The ball pivoting operation. The ball of radius ρ "sitting" on the triangle τ formed from the vertices σ_i, σ_j, σ_0 is pivoting around the edge m here perpendicular to the image. The center of the ball c_{ij0} rotates around the edge m and describes a circular trajectory γ. The radius of the trajectory is $\|c_{ij0}-m\|$. While pivoting the ball hits the point σ_k . The circle s_k is a intersection of ρ-ball centered at σ_k with z = 0. The circle s_{ij0} is the intersection of the pivoting ball with z = 0. (image courtesy of Fausto Bernardini [5]).

The ρ-ball is pivoted around each edge of the current mesh boundary. Triplets of points that the ball contacts form new triangles. The set of triangles formed while the ρ-ball "walks" on the surface form the interpolating mesh. The ball pivoting algorithm (BPA) is closely related to α-shapes. In fact, every triangle τ computed by the ρ-ball walk obviously has an empty smallest open ball b_τ whose radius is less than ρ. Thus, the BPA computes a subset of the 2-faces of the ρ-shape of P. These faces are also a subset of the 2-skeleton of the three-dimensional Delaunay triangulation of the point set. The surface reconstructed by the BPA retains some of the qualities of alpha-shapes: It has provable reconstruction guarantees under certain sampling assumptions and an intuitively simple geometric meaning.

The input data points are augmented with approximate surface normals computed from the range maps. The surface normals are used to disambiguate cases that occur when dealing with missing or noisy data. Areas of density higher than ρ present no problem to the algorithm, but missing points create holes that cannot be filled by the pivoting ball. Any post-process hole-filling algorithm could be employed; in particular, BPA can be applied multiple times with increasing ball radii. To handle possible ambiguities, the normals are used as stated above. When pivoting around a boundary edge, the ball can touch an unused point lying close to the surface. Therefore a triangle is rejected if the dot product of the triangle normal with the surface normal (in the vertex) is negative, see Figure (3).

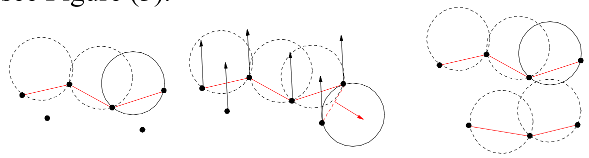


Fig. 3: The normal check (a 2D case). The rightmost point is boundary and the ball pivots around its edge and touches a hidden point, but the dot product of the triangle normal with the vertex normal is negative, so the triangle is rejected. (image courtesy of Fausto Bernardini [5])

This algorithm is very efficient because it exhibits linear time and storage requirements. However its disadvantages are the selection of the radius ρ of the ρ-ball and incapability to handle data that have native problems like low sampling density, holes and mis-registrations.

2.2. Radial basis functions

Given the set of N distinct points $P = \{p_1,\ldots,p_N\}$ of dimension d: $p_k \in \mathbb{R}^d$, and the set of values $\{h_1,\ldots,h_N\}$, we want to find a function f: $\mathbb{R}^d \rightarrow \mathbb{R}$ with

$$\forall i \qquad f(p_i) = h_i \tag{1}$$

In order to obtain a radial basis function reconstruction of the point set P, a function f satisfying the equation

$$f(p) = \sum_{i=1}^{k} \omega_i \phi(\|p, p_i\|) + \pi(p) \tag{2}$$

has to be found. We denote here $\|p_i, p_j\|$ the Euclidean distance, ω_i the weights, ϕ: $\mathbb{R} \rightarrow \mathbb{R}$ a basis function, and π a polynomial of degree m depending on the choice of ϕ. $\pi(p) = \sum c_i \pi_i(p)$ with $\{\pi_\alpha\}_{\alpha=1}^{Q}$ a basis in the d-dimensional null space containing all real-valued polynomials in d variables and of order at most m, hence $Q = \begin{pmatrix} m+d \\ d \end{pmatrix}$.

The basis function ϕ has to be conditionally positive definite [12], and some popular choices proposed in the literature are shown below:

biharmonic	$\phi(r) = r$ with π of degree 1	(3)
pseudo-cubic	$\phi(r) = r^3$ with π of degree 1	(4)
triharmonic	$\phi(r) = r^3$ with π of degree 2	(5)

The biharmonic and triharmonic splines are known as the “smoothest” interpolators in the sense that they minimize certain energy functions and interpolate the data [6]. So, the biharmonic spline is used in the present work. Given a set of nodes $\{p_i\}_{i=1}^{N} \subset \mathbb{R}^3$ and a set of function values $\{h_i\}_{i=1}^{N} \subset \mathbb{R}$, the biharmonic RBF $f(p)$ satisfies the interpolation conditions $f(p_i) = h_i$ and minimizes

$$\|f\|^2 = \int_{\mathbf{R}^3} \left(\frac{\partial^2 f(p)}{\partial x^2}\right)^2 + \left(\frac{\partial^2 f(p)}{\partial y^2}\right)^2 + \left(\frac{\partial^2 f(p)}{\partial z^2}\right)^2 + 2\left(\frac{\partial^2 f(p)}{\partial x \partial y}\right)^2 + 2\left(\frac{\partial^2 f(p)}{\partial x \partial z}\right)^2 + 2\left(\frac{\partial^2 f(p)}{\partial y \partial z}\right)^2 dx \tag{6}$$

$\|f\|^2$ is a measure of the energy in the second derivative of f.

As we have an under-determined system with $N+Q$ unknowns (ω and c) and N equations, so-called natural additional constraints for the coefficients ω are added, so that

$$\sum_i \omega_i c_1 = \sum_i \omega_i c_2 = ... = \sum_i \omega_i c_Q = 0 \tag{7}$$

The equations (1), (2), and (7) determine the following linear system:

$$Ax = b \tag{8}$$

$$A = \begin{bmatrix} \Phi & P^T \\ P & 0 \end{bmatrix} \tag{9}$$

$$\Phi_{ij} = \left[\phi\left(\|p_i, p_j\|\right)\big|_{\substack{i=1...N \\ j=1...N}} \right]$$

$$P = \left[\pi_\alpha(p_i)\big|_{\substack{i=1...N \\ \alpha=1...Q}} \right]$$

$$x = \left[\omega_1, \omega_2, ..., \omega_N, c_1, c_2, ..., c_Q\right]^T$$

$$b = \left[h_1, h_2, ..., h_N, \underbrace{0, 0, ..., 0}_{Q \text{ times}} \right]^T \tag{10}$$

The solution vector x is composed of the weights ω_i and the polynomial coefficients c_i for equation (2) and represents a solution of the interpolation problem given by (1).

3. Reconstruction scheme

3.1. Hybrid reconstruction

To overcome the disadvantages of the Ball Pivoting Algorithm, a hybrid system is developed by combining BPA and RBF and works as shown in Figure (4). The system calls the BPA iteratively by defining the triangulation domain. The RBF solution vectors (which contain the weights ω_i and the polynomial coefficients c_i) are computed once after the identification of holes and/or mis-registration domains and their neighborhood zones from the initial reconstructed surface obtained by the BPA or given as an input to the present system. Each iteration, holes are identified (which lie within that identified previously) then the RBF is employed as interpolator to uniformly generate surface points within the holes and the BPA is used to triangulate the generated points. The process of calling BPA and generating sampled points using RBF surface interpolator is repeated until reaching a perfect surface.

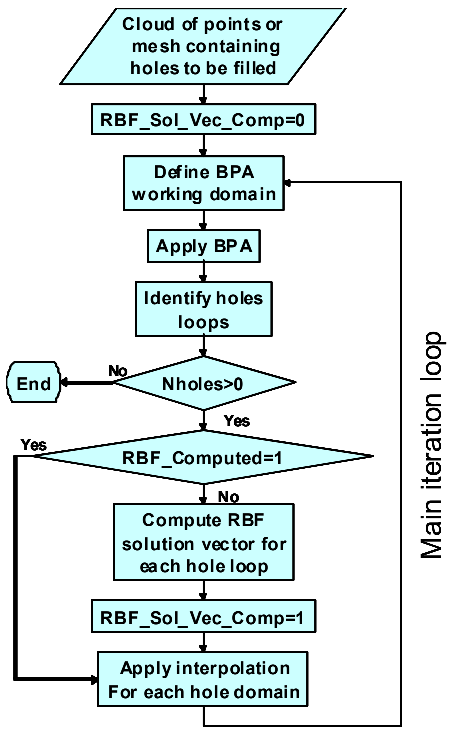


Fig. 4: The hybrid system for 3D object reconstruction

3.2. Hole or mis-registration identification

The application searches for every edge which lies in only one triangle and constructs a list of these edges. Starting with a seed vertex which lies on any edge within the constructed list, the application can trace a series of edges until it identifies a closed loop defining a hole. The identified loop of edges is removed from the list. The process of tracing and removing edges is repeated until identifying all the holes.

3.3. RBF computations

After the identification of the closed loops of holes and/or mis-registration domains, the RBF computations are performed for each of these loops. For each loop, the neighborhood zone depth should be defined priory (the zone depth is a user selected value as a ratio of the hole loop largest diagonal). The vertices within this zone are remarked by a simple tracing process through edges until reaching the predefined zone depth as shown in Figure (5). The remarked vertices are employed in the RBF computations to get the solution vector (which contains the weights and the polynomial coefficients) using equation (8). The z-components of these vertices will be considered as the function values in the right hand side of the equation. This process is applied once to get the solution vector for each hole or mis-registration domain. The interpolation process is applied to fill each hole or mis-registration domain by generating surface vertices using different x and y coordinates values, with nearly uniform distribution, then the corresponding z-coordinates are obtained by the same equation using the computed solution vector.

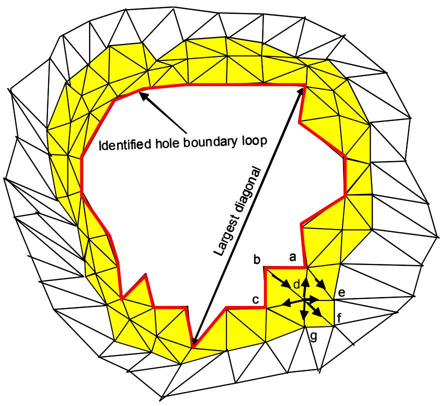


Fig. 5: Tracing of neighborhood zone vertices

Selecting x and y coordinates values is the key parameter in generating surface vertices with nearly uniform distribution within each hole or mis-registration domain. If the distribution of the generated surface vertices is not sufficiently uniform, more than one iteration will be needed until reaching perfect surface as mentioned in Figure (4). In such case, the interpolation process will be repeated to generate surface vertices within the identified holes using the same solution vector computed at the first iteration.

4. Applications and results

All results presented in this section were performed on Pentium IV 2.4 GHz PC computing platform with 512 MB RAM running windows XP. Source codes are developed using Microsoft visual C/C++ version 6.0. Also, Intel® Math Kernel Library was employed to improve the performance of matrix operations.

4.1. BPA performance verification

The performance of the BPA is verified against other eminent algorithms, namely Crust, Power Crust and Tight Cocone, in reconstructing bench mark objects of [14] with different resolutions. These algorithms are developed on the same platform and using the same programming tools to get proper conclusions. Figures (6.a) to (6.c) show the variation of computational time versus model size for the three selected objects for comparison (Bunny, Dragon and Happy Buddha) while Figures (6.d) to (6.f) show the corresponding variation of memory usage. All the algorithms exhibited linear performance independent of the object shape but BPA bested others from both computational time and memory usage points of view.

Figure (7) shows the reconstructed meshes using the different algorithms for the 1889 vertices Bunny model. The Crust output (Figure 7.a) is good but it is not a manifold. Besides being watertight, the Power Crust output (Figure 7.b) may be considered the best, except that the mesh is very dense and not triangular. The Tight Cocone output (Figure 7.c) is also watertight, but shows a problem at the left ear end (a triangle is not connected to the rest of the ear). The Ball Pivoting output (Figure 7.d) shows also problems at both of ears which can be considered as holes in the obtained mesh.

4.2. Hybrid system application and accuracy assessment

The system is tested via the reconstruction of the Liondog object used by Ohtake et al. [18], Figure (8.a), after making an artificial hole in its head to assess the accuracy of the present system in filling holes and mis-registration domains. The original model contains 24930 uniformly sampled points. The mesh is reconstructed initially by applying the BPA using a single ball (its radius is estimated as the average sampling distance) in 8.2 secs. Then, the artificial hole is successfully identified by its boundary loop as shown in Figure (8.b). Different values for the hole neighborhood zone depth are considered. The values considered are 50%, 70% and 100% of the hole loop largest diagonal (HLD). The RBF solution vector is computed for the three cases in 9, 16 and 27 secs using the direct method [6]. Finally, the iterative process between the BPA and RBF is applied until achieving a perfect surface. Only one iteration is performed in 6 secs to get surface with final quality as shown in the same figure. The accuracy of the present system in filling holes and mis-registration domains is assessed by comparing the reconstructed surface for the artificial hole and the original surface (for each of the considered values of the hole-neighborhood zone depth). The difference between the two surfaces at each point with the same x and y coordinates is considered as the difference in the magnitude of the z-coordinate. The contours representing the distribution of the difference between surfaces are shown in Figures (8.c) to (8.e) for the three considered values. It is found that increasing the hole-neighborhood zone depth decreasing the accuracy of filling and increasing the smoothness.

4.3. Reconstruction of realistic data

The robustness of the present system in reconstruction of 3D objects from large cloud of points that have native holes and/or mis-registration is confirmed using a realistic cloud of points for a statue of an ancient Egyptian man. The model contains 262,461 points with many mis-registration domains in its head, face, shoulders, hands and the base as shown in Figure (9.a). The mesh is reconstructed initially by applying the BPA using single ball in 91 secs. Then, holes and mis-registration domains are identified. The value of the neighborhood zone depth for the holes and mis-registration domains is taken as 50% of HLD of the corresponding holes or mis-registration domains. The RBF solution vectors for all the identified holes are computed in 194 secs. Then, three iterations are performed in 89 secs to get surface with final quality. Figures (9.b), (9.c) and (9.d) compare the reconstructed object using the present system against the results of the reconstruction using the Tight Cocone algorithm and the multi-level partition of unity implicits of [18]. It can be shown that the Tight Cocone, Figure (9.b), exhibited flat filling for the mis-registration domains while the reconstructed object using the multi-level partition of unity implicits of [18] suffers from spikes in the mis-registration domains besides losses in the object details as shown in Figure (9.c). On the other hand, the present system, Figure (9.d), exhibited curved filling for the mis-registration domains with continuity in the normals distribution at their boundaries as shown in Figure (9. e). Hence, the present system is more suitable for reconstruction of complicated surfaces from large clouds suffering from holes and/or mis-registration domains.

5. Conclusions

A hybrid system for surface reconstruction based on the Ball Pivoting Algorithm (BPA) and Radial Basis Functions (RBF) was developed and presented. The present system employs the Ball Pivoting algorithm as a surface reconstruction tool has the capability of processing very large data sets in a little execution time and using small amount of memory. To overcome the weakness of the BPA in the presence of holes, mis-registrations and low sample density, the RBF is employed by the system to uniformly interpolate points representing the missing surface parts. The RBF solution vector is computed once after identifying the holes and/or mis-registration domains from the initial reconstructed surface obtained by the BPA. Then, the process of the interpolation and calling the BPA to triangulate the identified holes and/or mis-registration domains are applied iteratively until reaching perfect surface. The accuracy and robustness were verified in reconstructing surfaces from realistic data with non-uniform sampling and mis-registrations.

References:

[1] Amenta N. and Bern M., Surface reconstruction by Voronoi filtering, *In Proc. of ACM SIGGRAPH '98*, 1998, pp. 514-521.

[2] Amenta N., Choi S., Dey T. and Leekha N., A simple algorithm for homeomorphic surface reconstruction, *In Proc. of 16th ACM Sym. on Comp. Geom.*, 2000, pp. 213-222.

[3] Amenta N., Choi S. and Kolluri R., The power crust, union of balls, and the medial axis transform, *In proc. of the 6th ACM Sym. on solid modeling and applications*, 2001, pp. 249-260.

[4] Boissonat J. and Cazals F., Smooth surface reconstruction via natural neighbor interpolation of distance functions, *In Proc. of the 16th annual sym. on Comp. Geom. 2000*, 2000, pp. 223-232.

[5] Bernardini F., Mittleman J., Rushmeier H., Silva C. and Taubin G., The ball-pivoting algorithm for surface reconstruction, *IEEE Trans. in Vis. and Comp. Graph. 5*, 4, 1999, pp. 349-359.
[6] Carr J., Beatson R., Cherrie J., Mitchell T., Fright W., McCallum B., and T. Evans., Reconstruction and representation of 3D objects with radial basis functions, *In Proc. of ACM SIGGRAPH '2001*, 2001, pp. 67-76.
[7] Carr J., Beatson R., McCallum B., Fright W., McLennan T., and Mitchell T., Smooth surface reconstruction from noisy range data, *In Proc. of Graphite '2003*, 2003, pp.199-126.
[8] Curless B. and Levoy M., A volumetric method for building complex models from range images, *In Proc. of SIGGRAPH '96*, 1996, pp.303-312.
[9] Dey T. and Goswami S., Tight Cocone: A water-tight surface reconstructor, *In Proc. of 8th ACM Sym. on solid modeling and applications*, 2003, pp. 127-134.
[10] Dey T., Giesen J. and Hudson J., Delaunay based shape reconstruction from large data, *In Proc. Of IEEE Sym. in parallel and large data visualization and graphics*, 2001, pp. 18-27.
[11] Edelsbrunner H. and Mücke D., Three-dimensional alpha shapes, *ACM TOG 13*, 1, 1994, pp. 43-72.
[12] Franke R., Scattered data interpolation: tests of some methods, *Mathematics of Computation 38*, 157, 1982, pp.181-200.
[13] Gopi M., Krishnan S. and Silva C., Surface reconstruction based on lower dimensional localized Delaunay triangulation, *In Proc. of Eurographics '2000 19, 3,* 2000.
[14]http://www.graphics.stanford.edu/data/3Dscanrep
[15] Hoppe H., Derose T., Duchamp T., McDonald J. and Stuetzle W., Surface reconstruction from unorganized points, *In Proc. of SIGGRAPH '92*, 1992, pp.71-78.
[16] Kolluri R., Shewchuk J., O'Brien J., Spectral surface reconstruction from noisy point clouds., *In Proc. of Eurographics Geometry Processing Sym.*, SGP04, 2004, pp. 11-21.
[17] Peng L. and Shamsuddin S., 3D object reconstruction and representation using neural networks, *In Proc. of the 2nd international conference on computer graphics and interactive techniques in Australasia and Southe East Asia*, 2004, pp.139-147.
[18] Ohtake Y., Belyaev A., Alexa M., Turk G., and Seidel H., Multi-level partition of unity implicits, *In Proc. of ACM SIGGRAPH '2003*, 2003, pp. 463-470.
[19] Ohtake Y., Belyaev A., and Seidel H., Multi-scale approach to 3D scattered data interpolation with compactly supported basis functions, *In Proc. of Shape Modeling International*, 2003, pp. 153-161
[20] Turk G. and O'brien J., Modelling with implicit surfaces that interpolate, *ACM TOG 21*, 4, 2002, pp. 855-875.
[21] Tobor I., Reuter P., and Schlick C., Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions, *WSCG'2004 12,3*, 2004.

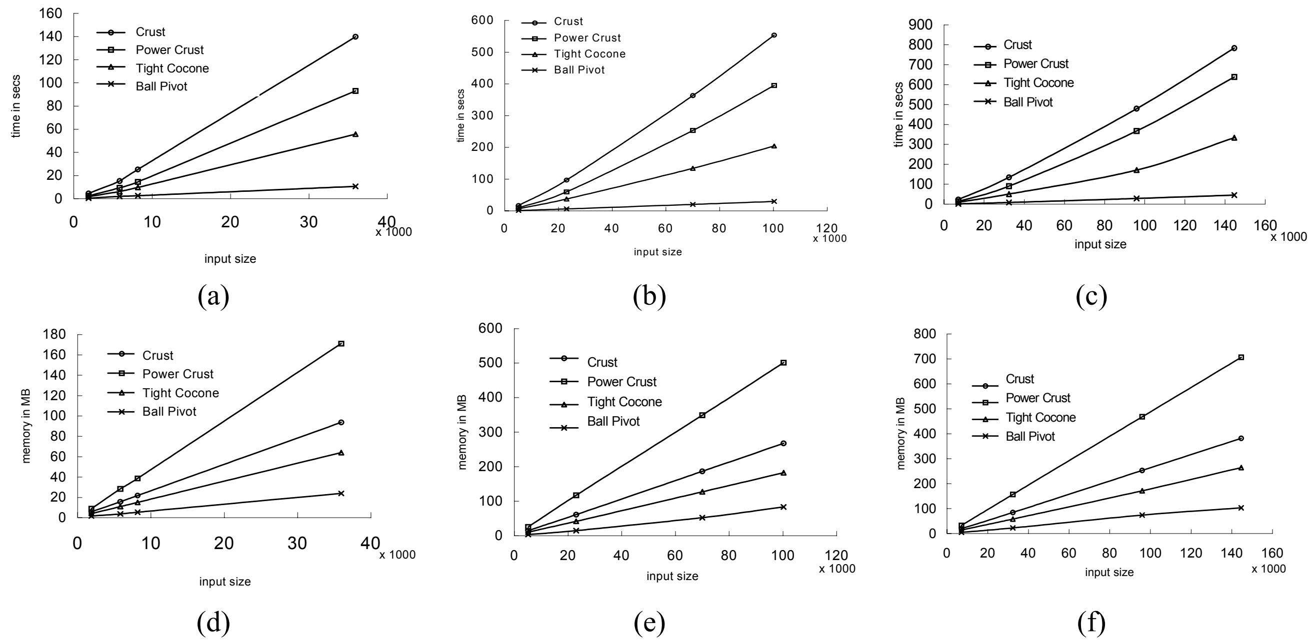
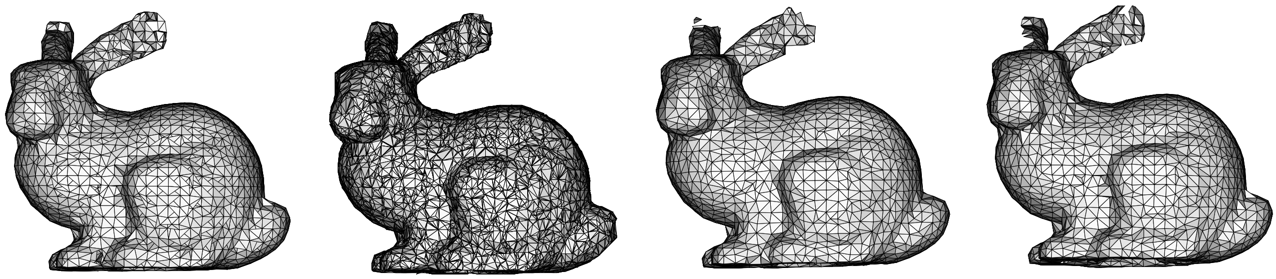
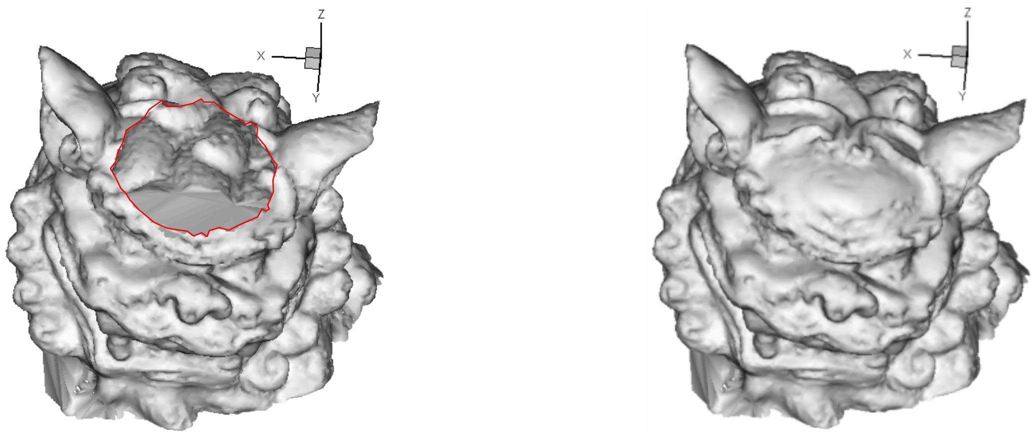


Fig. 6: Comparison between different algorithms from computational time and memory usage points of view. (a, d) Bunny Model. (b, e) Dragon Model. (c, f) Happy Buddha Model.

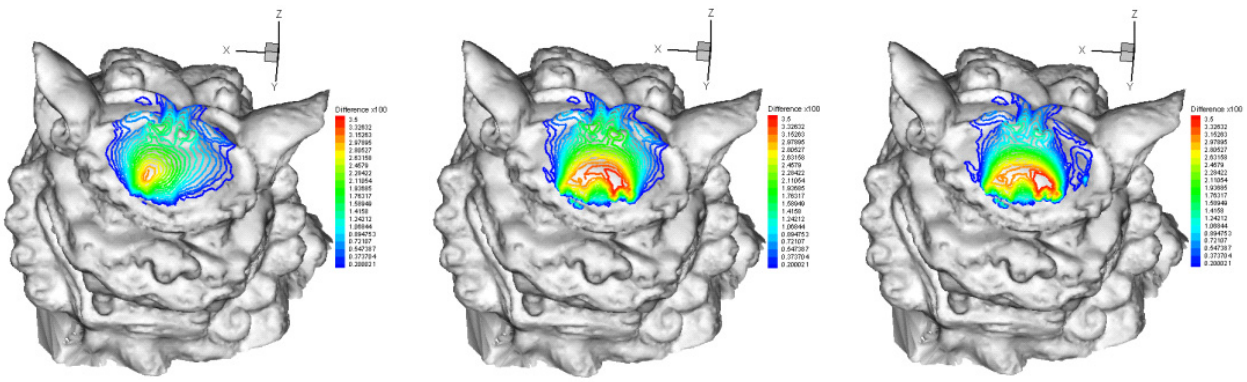


(a) (b) (c) (d)

Fig. 7: Reconstructed Stanford Bunny from 1889 vertices using different algorithms. (a) Crust. (b) Power Crust. (c) Tight Cocone. (d) BPA



(a) Identified hole after BPA reconstruction (b) Final quality reconstruction



(c) 50% HLD (d) 70% HLD (e) 100% HLD

Fig. 8: Accuracy assessment of the present system for three considered values of the hole-neighborhood zone depth. (a) Liondog after BPA reconstruction and hole identification. (b) final quality reconstruction using neighborhood zone depth of 50% HLD (c,d,e) contours of the offset distance between the generated surface and the original surface for the three considered values of the hole-neighborhood zone depth.

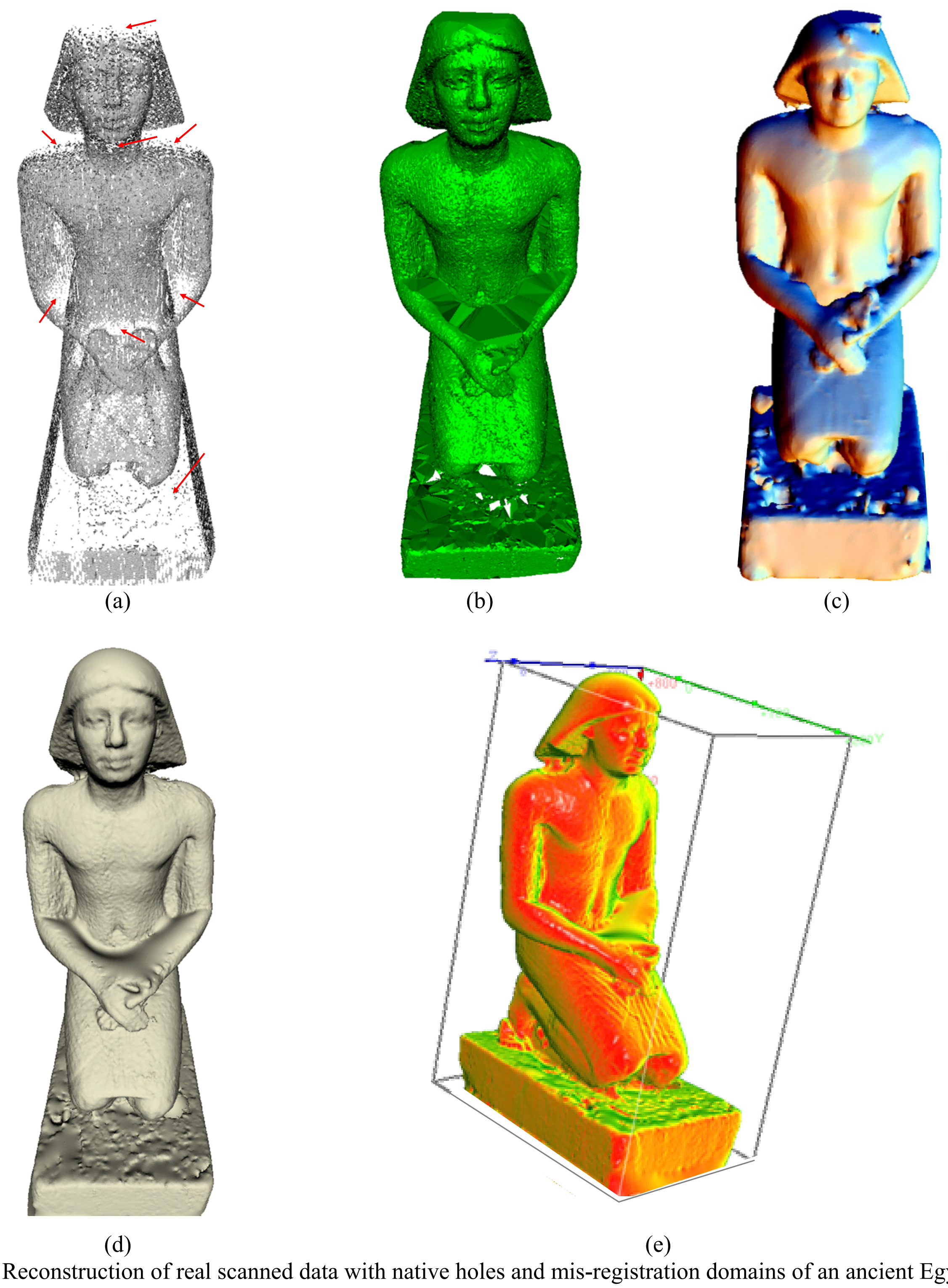


Fig. 9: Reconstruction of real scanned data with native holes and mis-registration domains of an ancient Egyptian man statue. (a) the scanned cloud of 262,461 points with mis-registration domains indicated by the red arrows. (b) reconstructed surface using the Tight Cocone algorithm. (c) reconstructed surface using multi-level partition of unity implicits of [18] (d) reconstructed surface using the present system.(e) surface-normal distribution over the reconstructed object using the present system.