Surface Reconstruction using 3D Morphological Operators for Objects Acquired with a Multi-Kinect System

Eliana Frigerio
DEIB, Politecnico di Milano
V.le Rimembranze di
Lambrate, 14
20134 Milano (Italy)
efrigerio@elet.polimi.it

Marco Marcon
DEIB, Politecnico di Milano
via Ponzio 34/5
20134 Milano (Italy)
marcon@elet.polimi.it

Stefano Tubaro
DEIB, Politecnico di Milano
via Ponzio 34/5
20134 Milano (Italy)
stefano.tubaro@polimi.it

ABSTRACT

This paper describes an innovative extension of morphological operators to the analysis and processing of three dimensional images represented by voxels. In particular, the morphological skeleton well approximates the Medial Axis of a 3D object rendered using a net of Kinect devices: it has proven to be a really effective tool to obtain a compact representation of the object surface that is accurate and very inexpensive in terms of computational time. The reconstructed surface is representable either as a union of balls or as an iso-surface of a 3D function defined as a linear combination of elementary functions with radial support. Furthermore, the representation is easily processable and hierarchically interpretable. The morphological operators neither require any prior information on the reconstructed volume, nor impose any resolution requirement on the object under analysis. The only input for morphological operators is a volume of voxels. Information on the volume nature or resolution requirements are related only to the particular application to which the morphological operators are applied, but do not limit the applicability of any of the proposed morphological operators.

Categories and Subject Descriptors

I.3.5 [**Computing Methodologies**]: Computer Graphics—*Shape modeling*

General Terms

Algorithms, Experimentation, Performance

Keywords

Surface Reconstruction, Morphological Operators, Kinect, Lightwave

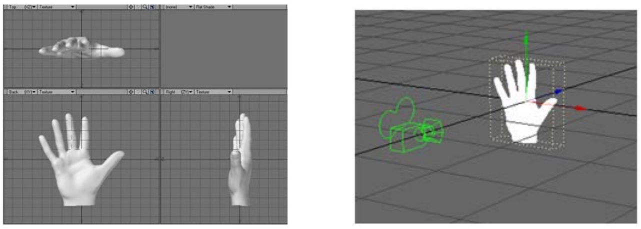
Mirage 2013 Berlin, Germany
Copyright 2013 ACM 978-1-4503-2023-8/13/06 $15.00.

1. INTRODUCTION

Recent improvements in automated shape acquisition have stimulated a profusion of surface reconstruction techniques for computer graphics and reverse engineering applications. A typical surface acquisition application consists of the following steps: (1) data point acquisition: measurement of 3D coordinates of points on the surface producing range images as depth maps; (2) registering and integration: if multiple views are used, a registering process is performed in order to find the transformations that put the data in a common coordinate system, followed by the integration of all the data in a unified representation; (3) surface fitting: construction of a concise and accurate approximation of the physical surface. The goal of a surface reconstruction algorithm is to construct a digital model of a physical surface. The methods compute a surface *S'* that approximates an unknown physical surface S using a sampled set $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ of points on it. Eventually, some additional information about the sampling process or the geometry of the manifold S can be available as well.

Surface reconstruction methods can be roughly classified as *Voronoi-based* and *mesh-free*. Voronoi-based reconstruction algorithms compute the Delaunay triangulation of the sample points, the dual of the Voronoi diagram. A subcomplex, interpolating the sampled surface, is then extracted from the Delaunay triangulation [1], [5]. In the mesh-free approaches, the surface is approximated or interpolated using explicit methods (such as deformable models), parametric methods (such as B-Spline), or implicit methods (such as Radial Basis Functions (RBF), Poisson (PDE) or Moving Least Squares (MLS)) [13], [10], [17]. Among the many techniques developed for surface reconstruction, we can observe that even if the results are very accurate, these methods are time consuming, computationally heavy, and, above all, based on data acquired in a very accurate manner, such as with a laser scanner.

The demand for domestic applications has led to the development of hardware that quickly extracts advanced information of the scene, as Microsoft Xbox Kinect sensor [9]. In particular, in the ISPG laboratory at Politecnico di Milano, an acquisition system made by 3 Kinect devices is developed and it is used to obtain a volumetric representation of an actor in the acquired scene [12]. In this case, however, the surface sampling accuracy is not as high as required by the aforementioned techniques. Basing on all these considerations, we decide to build a system for the reconstruction of the object surface acquired with not very accurate Depth-



(a) (b)

Figure 1: (a) Synthetic Right Hand (20 mm grid) and (b) configuration for the depth map acquisition.

Image-Based Rendering (DIBR) devices.

The proposed solutions for surface reconstruction (in this paper we use surface representation and reconstruction as synonyms) take inspiration from two well known algorithms: the surface representation as union of balls, proposed by Amenta and Kolluri [2], and the surface representation as an iso-surface of a function defined as linear combination of radial basis functions, proposed by Samozino *et al.* [13]. Both methods require high surface sampling density. Therefore, in order to compare the results of the proposed solutions for surface reconstruction with the two techniques taken as inspiration, we develop a simulated environment. This simulation allows to obtain a volumetric reconstruction with a space discretization finer than that achievable with the multi-Kinect acquisition setup developed in the ISPG laboratory.

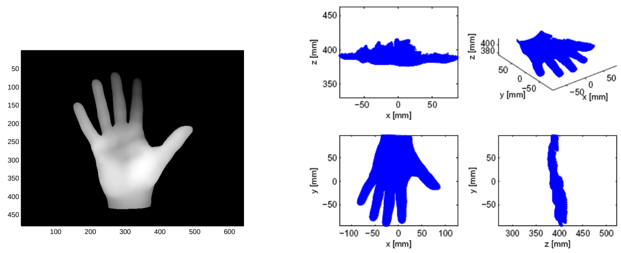
The rest of the paper is organized as follows: in Section 2, the method applied for building the depth maps database used to test the proposed surface reconstruction approaches is described. Section 3 talks about the surface representation as union of Maximal Balls [2] and as an iso-surface [13]. Section 4 revisits the described methods introducing new surface reconstruction methods that use morphological operators. Conclusions end the paper.

2. SIMULATED ACQUISITION SET-UP

Taking into account the high surface sampling density required by the surface algorithms presented in the literature, we cannot use the ISPG database of reconstructed volumes in order to test the implemented methods. We simulate a similar acquisition setup using LightWave 3D in order to achieve a 3D reconstruction of the object in the scene with a sampling density sufficient for applying both the aforementioned techniques for surface reconstruction. In this Section we describe the implemented process for simulating a 3D acquisition set-up similar to the multi-Kinect system, but able to obtain a finer sampling of the 3D space.

2.1 Synthetic Depth Maps using LightWave

LightWave 3D is a high end computer graphics program developed by NewTek [8]. It is a software package used for rendering 3D images, both animated and static. The great potential of LightWave [3] is here exploited to build perspective depth maps, where the gray tone of each pixel represents the distance of the corresponding 3D point to the optical center of the depth-camera taken as reference (the IR-camera of an hypothetic Kinect device). This result is achieved by imposing a specific texture on the illuminated surface of the object. In details, given any synthetic object, as shown in Fig. 1(a), a perspective depth map that reflects the configuration of Fig. 1(b) can be achieved by imposing as texture a gradient that changes the gray tone of the object linearly as a function of the distance from the IR-camera optical center. We use a model of a human hand in order to achieve the sampling accuracy required by the surface reconstruction algorithms implemented. The camera in Fig. 1(b) reflects the position of the Infra-Red (IR) camera of the simulated Kinect device. The gray tone I_{min} ($= 0$, *i.e.*, black) is associated to all the background pixels. They represent points at a distance greater than or equal to d_{max} from the IR-camera (corresponding to 435 mm for our example). The gray tone I_{max} ($= 255$, *i.e.*, white) is associated to all the pixels representing points at a distance less than or equal to d_{min} (corresponding to 375 mm for our case). All the pixels that represent points with distance between d_{min} and d_{max} have a gray tone that is linearly decreasing, while the distance of the corresponding point to the optical center of the device is increasing. Thanks to the linear gradient and imposing no light except the ambient light, we can achieve any depth map simulating a generic viewpoint, as the one shown in Fig. 2(a), which reflects the acquisition conditions of Fig. 1(b). Using LightWave, we are able to simulate perspective depth maps for any acquisition set up and with any resolution needed.



(a) (b)

Figure 2: (a) Depth map achieved using LightWave and (b) 3D surface points reconstruction.

2.2 Canonical Volumetric Reconstruction

Having the depth map and all the IR-camera parameters, we can reconstruct a sampled version of the object surface standing in front of the simulated Kinect device (as shown in Fig. 2(b)). Knowing the depth map and the linear gradient equation, the distance d of any 3D point which corresponds to any pixel in the image can be estimated as:

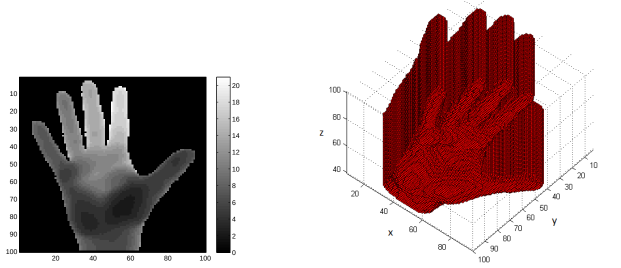
$$d = \frac{d_{min} - d_{max}}{I_{max} - I_{min}}(I - I_{max}) + d_{min}, \tag{1}$$

where I represents the gray tone of the pixel on the depth map.

Assuming that the system is calibrated, both the intrinsic and extrinsic parameters are known. Moreover, knowing the distance of each point from the IR-camera optical center, the 3D coordinates (p_x, p_y, p_z) of the point are easily evaluated:

$$p_z = d\frac{f}{\sqrt{(u-u_0)^2 + (v-v_0)^2 + f^2}}, \tag{2}$$

$$p_x = p_z\frac{u-u_0}{f}, \tag{3}$$



(a) (b)

Figure 3: (a) Elevation depth map and (b) volumetric rendering obtained from the depth map.

$$p_y = p_z \frac{v - v_0}{f}, \tag{4}$$

where (u, v) are the point coordinates on the image plane, (u_0, v_0) are the principal point coordinates, and f is the focal length.

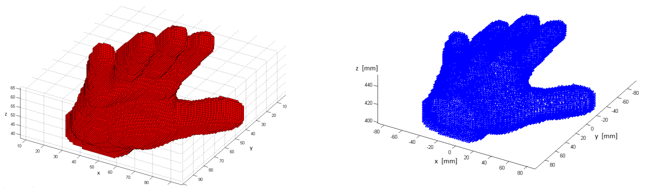
Before doing the 3D reconstruction, the image is processed in order to separate the background from the foreground and to remove any outlier (pixels with gray tone greatly different from those in the neighborhood). Moreover, using LightWave, we obtain perspective depth maps, that are then transformed in elevation depth maps in order to simulate a system that is as close as possible to the real system. In an elevation depth map the gray tone of each pixel represents the distance of the corresponding 3D point to the IR camera image plane, while in a perspective depth map the gray tone of each pixel represents the distance of the corresponding 3D point to the IR camera optical center [14]. Knowing the 3D coordinates of each point on the acquired surface, it is straightforward to compute the gray tone of each pixel in function of the distance from the image plane, as shown in Fig. 3(a). Each 3D estimated surface point is projected on the image plane with a gray tone (d_{el}) equals to:

$$d_{el} = p_z - f. \tag{5}$$

A bicubic interpolation is then applied in order to minimize artifacts due to quantization. Having this new calibrated depth map, we are able to compute the 3D coordinates of each pixel and to project each surface sample till the maximum extension of the considered space (the so called *projection cylinder*), as shown in Fig. 3(b). The volumetric rendering is achieved working with voxels (in this case using boxes of 2 mm side).

2.3 Volumetric Integration

We implement a multi-view acquisition setup to obtain a complete description of the 3D object. Therefore, it is necessary to merge the 3D information coming from different depth maps taken from different viewpoints. We first reconstruct the scene seen by each device and then merge the volumetric rendering achieved from any view. This volumetric reconstruction is performed for all the available views by assuming a regular grid for each of them (canonical space for each view). Once the voxel space centered on each device is obtained, applying the roto-translation matrix that brings each view to a predefined reference view, it is possible to intersect the different projection cylinders applying a logical *AND* between voxels. The result is a unique voxel set that gradually incorporates information coming from all the devices. In Table 1 the depth maps achieved with three other configurations, in which the devices are turned by $H = +45°$, $H = -45°$ and $H = 180°$ respectively (using the LightWave definition for angles), the volumetric rendering obtained with each of them, and the voxel set obtained by increasing the number of views are shown. The resulting voxel set compactly represents all the available information about the object surface and its volume occupancy.



(a) (b)

Figure 4: (a) Voxel surface and (b) sampled surface of the volumetric rendering obtained from the depth map.

A morphological algorithm to extract the voxel surface starting from the volumetric occupancy is proposed. There are different possible methods to extract the object surface having its volume voxel occupancy. For this application, we find that the best method is to operate a dilation ($\oplus E$) of the object B by the structuring element E and to define the surface β (approximation of the topological boundary of B) as the difference between the result and the original object:

$$\beta(B) = (B \oplus E) - B. \tag{6}$$

The structuring element E is a $3 \times 3 \times 3$ block of voxels equal to 1 (full voxels). The centers of the surface voxels are the available sampled set X for the surface representation methods described in this paper. In Fig. 4 the surface voxels (a) and their centers (b) are represented. The sampled set X is, in the proposed example, composed by the 56168 centers of the voxels obtained with the surface extraction algorithm. The achieved sampling accuracy is sufficient for the considered surface reconstruction algorithms.

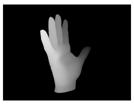
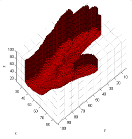
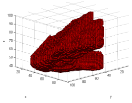
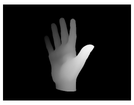
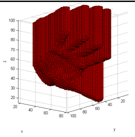
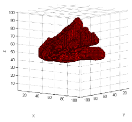
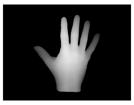
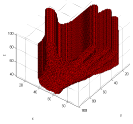
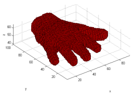
3. MEDIAL AXIS APPROXIMATION BY A SUBSET OF VORONOI VERTICES

In this Section we describe the idea behind two well known algorithms for surface representation [2] [13]. Starting from a dense point cloud of sample points, presumed to be on or close to the object boundary, first the Voronoi diagram of the scattered data points is computed and then the *medial axis* is approximated by a subset of Voronoi vertices, called *poles*. The medial axis (MA) and the medial axis transform (MAT), introduced by Blum [4], are fundamental concepts for surface reconstruction and they deserve to be explained in detail.

3.1 Medial Axis and Medial Axis Transform

In 2D, the Medial Axis (MA) of a shape is a set of curves defined as the locus of points that have at least two closest points on the boundary of the shape [11]. In the 3D case, the corresponding object is also called *medial surface* because, in addition to curves, it can also contain surface patches. Considering a 3D object $B \subset R^3$ and its boundary $\beta(B)$, the

Table 1: Volumetric Reconstruction using LightWave. Depth maps obtained with LightWave by rotating the acquisition system of H degrees. Volumetric reconstruction obtained from a single depth map. Volumetric reconstruction obtained downstream the merging.

Angle (H)	Depth Map	Projection Cylinder	Volumetric Reconstruction
$45°$			
$-45°$			
$180°$			

medial surface of B (in this paper we use medial axis and medial surface as synonyms because we consider only 3D objects) is defined as the locus of centers of all the maximal balls of B which touch β in, at least, two points. A ball of radius r centered at $\mathbf{x} \in B$ is defined as:

$$Sr(\mathbf{x}) = \{\mathbf{y} \in R^3, d(\mathbf{x}, \mathbf{y}) \leq r\}, \tag{7}$$

where $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between two points $\mathbf{x}$ and $\mathbf{y}$ in R^3. A ball $Sr(\mathbf{x}) \subset B$ is maximal if it is not completely included in any other ball included in B [11]. The *medial axis transform* MAT is defined as the function that assigns to each medial axis point the radius value of its Maximal Ball. It is shown that the object B could be extracted from MAT(B) as the union of all the Maximal Balls balls [2]. The boundary of B can be calculated as the boundary of the union of all the Maximal Balls (inverse MAT). MA has another great advantage: an object and its inner MA have the same topological structure and can continuously deform to each other. Exact computation of both the MA and the MAT is extremely difficult and slow in a three-dimensional space [1], [6].

3.2 Surface Representation as Union of Balls

The internal edges and faces of the Voronoi diagram of the scattered data is an approximation of the medial surface of the shape [2]. Each Voronoi vertex is the center of a circum-sphere of the Delaunay triangulation and the connection of all the centers of the circum-spheres produces the Voronoi diagram. Considering all the Voronoi vertexes produces many balls with centers close to the surface, corresponding to flat "sliver" tetrahedra in the Delaunay triangulation of X [2]. These balls may be (and almost always are) present even when the data are completely noise-free, at any finite sampling density. Their centers must be removed with a heuristic clean-up step. Different algorithms based on this approach use different methods for selecting the desired subset of the Voronoi diagram. In particular, after the computation of the Voronoi diagram of the scattered data points, Amenta *et al.* retrieve the poles [1]. The poles of a sampled surface point are the farthest vertex of its Voronoi cell in the interior of β and the farthest vertex of its Voronoi cell on the exterior of the boundary β. Connecting these interior poles P_I forms a good approximation to the MA and the union of the inner polar balls forms a good approximation of the object surface. Given a "good sample" from a smooth surface, they demonstrate that the output is guaranteed to be topologically correct and convergent to the original surface as the sampling density increases. The required sampling density varies locally, rigorously capturing the intuitive notion that featureless area can be reconstructed from fewer samples.

We implement the Amenta *et al.* algorithm [2] on the synthetic object shown in Fig. 1(a) and the result is shown in Fig. 5(a). Starting from the sampled 56168 points on the surface, we compute the Voronoi diagram and we select 8338 poles (Fig. 5(b)).

3.3 Surface Representation as Implicit Surface

Generally, implicit reconstruction methods attempt to find a smooth function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ is close to the zero set $Z(f)$. Among the mesh-free approaches, Samozino *et al.* algorithm [13] grabs our attention because it approximates the surface as a zero level set of a function defined as a linear combination of compactly supported radial basis functions (RBF), each one associated with a polar ball computed using Amenta *et al.* approach [2].

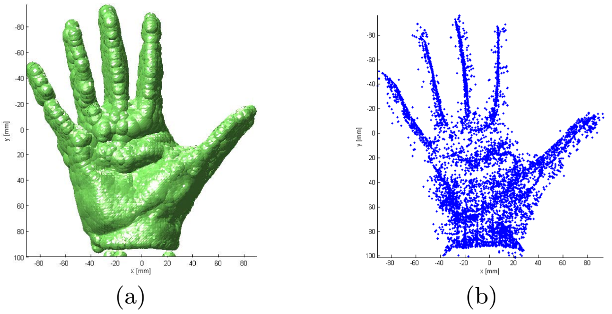
The approximation problem is formulated as follows. Given $\{\mathbf{p}_i\}_{i=1,...,N}$ a set of N points (called *constraints*) and N scalar numbers $F = \{f_i\}_{i=1,...,N}$, find a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ satisfying the approximation condition:

$$f^* = argmin_f E(f), \tag{8}$$

where E is the least square error:

$$E(f) = \sum_{i=1}^{N} (f_i - f(\mathbf{p}_i))^2. \tag{9}$$

In the RBF approach, the function f is defined from a class of basis functions $\Phi_j : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$, as a linear com-



(a) (b)

Figure 5: (a) Surface representation as union of balls. (b) Selected poles.

bination:

$$f(\mathbf{x}) = \sum_{j=1}^{m} \alpha_j \Phi_j(\mathbf{x}, \mathbf{c}_j), \tag{10}$$

where $\{\mathbf{c}_j\}_{j=1,\dots,m}$ denotes a set of m center points and $\{\alpha_j\}_{j=1,\dots,m}$ denotes a set of unknown weights to be solved for.

The reconstruction problem boils down to determine the vector $\alpha = [\alpha_1, \dots, \alpha_m]^T$ by computing the best least square approximation of the function f with respect to the constraints $\{\mathbf{p}_i\}$ (Eq. (9)). The minimization of E (Eq. (9)) implies solving the linear system of equations [13]:

$$G^T G\alpha = G^T \mathbf{f}, \tag{11}$$

where G is a $N \times m$ matrix defined as:

$$G = [\Phi(\mathbf{p}_i, \mathbf{c}_j)]_{i=1,\dots,N, j=1,\dots,m} \tag{12}$$

and $\mathbf{f}$ is a N-elements vector defined as $\mathbf{f} = [f_1, \dots, f_N]^T$. The size of the matrix to be inverted is $m \times m$, where m is the number of centers.

The set of points P, where the function value is specified a priori (called *constraints*), includes the surface points where the function f should vanish by definition, *i.e.*, where all the f_i should be zero. To avoid the trivial solution $\alpha = \mathbf{0}$, during the minimization of E in Eq. (9), several interior constraints are added, where the function f does not vanish. For each additional constraint point $\mathbf{p}_k$, we assign to f a signed value f_k. This value is the approximated signed distance between $\mathbf{p}_k$ and the sampled surface. The N constraints $\{\mathbf{p}_i\}$, with $\{i = 1, \dots, N\}$ are, therefore, composed by the n data points (surface points) and by the m additional off-surface constraints, where the function f is specified.

In particular, in the Samozino *et al.* algorithm [13], the additional m constraints are called *centers* and they are a subset of the poles computed using the Amenta and Kolluri approach [2]. Since each center $\mathbf{c}_j$ is an inside pole, it has as corresponding scalar value (f_j) the radius of its corresponding circum-sphere of the Delaunay triangulation (r_i). The algorithm of Samozino *et al.* [13] proceeds as follows: given a 3D point set scattered on a surface, first its Delaunay triangulation and the dual Voronoi diagram are computed. Then the authors repeatedly refine a subset of the Voronoi vertices. In the first stage, poles are extracted from the Voronoi vertices and they are classified as inside or outside (following the Amenta *et al.* approach [2]). In the second stage, a user-defined number of m centers is selected, among the set of poles. The selection is performed by filtering and then clustering the set of poles. Poles are filtered in order to adjust the level of detail to the budget of centers and they are clustered in order to achieve a center distribution nicely spread on the medial axis.

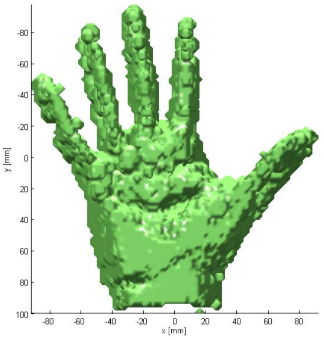


Figure 6: Surface representation as iso-surface of a function defined as a linear combination of RBFs.

The reconstructed surface is required to be independent from Euclidean transformations. The function Φ is thus restricted to the set of radial functions:

$$\Phi(\mathbf{x}, \mathbf{c}_i) = \phi(\|\mathbf{x} - \mathbf{c}_i\|), \tag{13}$$

where $\|.\|$ denotes the Euclidean distance and $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$. In order to obtain a sparse matrix G, making the computational load tractable also for a great amount of sampled surface points, compactly supported RBF are introduced by Wendland [16]. Other compactly supported RBFs (CSRBF) can be used for reconstruction [15]. The basis function, centered on $\mathbf{c}_i$, chosen in our implementation (defined in Eq. (13)) is:

$$\phi(r_i) = (1 - r_i)_+^4 (1 + 4r_i), \tag{14}$$

where the symbol + means:

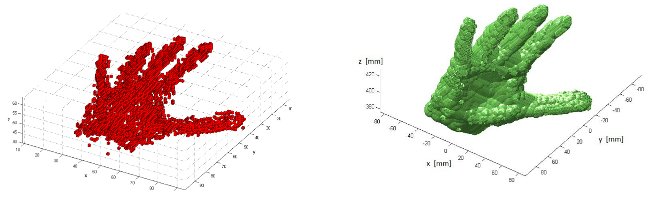
$$\begin{cases} (w)_+ = w & \text{if} \quad w > 0 \\ (w)_+ = 0 & \text{otherwise} \end{cases}, \tag{15}$$

and $r_i = \|\mathbf{x} - \mathbf{c}_i\|$. The use of compactly supported functions f_i leads to a sparse matrix $G^T G$ with about 90% zero elements. Each term $e_{i,j}$ of the matrix $G^T G$ is computed as a sum:

$$e_{i,j} = \sum_{k=1}^{N} \phi_i(\|\mathbf{p}_k - \mathbf{c}_i\|)\phi_j(\|\mathbf{p}_k - \mathbf{c}_j\|). \tag{16}$$

Therefore, for each constraint $\mathbf{p}$, we need to find the list l_p of centers which contains $\mathbf{p}$ in their support. For each pair of centers $(\mathbf{c}_i, \mathbf{c}_j)$ contained in the list l_p, we add a term for $\mathbf{p}$ in the computation of $e_{i,j}$.

We implement the algorithm in Matlab and, for the purposed example (n=56168 surface points and m=8338 centers), the computation of $G^T G$ takes about 3 minutes, avoiding any efficient research but making it exhaustively. The obtained matrix is sparse: only 6% of its entries are different from zero. The zero level of the obtained function $f(\mathbf{x})$, computed from Eq. (8), is presented in Fig. 6 and the computation time needed for $100 \times 100 \times 100$ points is 1 minutes, greatly less than the time needed to represent the balls in Fig. 5 (several minutes) because only the iso-surface $f(\mathbf{x}) = 0$ is represented and not all the balls.



(a) (b)

Figure 7: (a) Medial surface approximation with the morphological skeleton. (b) Surface approximation as union of balls associated to the morphological skeleton.

4. IMPLEMENTED ALGORITHMS FOR SURFACE REPRESENTATION

Thinking on applications that do not require high level of accuracy or details in the reconstructed surface (thinking on an avatar representation and not on a reconstruction of a digital model for a mechanical artifact for example), we revisit the two described algorithms to approximate the object surface.

The Voronoi-based methods do not work well if the surface resolution is low, as with the 3D reconstructions using a multi-Kinect system [12]. If the surface sampling is too sparse, the Voronoi cells are neither narrow, nor orthogonally oriented respect to the surface [1]. Therefore, the poles selection is useless and the reconstruction fails. Considering this limitation, we still focus on the medial surface concept, but we look for a morphological approximation of it (instead of that based on the Voronoi diagram).

4.1 The Morphological Skeleton

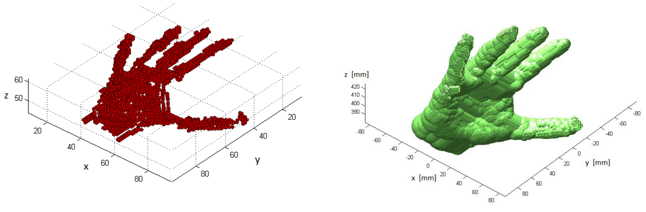
We approximate successfully the medial surface implementing a 3D extension of the 2D morphological skeletonization algorithm [7]. The morphological skeleton operator can be expressed in terms of sequence of morphological operations of erosion and opening. Given the structuring element E and indicating with $\ominus k$ a sequence of k morphological erosions and with $\circ$ a morphological opening, the morphological skeleton of B ($S(B)$) can be expressed as:

$$S(B) = \cup_{k=1}^{K} S_k(B), \tag{17}$$

where:

$$S_k(B) = (B \ominus kE) - ((B \ominus kE) \circ E). \tag{18}$$

This procedure is iterated until $B \ominus KE$ is an empty set. This algorithm, as well as its version in 2D, does not produce a connected skeleton even if the original object is connected. The obtained morphological skeleton (shown in Fig. 7(a) for the reconstructed object at the right-bottom of Table 1) does not preserve the topology of the original object, as well as it is not maintained approximating the MA as a set of poles of the Voronoi diagram. The main important property is that the reconstructed surface maintains the original object connectivity and this is true both for the proposed and the Voronoi-based methods. Moreover the proposed MA approximation is applicable independently from the sampling accuracy of the available volumetric data, while Voronoi based MAT approximation methods work well only for a required sampling density [1]. Another great advantage of our method is its low computational load that makes it greatly faster than any Voronoi based method. The computation of the 4043 morphological skeleton voxels extracted with the proposed algorithm takes less than a second, greatly less than the hour taken for extracting the poles from the computed Voronoi diagram implementing the Amenta *et al.* approach [2]. Both algorithms run on a PC under Windows 7 (64 bit), using a 1.60 + 1.60 GHz Intel Dual Core CPU.



(a) (b)

Figure 8: (a) Morphological skeleton filtered imposing that each block has at least 20 element 26-connected and (b) the 3074 (76% of the original) corresponding balls.

4.2 Union of Balls

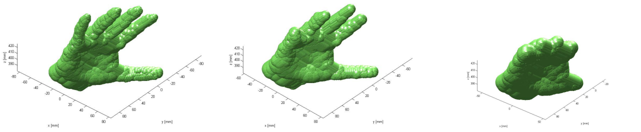
Starting from our very fast MA approximation as the morphological skeleton, it is possible to represent the object surface as union of balls. Each of the $S_k(B)$ voxels can be considered as the center of a ball, with a radius that depends on the step at which it is generated. Since we use as structuring element a $3 \times 3 \times 3$ block of voxels with origin at its center, to each skeleton-voxel corresponds a sphere with center located in the center of the voxel itself and with radius r equal to:

$$r = (k - \frac{1}{2})l, \tag{19}$$

where k indicates the step of the iterative algorithm in which the skeleton voxel is generated, and l is the 3D space sampling interval size expressed in millimeters. Fig. 7 (b) shows the balls associated to the skeleton of Fig. 7(a) and it can be observed that the obtained surface is homotopic with the original object. In particular, we avoid the $S_0(B)$ morphological skeleton voxels because they are voxels belonging to the surface of B and do not produce any sphere (as can be noted from the subscript in the summation in Eq. (17)).

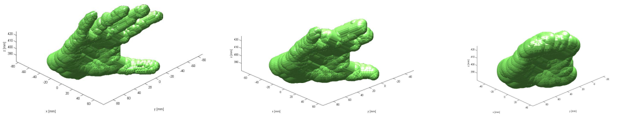
Each of these balls can be also easily interpreted in statistical terms as a Gaussian probability distribution, with mean (μ) equal to the center of the sphere and standard deviation (σ) for each direction equal to one third of the ball radius. As for a Gaussian distribution, the 99.7% of possible values are inside the interval $\mu \pm 3\sigma$, the union of the spheres can be seen as the union of the iso-surfaces at 0.997 probability of each Gaussian distribution.

Comparing the results obtained with the Amenta *et al.* [2] and the proposed representation as union of balls (Fig. 5(a) and Fig. 7(b)) we can observe that both representations are quite noisy: many balls with small radius and near to the surface are present, even if we work with a noise free object in this case. More refinements in the centers choice must be done in order to obtain a more smooth and appreciable result, incrementing the computational time and the number of machine operations. A simple method to reduce the number of balls representing the object surface is to perform a k-means clustering over the set of all the possible centers.



(Radius $\geq$ 5 mm) (Radius $\geq$ 7 mm) (Radius $\geq$ 9 mm)

Figure 9: Balls filtered according to the minimum allowable radius.



(Nsec $\geq$ 50) (Nsec $\geq$ 100) (Nsec $\geq$ 250)

Figure 10: Balls filtered according to the minimum allowable number of intersecting balls.

After the convergence of the clustering procedure, the centroid of each cluster is replaced by the closest voxel within its cluster. It guarantees that the final centers are located on the original morphological skeleton (or Voronoi diagram).

Other approaches seem more interesting in our case. In order to filter the resulting balls, we can compute the number of 26-connected voxels that form a morphological skeleton block and cut the blocks with less elements than a threshold. We implement a morphological algorithm for counting how many 26-connected blocks compose a 3D image and also for counting by how many voxels each block is made. Starting from a foreground voxel belonging to B ($\mathbf{p}_0$), the goal is to find all the components connected to that voxel by following the iterative procedure:

$$\mathbf{p}_k = (\mathbf{p}_{k-1} \oplus E) \cap B \qquad k = 1, 2, 3, ... \qquad (20)$$

The procedure terminates when $\mathbf{p}_k = \mathbf{p}_{k-1}$ and produces the connected component B_i. Then the same iterative procedure starts again on another foreground voxel of $B - \{\cup_{j=1}^{i} B_i\}$ till no more foreground voxels remain. Fig. 8 shows the result imposing that each block must contain at least 20 voxels that are 26-connected.

Moreover, this representation allows a hierarchical interpretation: setting the minimum acceptable radius, we can range from the most massive elements of the object representation till a more detailed description, as summarized in Fig. 9. In this case we have 4043 spheres with minimum radius equal to 3 mm and maximum radius equal to 17 mm, that is likely equal to the thickness of the hand palm. Fig. 9 shows that by increasing the minimum acceptable radius, first the fingertips are missed and then only the palm remains. Considering all the spheres with radius greater than 5 mm, 3633 spheres remain, with radius greater than 7 mm, 2301 spheres remain, and, with radius greater than 9 mm, only 1673 spheres are left.

Another interesting way to study the hierarchy of the object parts is to consider the number of balls that intersect each skeleton ball: Fig. 10 represents the results setting the minimum number of intersecting spheres allowed. The spheres with more than 50 intersecting spheres are 2825, with more than 100 intersecting spheres are 2208 and with more than 250 spheres are 1455.

In order to obtain a MA that maintains the topology of the original object, a postprocessing procedure can be applied: first the number of 26-connected elements of the skeleton can be estimated and, eventually, filtered; then they can be linked to each other following the path indicated by the linked spheres.

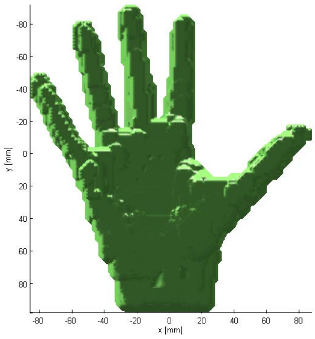


Figure 11: Surface representation as zero level of a function defined as linear combination of RBFs associated to the morphological skeleton.

4.3 Implicit Surface

Starting from the proposed approximation of the medial surface using the morphological skeleton, also the Samozino *et al.* algorithm [13] can be reviewed and it is possible to represent the object surface as the zero level of a function defined as linear combination of radial basis functions.

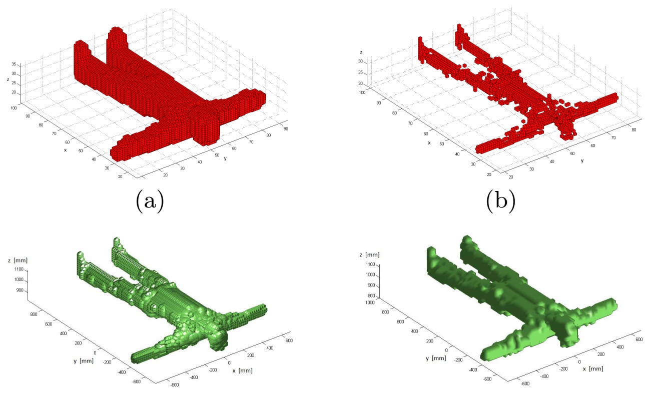
Data points are the centers of surface voxels and the m centers are the centers of the morphological skeleton voxels (neglecting the voxels resulting from the first iteration). Therefore the N constraints are the $n = 56168$ surface points plus the $m = 4043$ voxels on the approximated medial surface. We impose the function f to be zero at the surface points and to be equal to the radius of the corresponding maximal ball (r_i) at the morphological skeleton voxels.

For the proposed example, the computation of $G^T G$ takes 31 s and only 1% of its entries are different from zero. The computation time needed for $100 \times 100 \times 100$ points of the zero level of the function $f(\mathbf{x})$, computed from Eq. (8) and represented in Fig. 11, is less than 1 minute, greatly less than the time needed to represent all the balls as in Fig. 7(b) (more than 5 minutes). In comparison to the algorithm proposed by Samozino *et al.* [13], fewer centers are used. This leads to a faster computation of the weights and a faster evaluation of the functions. The effects of some noisy balls can be seen on the tip of the thumb, for example. These balls can be removed by applying one of the previously proposed filtering methods.

The main difference with the Samozino *et al.* method is that they compute the Delaunay triangulation and the dual Voronoi diagram of the given 3D surface points. Then they repeatedly refine a subset of the Voronoi vertices. In our algorithm we apply the 3D morphological skeleton extraction algorithm and, eventually, we filter the extracted voxels.

5. CONCLUSIONS

In this paper we propose a technique, based on morphological operators, to approximate the Medial Axis Transform in terms of surface representations as the union of balls (Fig. 12(c)). This method can be applied independently from the sampling accuracy of the available volumetric data, while



(a) (b)

(c) (d)

Figure 12: Surface representation as (c) union of balls and as (d) zero level of a function defined as linear combination of RBFs. Balls or RBFs are associated to the (b) morphological skeleton (m= 1835 voxels). (a) Surface (n=9822 voxels). The 3D object is taken from [12].

Voronoi based MAT approximation methods work well only for a required sampling density [1]. This important feature suggests its use with a 3D reconstruction obtained with not very accurate DIBR devices (as the multi-Kinect system implemented in the ISPG laboratory [12]). We obtain a really fast and reliable method to represent human body surface for each frame. Another great advantage of our method is its low computational load that makes it greatly faster than any Voronoi based method. Moreover, by applying an easy post-processing (filtering considering the number of 26-connected elements for each block, or the radius, or the number of intersecting spheres, or a combination of the previous), we are able to reduce significatively the number of balls used to represent the object and to extract its most important components (in term of the metric used). This approach allows us also to obtain a surface representation as an iso-surface of a three-dimensional function (Fig. 12(d)). The matrix G^TG, for the example proposed in Fig. 12 (taken from [12]), has only 1.49% of non null elements.

We believe that this light computational and fast algorithm is attractive for the development of domestic and user friendly applications.

6. REFERENCES

[1] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266, 2001.

[2] N. Amenta and R. Kolluri. Accurate and efficient unions of balls. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 119–128, 2000.

[3] G. Avgerakis. *Digital animation bible*, volume 1. McGraw-Hill/TAB Electronics, 2003.

[4] H. Blum et al. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form*, 19(5):362–380, 1967.

[5] F. Cazals and J. Giesen. Delaunay triangulation based surface reconstruction. *Effective Computational Geometry for Curves and Surfaces*, pages 231–276, 2006.

[6] T. Dey and W. Zhao. Approximate medial axis as a voronoi subcomplex. *Computer-Aided Design*, 36(2):195–202, 2004.

[7] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002.

[8] J. Hebert. Lightwave 3d user guide. *NewTek*, 2011.

[9] I. iSuppli. The teardown: The kinect for xbox 360. *IET Engineering Technology*, 6(3):94 –95, Apr. 2011.

[10] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[11] T. Kong and A. Rosenfeld. Digital topology: introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393, 1989.

[12] M. Marcon, A. Sarti, and S. Tubaro. http://www-dsp.elet.polimi.it/ispg/index.php/description.html, 2010.

[13] M. Samozino, M. Alexa, P. Alliez, and M. Yvinec. Reconstruction with voronoi centered radial basis functions. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 51–60, 2006.

[14] A. Sarti and S. Tubaro. Image-based surface modeling: a multi-resolution approach. *Signal processing*, 82(9):1215–1232, 2002.

[15] R. Schaback. Creating surfaces from scattered data using radial basis functions. *Mathematical methods for curves and surfaces*, pages 477–496, 1995.

[16] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, 4(1):389–396, 1995.

[17] H. Wendland. *Scattered data approximation*, volume 17. Cambridge University Press, 2005.