Efficient surface reconstruction method for distributed CAD

Sergei Azernikov, Anath Fischer
Laboratory for Computer Graphics and CAD, Faculty of Mechanical Engineering, Technion, Haifa 32000, Israel
Accepted 12 September 2003

Abstract

This paper describes a new fast Reverse Engineering (RE) method for creating a 3D computerized model from an unorganized cloud of points. The proposed method is derived directly from the problems and difficulties currently associated with remote design over the Internet, such as accuracy, transmission time and representation at different levels of abstraction. With the proposed method, 3D models suitable for distributed design systems can be reconstructed in real time. The mesh reconstruction approach is based on aggregating very large scale 3D scanned data into a Hierarchical Space Decomposition Model (HSDM), realized by the Octree data structure. Then, a Connectivity Graph (CG) is extracted and filled with facets. The HSDM can represent both the boundary surface and the interior volume of an object. Based on the proposed volumetric model, the surface reconstruction process becomes robust and stable with respect to sampling noise. Moreover, the data received from different surface/volume sampling devices can be handled naturally. The hierarchical structure of the proposed volumetric model enables data reduction, while preserving significant geometrical features and object topology. As a result, reconstruction and transmission over the network are efficient. Furthermore, the hierarchical representation provides a capability for extracting models at desired levels of detail, thus enabling designers to collaborate at any product development stage: draft or detailed design.

© 2003 Elsevier Ltd. All rights reserved.

Keywords: Surface reconstruction; Reverse engineering; Octree; Multiresolution; Distributed CAD

1. Introduction

1.1. A volumetric multiresolution model for distributed design

One of the main challenges in design for manufacturing today is to reduce the cycle of time to market for designing and prototyping new products. This goal can be achieved by integrating technologies such as reverse engineering and by increasing interactive collaboration between designers or between remote design and manufacturing centers, utilizing web media in a distributed environment. This distributed design structure seems to be a promising candidate for significantly improving the process of product design. In this paper, a method is proposed that reconstructs a volumetric multiresolution model suitable for distributed design over the web. The proposed reconstruction method is derived directly from problems and difficulties currently associated with remote design over the Internet such as accuracy, time and representation at different levels of abstraction. Thus, the reconstruction method must satisfy the following requirements:

- The method should be robust and handle a variety of sampling technologies (i.e. 3D scanners, 3D cameras, CT and MRI).
- The method should be fast so designers can create computerized models from initial products in real time.
- The resulting model should have an inherent structure such that it can be transmitted over the web at different levels of detail and abstraction, so designers can collaborate on any stage of the design: draft or detailed design.
- The reconstructed model should be transmitted as a surface for modeling and rendering applications so as to provide the basis for a mechanical analysis.

Using the proposed reconstruction method, a multi-resolution model is created that can be used in a closed and complete cycle based on remote design, geometry submission and mechanical analysis. The improvements expected as a result of using this model should make...
a significant contribution in the context of interactive real-time remote design over the web.

1.2. Problem statement

This work addresses the problem of reconstructing a computerized model from a sampled physical object. The paper focuses on the problem of surface reconstruction, as follows

Problem 1.1. Given a cloud of points $\mathcal{P}$ densely sampled from an unknown object surface $\mathcal{O}$, reconstruct a mesh $\mathcal{M}$ that approximates $\mathcal{O}$, while preserving its topology and features.

The sampling process usually involves noise, so the sampled points are an approximation under some tolerance. The mesh $\mathcal{M}$ is not constrained to interpolate the sampled points. Instead, an approximation is desired, while preserving topology and critical features such as sharp edges and corners that are common in mechanical parts. The properties required from the reconstructed mesh $\mathcal{M}$ are as follows:

- It should have the same topology as the object.
- It must be a geometrical approximation of the object.
- It has to preserve critical features such as sharp edges and corners.
- It has to be represented as a mesh in a standard format suitable for CAD/CAM applications, such as Stereo Lithography manufacturing or Finite Element simulation.

2. Overview

During the last decade, researchers from the fields of image processing, computational geometry and computer graphics have proposed different approaches for the surface reconstruction problem, utilizing methods common in those fields. Following is an overview of these approaches with respect to RE.

In the image processing field, the surface reconstruction problem can be basically considered a special case of the segmentation problem in $\mathbb{R}^2$. In the last two decades, the focus has been on developing fast and robust algorithms for 2D segmentation. These algorithms are based primarily on the active contours or snakes approach [16]. The idea behind this approach is as follows: starting from an initial bounding contour, change its position according to a predefined energy functional. That is, an elastic boundary is deformed under the exterior forces, elastic forces and constraints inherent in the input data. The same approach can be applied to segmentation of 3D images (i.e. MRI). Starting from an initial bounding surface, deform it towards the boundary of the desired object. Zhao et al. [23] utilized level sets for a deformable surface representation. Doi et al. [10] used the 3D Active Grid model for volumetric medical data. This approach extends the idea of the deformable surface to a non-rigid volume.

The main drawback of these image-processing algorithms is the relatively high number of user-specified parameters. In order for the active contour to be attracted to the object’s boundary, the forces applied on the contour must be tuned. This tuning depends on the curvature of the object’s boundary, the sampling noise and other factors. Another difficulty with these algorithms is their computational complexity. Their iterative nature and heavy dependence on the initial condition make them unusable in real-time automatic systems.

Computational geometry studies have focused on piece-wise-linear interpolation of unorganized points. These algorithms start from the convex hull of the points set and then subtract the redundant entities in order to accommodate the space portion surrounded by the points set. Boissonat [5] applied the Delaunay triangulation to the points’ convex hull. Then, redundant tetrahedra were eliminated, preserving the manifoldness of the exterior surface. To define the order of the tetrahedra to be removed, some geometrical measures were predefined. Amenta and Bern [1] proposed a method that eliminates triangles from the Delaunay triangulation using Voronoi filtering. Later, Amenta et al. [2] proposed an approach based on medial axes transform. Recently, Dey et al. [9] proposed another Delaunay-based approach focused on reconstruction from large scale data.

Edelsbrunner and Mücke [11] introduced the alpha shape structure. Alpha shape associated with a set of points is a subcomplex of the Delaunay triangulation, which consists of simplices whose circumsphere of radius is at most $\alpha$. Bernardini et al. [3] improved the alpha shape structure as follows: (1) automatic selection of an optimal $\alpha$ value, (2) sharp edge detection and (3) piecewise algebraic implicit surface fitting. Alpha shapes have been proven to be a precise and elegant representation of reconstructed smooth surfaces. However, if the sampling process is noisy or the original surface is rough, the alpha shape of the points set will not be a 2-manifold. It might suffer from residuals, depending on the $\alpha$ value. Mencel and Müller [19] proposed a graph-based surface reconstruction method. This method is based on constructing a graph that approximates a wire frame of an unknown surface. This wire frame is later filled with triangles by applying several restricting rules. Our approach utilizes a similar graph-based approach in Section 3.2.1.

As was mentioned above, computational geometry surface reconstruction approaches interpolate the data points. These approaches provide high performance for noiseless data. However, if the noise is significant, approximating the surface is preferable to interpolating it through the data points.

In the fields of computer graphics and computer aided geometric design, researchers have focused on the visual
triangles and bad approximation of sharp features. Curless and Marching Cubes algorithm suffer from nearly singular with arbitrary topology. However, meshes produced by the method worked effectively on real test cases and objects extracted using the Marching Cubes algorithm [18]. This low curvature Then, normal directions are propagated through smooth, every sub-set of sampled points by

scanner

tection process much more robust, because in this case the scanner projection plane can be found and utilized. However, this assumption restricts the applicability of the system to objects, which are scanned by laser range devices where the scanning direction is known. This method is based on defining a volumetric distance function from a set of range images and approximating a surface which is a zero set of this function.

As noted in Ref. [15], the major obstacle in surface reconstruction is consistent normal orientation. Contemporar- scan device is capable of associating a normal direction with each sample point. Surface approximation can be significantly improved by utilizing this additional information. A number of works make use of normal direction information and are related to our approach, some of which are mentioned below. Bernardini et al. [4] proposed a Delaunay triangulation-based method called the ball-pivoting algorithm. In this method, triangles are created incrementally by pivoting a ball with user-specified radius around sampled points. In order to correctly recover the topology of an object, this method utilizes a normal consistency criterion. The same criterion is used by our approach in Section 3.4. As with most reconstruction methods in the field of computational geometry, this method interpolates the data points. This becomes a drawback when the data contains a significant amount of noise. Recently, Kobelt et al. [17] proposed an improved surface extraction technique based on feature sensitive sampling. Feature location is estimated using the normal direction. In this work, the authors extended the distance function notation from the scalar to the vector field and proposed an improved version of the Marching Cubes algorithm. The resulting triangular mesh very closely approximates the sharp features of an object, though it still requires optimization because of bad aspect ratio triangles. Ohtake and Belyaev [20] proposed dual/primal optimization for improving meshes produced by the Marching Cubes algorithm, especially in high curvature regions. This method is based on projecting the reconstructed mesh vertices on surface tangent planes using the normal direction. It cannot be directly utilized for surface reconstruction, but rather for optimizing low quality meshes reconstructed by other algorithms.

3. The approach

This research proposes a fast and robust approach to initial mesh reconstruction from unorganized point cloud data. The goal is to improve the geometry approximation, especially at high curvature regions, and to improve the quality of the mesh during this early phase of the reconstruction process. The advantages of the proposed method can be summarized as follows:

- Produces high quality meshes for real data.
- Reconstructs the meshes in real time.
- Preserves sharp features.
- Reconstructs multiresolution meshes.
- Reconstructs meshes with complex topology.
- Bounds the reconstructed surface error.
- Is simple to implement.

While previous works partially fulfilled these requirements, our approach meets all of them simultaneously. Another contribution of this work is that the proposed method can be applied in real time on a common hardware configuration. This is extremely important for bringing RE technology to the designer’s desktop or even for home users.

Our approach is based on connectivity graph approximation from a Hierarchical Space Decomposition Model (HSDM) and facet reconstruction. Unlike other methods [3, 7, 15], we do not define a distance function from sampled points and extract its zero contour by the Marching Cubes algorithm. Instead, we directly reconstruct the approximating surface of the sampled points. Meshes produced by our algorithm have almost equilateral facets, and are usually better than meshes produced by the Marching Cubes algorithm. In general, dual meshing techniques tend to produce meshes with better aspect ratios, because vertices have more degrees of freedom to move inside a voxel and are not constrained to lie on the voxel edges only. For the same reason, dual approximations are capable of better preserving the behavior of the original surface, which is of course unknown. Fig. 1 illustrates this observation by a 2-dimensional example.

The proposed method is described in two stages. The first stage, the core of the method, is composed of mesh reconstruction from a cloud of points without any additional information. The second stage is an extension of the method, incorporating information regarding normal directions. Each stage consists of three phases, as presented in Fig. 2 (1) HSDM construction from a cloud of points, (2) surface extraction, and (3) feature classification.
The following discussion is organized as follows: Sections 3.1–3.3 describe the basic method for a cloud of points; in Section 3.4, the method is extended for points with normals data; in Section 4, the feasibility of the method is demonstrated on complex objects; Section 5 provides a complexity analysis of the method; Section 6 includes a summary and conclusions.

3.1. Phase 1—hierarchical space decomposition

Contemporary digitizing scanning devices are capable of generating huge clouds of points, sampled from the boundary of a 3D object. In the case of unordered sampled points, processing will lead to $O(n^2)$ time complexity, which is unsuitable for real-time applications. In the proposed approach, a cloud of points is inserted into the Hierarchical Space Decomposition Model (HSDM) in order to overcome this obstacle. As a result, the cloud is aggregated into voxels. This procedure can be considered a sorting of the point set into a 3D grid. The 3D grid is divided recursively, and every voxel is stored as a node in the Maximal Division Classical Octree (MDCO) [12]. This grid structure is uniform.

Every octree voxel $v$ referred to as a Voronoi region of an unknown site. Thus we associate a single vertex with each voxel, contrary to the Marching Cubes family algorithms, which associate a set of faces with each voxel. Optimally, the geometric entity associated with the voxel should be dictated by the surface behavior inside the voxel. This idea motivated Brunet and Ayala [6] to introduce an Extended Octree notation. In this approach, every voxel of the Extended Octree is classified into one of three categories: face, edge and vertex node, depending on the features that persist in the voxel. However, in our case the surface is unknown; consequently, this type of classification cannot be used because the location of the features is unknown a priori. Instead we chose the most conservative but universal alternative—classifying all the voxels as vertex nodes. As a result, in the surface extraction phase, voxels will be replaced by vertices.

3.2. Phase 2—surface extraction surface

In this phase a piecewise linear approximation of the object’s surface is extracted from the HSDM constructed in Phase 1. The proposed surface extraction method consists of two steps: (1) connectivity graph extraction and (2) facet reconstruction.

3.2.1. Connectivity graph extraction

A Connectivity Graph $G$ vertices approximates the Riemannian Graph [14] of an unknown object surface $O$. This structure can be extracted from the HSDM by calculating its dual graph, namely, by connecting vertices associated with carefully chosen neighboring Octree voxels. In order to implement the above, the following steps must be applied on each voxel: (1) calculating the associated vertex position, (2) finding neighboring voxels and (3) connecting vertices associated with neighboring voxels.

The position of $G$ vertices defines the geometry of a surface. In the previous phase, we mapped all the Octree voxels to vertices. The position of a vertex associated with a voxel is defined as a centroid of the sampled points lying inside the voxel. This clustering technique was introduced by Rossignac and Borrel [22] for complex scene simplification.

Definition 3.1. Let $v$ be a cubic voxel and $P_v = \{p_1, ..., p_n\}$ is a set of points such that $P_v \subseteq v$. Then $c_v$ is defined as

$$c_v = \frac{\sum_{i=1}^{n} p_i}{n} \quad (1)$$

Definition 3.2. Let $\mathcal{F} = \{v_i\}$ be an Octree, and $\mathcal{P}_v$ is a cloud of points such that $\mathcal{P}_v \subseteq \mathcal{F}$. Connectivity Graph $G$ defined as $G = (V, E)$, where $V = \{c_v\}$ and $E = \{c_{v_i}, c_{v_j}\}$.

The proposed connectivity reconstruction method is based on the fact that a neighborhood of any regular point on a 2-manifold surface can be represented by a pair of intersecting curves. Consequently, for each vertex that represents a regular point, the algorithm has to detect neighbors in four directions. For singular points like corners

---

**Fig. 1.** Curve reconstruction: original curve is in red dots (a) approximation with Marching Cubes, (b) approximation with Connectivity Graph.

**Fig. 2.** The proposed approach.
(Fig. 3(d)), only three neighbors would be sufficient. For each voxel \( n \), the algorithm starts from detecting face neighbors \( v_6 \)-adjacent. Then, three possible configurations are considered:

1. \( v_6 \)-adjacent > 4: \( n \) is identified as interior and to be ignored by the surface extraction procedure.
2. \( v_6 \)-adjacent = 4 (Fig. 3(a) and (b)): a connectivity of \( n \) completely defined by \( v_6 \)-adjacent and no more neighbors should be found.
3. \( v_6 \)-adjacent < 4 (Fig. 3(c) and (d)) edge neighbors \( v_{18} \)-adjacent must be considered. From the found \( v_{18} \)-adjacent algorithm choose only those who have no common face neighbors with \( n \).

The Octree data structure allows efficient neighbor finding. Due to this feature, the current step can proceed in real time. After the neighbor-finding procedure is performed, each vertex is connected with vertices associated with its neighboring voxels. Then, the resulting edges are added to the connectivity graph. In this way, a Doubly Connected Edge List (DCEL) is constructed, because every edge in the graph is bi-directional. An important property of this graph can be observed in Fig. 4(c).

**Observation 3.1.** Vertices \( V \) and edges \( E \) of the Connectivity Graph \( G = \{ V, E \} \) lie on or near an unknown object surface \( C \).

### 3.2.2. Facet reconstruction

Each edge of a connectivity graph \( G \) associated with a facet, which is a piecewise-linear approximation of the unknown surface. This reconstruction is accomplished using the minimal loops strategy, namely, by starting from an arbitrary edge and trying to enclose the facet with a minimal number of edges. The proposed facet reconstruction algorithm can be summarized as follows (Fig. 5).

(a) Choose an arbitrary edge and assign it as current_edge
(b) Proceed to a neighbor edge next_edge that fits the following conditions
   - next_edge is not a TWIN(current_edge).
   - current_facet with next_edge stays convex and nearly planar.
   - The Euclidean distance between the ENDVERTEX(next_edge) and the STARTVERTEX(init_edge) is minimal.
(c) Add the current_edge to the current_facet and proceed to the next edge.
(d) If the current_facet is not closed, go to step (b).
(e) Add the current_facet to the list of facets.
(f) Recursively apply the reconstruction process for TWINS() of the current_facet edges.

In general, the facet reconstruction algorithm creates non-planar, bi-linear facets. However, these facets must be almost planar. The coplanarity threshold can be controlled by the user. As can be seen from the described algorithm (see step (f)), facets are reconstructed continuously, through the neighboring edges only. By definition, a DCEL can include adjacent facets with consistent orientation only. Consequently, normal consistent orientation is imposed naturally. Moreover, this method is insensitive to sharp edges, because the normal direction is not propagated explicitly by checking the sign of the scalar product of the adjacent facets, as proposed in Ref. [15]. Rather, it is propagated implicitly by preserving adjacent facet
orientation consistency. Fig. 6 presents neighboring facets with obtuse and acute dihedral angles. In both configurations, orientation of the facets stays consistent, while the normals in Fig. 6(b) are in opposite directions.

3.2.2.1. False facets. During the facet reconstruction process described above, facets might appear which exist in the connectivity graph but do not belong to the original surface. These facets are referred to as false facets. Fig. 7(a) presents examples of false facets in the mechanical part model. We have tested several techniques for false facet elimination. These techniques are listed below:

Hit test. False facets are basically interior facets in the sense that they lie inside an unknown object and not on its surface. Therefore, a false facet can be detected by casting a ray in the facet normal direction. If the ray hits neighbor voxels, the facet should be rejected.

Distance criterion. A false facet occurs in an ambiguous configuration, where several facets can be associated with an edge. It is appropriate to assume that a true facet lies closer to the sampled points than a false facet. Consequently, a true facet can be detected by comparing the maximal deviation of facets from the close points. Close points by definition lie in voxels intersected by the facets.

Angle comparison. As in the previous test, here again we consider a case where several facets can be created. The true facet has to be the outer one. Thus, it can be detected by calculating angles between the facet previously created and next facet candidates (Fig. 7(b)).

The surface extraction procedure described in this section can be applied to any desired level of detail utilizing HSDM. Fig. 8 presents meshes extracted from different levels of Octree. As can be observed, the topology is simplified, extracting a mesh from low Octree depth.

3.3. Phase 3—feature extraction and classification

In order to complete the boundary representation created in the previous phase, a normal is associated with every mesh vertex. A naive approach would be to assign to every vertex the average normal of the neighboring facets:

$$\hat{\mathbf{N}}_v = \frac{1}{n} \sum_{j=1}^{n} \mathbf{N}_j,$$

where $n$ is the number of neighboring facets. However, this approach leads to smoothing of sharp features (Fig. 9).

Consequently, sharp features must be detected and preserved. This idea is motivated by the anisotropic diffusion approach, which is commonly used in image processing [21]. The standard sharp edges detection procedure is based on the dihedral angle test. This procedure is performed in three steps: (1) calculate the angles between neighbor facet normals, (2) define the sharpness threshold, and (3) mark the sharp edges. Once the sharp edges have been detected, the vertices are classified according to a scheme proposed by
Bernardini et al. [3]. In this scheme, a vertex can have one, two or three normals, depending on its type. Fig. 10 presents these vertex types.

3.4. Extension for point clouds with normals

This paper focuses on mesh reconstruction from an unorganized point cloud. However, there are many objects, which cannot be recovered from scattered data only. These objects are very close to self-intersection; consequently, additional information is required in order to reconstruct their boundary surface properly. An aircraft wing which is relatively thin is a good illustrative example of such an object.

Contemporary 3D scanning devices are capable of providing a normal direction with every point sampled from a surface. This additional information can be easily incorporated in the proposed reconstruction algorithm. In such a case, the aggregation of points into HSDM is based not on the Euclidean distance only, but also on a consistent normal orientation. Namely, we check whether all the points associated with a voxel have a consistent normal. If so, this voxel remains as is; otherwise the set of points associated with the voxel must be subdivided into smaller subsets, until the condition of consistent normals is fulfilled. In our cases, the number of these subsets was at most four. Fig. 13 demonstrates reconstruction of the sampled toy airplane.

4. Examples

We have applied the proposed surface reconstruction method on several complex objects received from different sources. The mechanical part, car, and oil pump models were downloaded from the Hugues Hoppe web site [13]; the toy airplane model was scanned with a 3D scanner that provided the normal direction with each sampled point.

For all these models, meshes with correct topology were reconstructed in real time. The Hausdorff distance between the data points and the reconstructed mesh is bounded by a given voxel size. This measure, normalized by a bounding cuboid size, is chosen as an error estimator. The error can be controlled by the voxel size requirement. Table 1 presents the performance of the proposed method on an Intel Pentium 4, 1.6 GHz (Figs. 4, 11, 12 and 13).

5. Complexity analysis

One of the major advantages of the proposed surface reconstruction approach is its low computational complexity. In this section the phases of the algorithm are analyzed, and the time and space complexities are computed.

The first phase of the proposed method is hierarchical space decomposition. In this phase, data points are inserted into an Octree.

**Theorem 5.1.** An Octree of depth \( d \) storing a set of \( n \) points has \( O(dn) \) nodes and can be constructed in \( O(dn) \) time.

The proof is given in Ref. [8]. In our implementation, the depth of the Octree is limited (usually \( d < 10 \)), while \( n \approx 10^7 \); this implies \( d \ll n \). Consequently, the time and space complexity of the Octree construction is nearly linear with a number of sampled points.

The next phase of the reconstruction process is connectivity graph extraction. This phase involves a neighbor-finding procedure.

**Theorem 5.2.** Let \( \mathcal{F} \) be an Octree of depth \( d \). The neighbor of a given node \( v \) in \( \mathcal{F} \) in a given direction can be found in \( O(d) \) time.

<table>
<thead>
<tr>
<th>Object</th>
<th>( n )</th>
<th>( d )</th>
<th>( e ) (%)</th>
<th>( t ) (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanical part</td>
<td>4102</td>
<td>4</td>
<td>2</td>
<td>0.1</td>
</tr>
<tr>
<td>Car</td>
<td>20,623</td>
<td>5</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>Toy airplane</td>
<td>117,152</td>
<td>6</td>
<td>1.14</td>
<td>3.9</td>
</tr>
<tr>
<td>CAD model</td>
<td>166,087</td>
<td>6</td>
<td>1.4</td>
<td>5.98</td>
</tr>
</tbody>
</table>

\( n \), Number of points; \( d \), Octree depth; \( e \), error; \( t \), execution time.

Fig. 9. (a) Isotropic smoothing, (b) anisotropic smoothing.

Fig. 10. Feature classification: (a) smooth vertex, (b) sharp edge, (c) sharp corner.
Fig. 11. Surface reconstruction examples: car (a) point cloud, (b) reconstructed mesh, (c) point cloud, (d) reconstructed mesh.

Fig. 12. Reconstruction of mechanical part with complex topology: (a) reconstructed mesh, (c) recovered features.
The proof is given in Ref. [8].

**Theorem 5.3.** Let $T$ be an Octree of depth $d$. Graph $\mathcal{G}$ connecting leaf nodes of $T$ can be constructed in $O(n)$ time and will consist of $O(n)$ edges.

**Proof.** Assuming that Octree depth $d$ is bounded, the neighbor-finding procedure is performed in $O(1)$ time. This procedure is applied for each Octree leaf node. From Theorem 5.1 we know that the number of Octree nodes is $O(n)$. Thus, the resulting time complexity of connectivity graph construction is $O(n)$.

The connectivity graph consists of vertices and edges. The number of vertices is equal to the number of Octree leaf nodes $O(n)$. Due to the fact that this graph is a Riemannian Graph of an unknown 2-manifold surface, the Euler theorem is valid in this case. Although the graph is not triangular, the linear relation between vertices, edges and faces still holds. Consequently, the space complexity of the connectivity graph is $O(n)$. This concludes the proof of Theorem 5.3.

The next phase of the algorithm is facet reconstruction from the connectivity graph. This procedure is carried out by recursively scanning the connectivity graph edges. The rank of the graph vertices is bounded ($\approx 4$). As a result, the next edge is found in $O(1)$ time. From Theorem 5.3 the number of edges is $O(n)$; thus, facet reconstruction time complexity is $O(n)$. Again applying the Euler theorem yields the space complexity of the current phase, $O(n)$.

By summing up all phases, we conclude that the time and space complexity of the overall reconstruction process is $O(dn)$. Results of this section are summarized in Table 2.

### 6. Summary and conclusions

We have presented a new fast approach for surface reconstruction from unorganized points. Our method is based on extraction of a connectivity graph from the Hierarchical Space Decomposition Model (HSDM) and facet reconstruction. Due to the efficiency of the Octree, reconstruction is performed in real-time (Table 1). Therefore, this representation is suitable for processing large-scale 3D data. This inherent capability is very important in distributed CAD systems, where surface reconstruction, rendering and modeling must be performed efficiently and in real time. The reconstructed surface is approximated by a mesh composed of bi-linear facets. These facets have a good aspect ratio and are quadrilateral almost everywhere. The proposed method is very flexible; consequently, additional data, such as normals, can be incorporated straightforwardly.

### Acknowledgements

This research has been supported in part by the CONSIST project of the Samuel Neaman Institute-Technion as part of the Magnet Consortium of the Israeli Ministry of Industry and Commerce.
References


Sergei Azernikov received a BSc degree in Mechanical Engineering from the Technion, Israel Institute of Technology (1998). Since then he has been involved in international research projects with universities in Korea and France. Mr Azernikov is currently on a direct doctoral track toward the DSc degree in Mechanical Engineering at the Technion. His primary research interests include volumetric modeling, 3D reconstruction and meshing.

Anath Fischer is an Associate Professor in the Department of Mechanical Engineering, Technion-Israel Institute of Technology (IIT). She received a BSc (1985) in Computer Science and an MSc (1987) and PhD (1992) in Mechanical Engineering from the Technion. From 1992 to 1994, she was a postdoctoral associate in the Department of Mechanical Engineering, Cornell University, USA. Prof. Fischer is currently head of the Laboratory for Computer Graphics and head of the Technion-IBM-Dassault Systems Product Lifecycle Management Competency Center (PLMCC). Her main interests include geometric modeling for CAD/CAM, PDM/PLM applications, physical-based systems, collaborative design, reverse engineering and rapid prototyping.