Robust Reconstruction of Watertight 3D Models from Non-uniformly Sampled Point Clouds Without Normal Information

Alexander Hornung† and Leif Kobbelt‡

Computer Graphics Group, RWTH Aachen University

Abstract
We present a new volumetric method for reconstructing watertight triangle meshes from arbitrary, unoriented point clouds. While previous techniques usually reconstruct surfaces as the zero level-set of a signed distance function, our method uses an unsigned distance function and hence does not require any information about the local surface orientation. Our algorithm estimates local surface confidence values within a dilated crust around the input samples. The surface which maximizes the global confidence is then extracted by computing the minimum cut of a weighted spatial graph structure. We present an algorithm, which efficiently converts this cut into a closed, manifold triangle mesh with a minimal number of vertices. The use of an unsigned distance function avoids the topological noise artifacts caused by misalignment of 3D scans, which are common to most volumetric reconstruction techniques. Due to a hierarchical approach our method efficiently produces solid models of low genus even for noisy and highly irregular data containing large holes, without loosing fine details in densely sampled regions. We show several examples for different application settings such as model generation from raw laser-scanned data, image-based 3D reconstruction, and mesh repair.

1. Introduction
The high quality reconstruction of a proper, watertight surface mesh from scattered point samples remains a difficult problem in many areas of computer graphics, including laser-scanning or image-based surface reconstruction techniques as well as repairing non-manifold or topologically noisy meshes.

Most previous work can be classified into computational geometry approaches based on Voronoi diagrams and global volumetric reconstruction techniques based on signed distance functions. Voronoi based approaches reconstruct a mesh directly from the input samples, with the particular strength of reconstructing even fine surface details. However, it is generally difficult to guarantee the reconstruction of a smooth and manifold surface, especially in the presence of noise and for varying sampling density.

Volumetric methods on the other hand attempt to reconstruct a signed distance function to the point cloud samples, and then reconstruct the zero level-set using, e.g., Marching Cubes [LC87]. The generation of the signed distance function however requires that the unstructured cloud of input points comes with consistently oriented normal information. This, however, is known to be one of the most critical steps in the reconstruction pipeline. The derivation of consistent normals from a point cloud poses a number of significant conceptual and computationally intensive problems, especially in the presence of noise, non-uniform sampling, or thin features. As a consequence, methods based on signed distance functions generally cannot guarantee that the resulting model is of the lowest possible genus. For instance, misaligned or noisy 3D scans are known to lead to severe topological noise artifacts (cf. Fig. 1 and 14).
These artifacts are a fundamental problem of methods based on extracting the zero level-set of a signed distance function. An unnecessarily high genus prevents the possibility of immediate post-processing such as mesh decimation, hence these techniques generally require subsequent post-processing [ESV97, GW01, NT03, WHDS04] for artifact removal or general mesh repair [BNK02, Ju04, BPK05].

In this paper we present a robust algorithm to overcome these drawbacks. In contrast to the above mentioned approaches, our method reconstructs the surface from a volumetric unsigned distance function, which represents the probability that the surface passes through a given voxel. Since the unsigned distance function does not carry information about the local surface orientation we are able to process input data consisting solely of 3D sample positions without any normal information. Moreover, since the surface extraction does not depend on a sign-change of the implicit representation anymore, our method is immune to noisy and non-uniformly distributed samples (cf. Fig. 1). As an important consequence our method produces meshes of low genus without the small-scale topological artifacts.

The particular contribution of this paper is a method to compute an unsigned distance function from pure point cloud data, from which a closed surface can be extracted via graph-cut based energy minimization. We show how this algorithm can be embedded into a hierarchical framework allowing for efficient processing of highly non-uniformly sampled input data with large gaps, without loosing fine details in densely sampled regions. Finally we present a new algorithm to convert the graph-cut representation of the surface into a smooth and guaranteed watertight triangle mesh.

2. Related Work

Previous work on surface extraction from point clouds can be roughly classified into the following approaches.

Voronoi based techniques such as [ABK98, ACK01, BC02, DG03] have the advantage of computing output meshes with a complexity in the order of the input data, and produce good results for data sets with known sampling density. Wrapping approaches such as [BMR∗99] provide a good local feature preservation. For non-uniformly sampled or noisy input data containing outliers, however, both types of approaches often cannot guarantee the reconstruction of a globally optimal, watertight surface. Improvements in these fields concerning noisy input data and outliers have been achieved recently in [MAVdF05, SFS05].

Methods based on deformable models for point cloud reconstruction have been presented in [EBV05, SLS∗06]. They solve the problem of computing a watertight surface by incrementally deforming an initial mesh along an energy field induced by the point cloud. Although they guarantee watertight reconstructions, they have the potential problem of creating overly smoothed surfaces since it is often difficult to find appropriate surface tension parameters.

Most related to our work are approaches such as [HDD∗92, CL96, ABCO∗01, DMGL02, OB∗03, OBS04] which reconstruct the unknown surface as the zero level-set of a signed distance function. These methods, however, often rely on accurate normal orientation and fairly uniform sampling densities, which are both requirements generally not met by real world data sets. Furthermore they can be quite sensitive to noise or outliers, e.g., for badly aligned scan patches they tend to introduce topological artifacts such as handles or bridges due to spurious zero crossings of the signed distance function (cf. Fig. 1). This often leads to reconstructions with significantly increased genus. Recently, these issues have been addressed in [FCOS05, Kaz05].

To summarize, most of the above methods have in common that it is generally difficult or even impossible to generate proper meshes from highly non-uniformly sampled point cloud data without reliable normal information. We explicitly address these problems in our work.

Recently research on combinatorial energy minimization has shown that globally optimal solutions to discrete volumetric segmentation problems can be found efficiently by reformulating them into a maximum flow / minimum cut problem of a specific spatial graph structure [BK03, KB05]. Applications using graph cuts have been presented for problems such as image segmentation [LSGX05] or 3D stereo reconstruction [HK06, VTC05]. We will show in this work, how our method presented in [HK06] can be extended to the problem of point cloud reconstruction.

© The Eurographics Association 2006.
Vext and the inner boundary $V_{int}$. We embed a spatial graph approximation of the unknown surface with maximum confidence, i.e., for a faithful approximation of the unknown surface, noise or non-uniform sampling of the input samples do not significantly influence the quality of the reconstruction. Section 4 describes the steps for computing $\phi$ in detail.

Given such a grid of confidence weighted voxels we want to extract a minimal subset $S_{opt} \subseteq V$, representing a closed surface with maximum confidence, i.e., for a faithful approximation of the unknown surface $S$ the sum of unsigned distance values has to be minimized $\sum_{v \in S_{opt}} \phi(v) \rightarrow \min$. Methods for iso-surface extraction are not suitable to reconstruct a surface represented by such a probability distribution. Previous work [BK03] has shown that similar types of combinatorial optimization problems involving the minimization of certain energy functionals can be efficiently solved by transforming them into a max-flow / min-cut problem of an embedded spatial graph $G$. Our specific problem formulation for surface reconstruction from a set of confidence weighted voxels is highly related to our previous work on image based stereo reconstruction [HK06]. We will show in Sect. 5 how this graph based algorithm can be adapted to the setting of this work.

For non-uniformly sampled point clouds it is generally difficult to estimate an optimal volumetric grid resolution such that holes in sparsely sampled areas can be efficiently detected and closed without losing details in densely sampled regions. On the other hand, simply computing the above mentioned unsigned distance function and surface extraction on a high resolution grid would result in a significant computational overhead. In Sect. 6 we show how to integrate the confidence estimation and graph-based surface extraction into a hierarchical framework such that the above mentioned problems are effectively resolved.

Once the desired target resolution is reached the voxel based representation of the surface $S_{opt}$ has to be converted into a triangle mesh to be usable for further geometric processing steps. Sect. 7 describes a new algorithm to generate a smooth and manifold mesh derived from $S_{opt}$ and the min-cut edges of $G$.

3. Overview

In this section we outline the four central ideas and corresponding processing steps of our algorithm to achieve our goals. The different phases are then described in detail in the following sections.

The input to our algorithm is a point cloud, i.e., a set of scattered 3D positions $p \in P$ of a surface $S$. Our basic idea is to first derive a confidence map in the vicinity of these point samples (similar to Pauly et al. [PMG04]), representing the probability that the unknown surface passes through a certain part of 3D space. We compute these confidence values as an unsigned distance function $\phi : v \rightarrow c \in [0,1]$ over the voxels $v \in V$ in a volumetric grid, where $c$ can be interpreted as the pseudo-distance of a voxel to the closest point sample $p$ (cf. Fig. 2). Since this representation, unlike signed distance functions, does not imply any local orientation properties of the unknown surface, noise or non-uniform sampling of the input samples do not significantly influence the quality of the reconstruction. Sect. 4 describes the steps for computing $\phi$ in detail.

For non-uniformly sampled point clouds it is generally difficult to estimate an optimal volumetric grid resolution such that holes in sparsely sampled areas can be efficiently detected and closed without losing details in densely sampled regions. On the other hand, simply computing the above mentioned unsigned distance function and surface extraction on a high resolution grid would result in a significant computational overhead. In Sect. 6 we show how to integrate the confidence estimation and graph-based surface extraction into a hierarchical framework such that the above mentioned problems are effectively resolved.

Once the desired target resolution is reached the voxel based representation of the surface $S_{opt}$ has to be converted into a triangle mesh to be usable for further geometric processing steps. Sect. 7 describes a new algorithm to generate a smooth and manifold mesh derived from $S_{opt}$ and the min-cut edges of $G$.

4. Surface Confidence Estimation

Initially we insert each 3D sample $p \in P$ into a volumetric grid $V$, resulting in a sparse set of occupied voxels $v$ (cf. Fig. 2, upper left). As mentioned above the probability or confidence that a voxel $v$ is part of the unknown surface can be approximated by an unsigned distance function $\phi$ over $V$. To compute $\phi$ in the vicinity of $S$ we first apply several steps of a morphological dilation operator to the 6-neighborhood of occupied voxels, generating an extended crust of voxels $V_{crust}$. The distance function $\phi(v)$ for each voxel $v \in V_{crust}$ is then computed by volumetric diffusion (cf. Fig. 2, lower left).

For the graph-based surface computation we have to ensure that the computed crust is watertight (i.e., 6-connected) and has two interfaces $V_{ext}$ and $V_{int}$, to an outer and inner volumetric component, respectively (cf. Fig. 3 b). In most cases the number of necessary dilation steps for computing this crust can be computed robustly with a simple heuristic. By flood-filling unoccupied voxels from the outer boundaries of $V$ we can easily determine the current number of different volumetric components separated by $V_{crust}$. Initially we generally have only one (outer) component. This number increases during the dilation process as the crust grows.
Sect. 6), since we generally start at low volumetric resolution, which is computationally irrelevant in our overall hierarchical setting (cf. Sect. 6). Please note that this repeated flood-filling and dilation process is completed by iterative averaging over the 6-neighborhood \( N(v) \) (in \( V_{\text{crust}} \)) of a voxel

\[
\phi(v) = \frac{1}{|N(v)| + 1} \left( \phi(v) + \sum_{u \in N(v)} \phi(u) \right),
\]

while keeping \( \phi(v) = 0 \) fixed for voxels containing surface samples (cf. Fig. 3 c). The overall algorithm is not very sensitive to the number of diffusion steps. In fact, a valid surface can already be computed after the initialization of \( \phi \) without any diffusion. However, the surface becomes slightly smoother with more diffusion steps, and we additionally show in Sect. 7 how the unsigned distance values allow for confidence weighted mesh smoothing of the extracted mesh. In our results presented in Sect. 8 we simply use three diffusion steps for all reconstructed models.

We also experimented with initial confidence values computed from the sample density within a voxel instead of setting all occupied voxels to \( \phi(v) = 0 \). However, the current approach has the advantage of handling strongly non-uniformly sampled regions in a more uniform manner. Similarly, keeping \( \phi(v) = 0 \) fixed for voxels containing surface samples instead of including them in the diffusion process preserves fine details more faithfully.

5. Graph-based Surface Extraction

Given a function \( \phi \) of surface confidence values in a volumetric region \( V_{\text{crust}} \), we have presented a method [HK06] to compute a closed 2-manifold surface \( S \) embedded in \( V_{\text{crust}} \) which minimizes an energy functional

\[
E(S) = \int_S \phi(x) \, dx + \int_\partial S a \, dS,
\]

with \( a \) being a regularizing parameter of the surface tension. Our algorithm efficiently computes a set of surface voxels \( S_{\text{opt}} \subseteq V_{\text{crust}} \), which minimizes the discretization of \( E(S) \) by computing the minimum cut of a weighted graph \( G \) embedded into the volumetric grid. For completeness we briefly describe the graph construction and edge weight computation.

In \( G \), a graph node is associated with each voxel face, and a weighted graph edge is created for each voxel edge, such that each voxel contains an octahedral subgraph (cf. Fig. 4 a). An edge weight \( w(v) \),

![Figure 3: An example in 3D for point cloud dilation and confidence estimation for the dragon model. Image (a) shows the initial set of occupied voxels which contain a sample of \( P \). In (b) the dilated voxels generate a watertight crust \( V_{\text{crust}} \) in which the unknown surface is contained. Note that the outer boundary is genus 0, while the inner component is split into several components. (c) shows a section of the unsigned distance function in the head region after the diffusion process. Darker values indicate higher surface confidence. The set \( S_{\text{opt}} \) of surface intersected voxels after the graph-cut computation is shown in (d). The resulting model has the correct genus 1.](Image 168x528 to 258x598)

![Figure 4: Because of the duality of the voxel and the octahedron, a cut through the octahedral subgraph corresponds to a split of the voxel faces into an exterior and an interior component (a). The splitting surface is visualized in (b).](Image 52x536 to 142x674)
depending on the voxel’s unsigned distance value $\phi(v)$ and the constant $a$, is assigned to all edges of the corresponding subgraph

$$w(v) = \phi(v)^3 + a.$$  (3)

The exponent $s$ can be used to tune the unsigned distance function to some extend, such that the maxima of the confidence values are emphasized more or less strongly. However, in all our experiments we set the parameters for the edge weight computation (cf. Eq. 3) as proposed in [HK06] to $s = 4$ and $a = 10^{-5}$. The outer and inner boundaries nodes (voxel faces) exposed at the interface to $V_{ext}$ and $V_{int}$ are connected to a sink and a source node, respectively.

The minimum cut of $G$ then yields a set of cut-edges $C$ which form a watertight, manifold separation between the sink and the source node (and hence $V_{ext}$ and $V_{int}$) and are the globally optimal solution in terms of the surface energy functional (2) (cf. Fig. 2 and 3 d), and hence can be considered as a faithful approximation to the surface $S$. The corresponding set of surface voxels $S_{opt}$ is defined by those voxels containing at least one cut edge.

6. Hierarchical Hole Filling and Detail Preservation

For high volumetric resolutions the computational complexity for generating the unsigned distance function and the graph cut computation become impractical, especially for strongly non-uniform data containing sparsely sampled regions as well as fine details. Hence we employ an iterative hierarchical framework on an adaptive volumetric grid (e.g., using an Octree), and use the surface approximation obtained on a lower volumetric resolution to constrain the crust and surface computation on the respective higher level. Starting at a low volumetric resolution allows for an efficient generation of a proper initial crust even for highly non-uniformly sampled point clouds with large gaps.

Previous work (e.g., [LSGX05]) has shown however that a simple hierarchical refinement of voxels within a fixed distance to the cut surface $S_{opt}$ potentially leads to a loss of fine details if the corresponding input data samples are not contained inside this fixed distance crust. In our application setting however, where explicit data samples are available, we can derive an efficient hierarchical algorithm which effectively avoids the above mentioned problems.

On a given volumetric refinement level $l$ we compute a surface approximation $S_{opt}$ within a crust $V^{l+1}_{crust}$ as described in the previous sections. Then, we compute a new, thinned crust $V^{l+1}_{crust}$ on the next higher resolution level by refining the surface voxels $v \in S^l_{opt}$, and applying a number of morphological dilation steps. This effectively constrains the volumetric region for surface extraction based on our current surface proxy $S^l_{opt}$. To preserve fine details represented by point samples of $P$ outside of this crust, we re-insert the corresponding input samples as occupied cells into the volumetric grid at the new resolution $l + 1$, and dilate these cells until they merge with $V^{l+1}_{crust}$ (cf. Fig. 5). In our experiments we simply used a fixed number of 3 dilation steps. The cut computation then automatically includes these voxels for the surface extraction, and fine details are preserved.

The number of dilation steps to compute the thinned crust $V^{l+1}_{crust}$ basically depends on the amount of noise of the input data $P$. Without noise two dilation step at level $l + 1$ would be sufficient assuming that the low frequency parts of the surface have been reconstructed up to voxel accuracy at level $l$. In all our experiments with real data four dilation steps proved to be a good choice. However, since all original data points outside of the crust are re-inserted and dilated in the higher resolution grid anyway, the choice of this parameter is mostly of importance for the computational performance. An unnecessarily thick crust would include too many voxels which do not contribute to the surface extraction at all, while a too thin crust would lead to a higher number of samples outside of the crust, such that a higher number of additional dilation steps is necessary.

To summarize, our hierarchical point cloud reconstruction consists of the following three iterative phases. Starting at a volumetric resolution $l$:

© The Eurographics Association 2006.
Figure 6: The computed min-cut of $G$ splits the voxels in $S_{\text{opt}}$ into exterior and interior faces (a). The set of cut-edges $C$ defines a loop of split-edges for each voxel, corresponding to a non-planar polygonal face (b). In the dual mesh, the non-planar polygonal faces are defined by cut-corners which are shared by at least 3 surface voxels. We extract the corresponding mesh by placing a single mesh vertex at the center of each voxel, and visiting the voxels associated with a cut-corner by cycling over shared cut-edges (c). Triangulating the corresponding polygons with a triangle fan yields the final mesh (d).

1. Surface confidence estimation
   - Insert points $p \in P$ as occupied voxels $v$ into the volumetric grid $V_l$.
   - Dilate these voxels to a crust $V_l^{\text{crust}}$.
   - Compute the unsigned distance function $\phi(v)$ for all $v \in V_l^{\text{crust}}$ by volumetric diffusion (cf. Eq. 1).

2. Graph-based surface extraction
   - Generate graph $G$ consisting of octahedral subgraphs for each voxel $v \in V_l^{\text{crust}}$ with edge weights according to Eq. 3. Nodes at the boundaries of $V_l^{\text{crust}}$ are connected to a corresponding terminal node of $G$.
   - Compute the min-cut of $G$ (e.g., using [BK04]), resulting in a set of surface intersected voxels $S_{\text{opt}}$ and cut edges $C$.
   - If $l$ is the target resolution then terminate and extract the final surface mesh (cf. Sect. 7).

3. Volumetric refinement
   - Refine surface voxels $S_{\text{opt}}^l$ to the next higher resolution and set $l = l + 1$.
   - Compute the new crust $V_l^{\text{crust}}$ by dilation and proceed with step 1.

Please note that the computation of the unsigned distance function and the graph cut in steps 1 and 2 are not affected by the hierarchical approach, since all voxels in the current crust $V_l^{\text{crust}}$ are at the same refinement level.

This algorithm computes a closed surface representation even from strongly non-uniformly sampled point clouds. Large gaps are effectively closed with a reasonable surface due to the surface tension in Eq. 2, and fine details are preserved due to the iterative point insertion and dilation.

Figure 7: Different cut-corner configurations for a low resolution mesh of the buddha model with 3, 4, and 5 incident cut-edges (shown in red), respectively.

7. Mesh Extraction

In the final step of the surface reconstruction algorithm we have to extract a polygonal representation from the set of voxels $S_{\text{opt}}$. Due to the duality of the octahedron and the cube, we can interpret the cut through the octahedron edges in $G$ as a cut along the cube edges ("cut-edges", see Fig. 6 a,b). By this, the global graph cut through $S_{\text{opt}}$ defines a polygonal mesh $M$ with non-planar faces, which corresponds to a closed manifold surface. The vertices of $M$ lie on the voxel corners and the mesh edges coincide with voxel cut-edges (cf. Fig. 6 b). In [HK06] an algorithm is described to extract a triangle mesh by generating a triangle fan for each polygon/voxel.

However, since confidence values are estimated per voxel and also in order to reduce the output mesh complexity, it seems more natural to extract a polygon mesh $M'$ which is dual to $M$, i.e. vertices are lying in the voxel centers and the non-planar polygonal faces correspond to voxel corners (cf. Fig. 8).

The polygonal mesh $M'$ can easily be generated by running through all $2 \times 2 \times 2$ blocks of voxels. For each block $B$ the center voxel corner corresponds to a polygon face of $M'$ if the block contains at least three voxels from $S_{\text{opt}}$ (cf. Fig. 6 c and 7). The edges for a polygonal face of $M'$ are enumerated by cycling through the voxels of the $2 \times 2 \times 2$ block. Since every voxel in $S_{\text{opt}}$ has exactly two cut-edges incident to the corner in the center of $B$, the ordering is given
by cut-edges that two neighboring voxels have in common. The following pseudo-code describes the procedure:

```plaintext
for each 2x2x2 block B with at least three voxels in Opt
  pick a starting voxel v from Opt \cap B;
  pick a cut-edge e in v adjacent to the center voxel corner c of B;
  do
    find the second cut-edge f in v adjacent to c;
    find the neighbor voxel w from Opt \cap B sharing the cut-edge f with v;
    generate a polygon edge from v to w;
    v ← w, e ← f;
  until the first voxel is reached again;
```

This code works correctly because each voxel from Opt \cap B has exactly two cut edges adjacent to the center corner and no more than two voxels share a common cut-edge. A consistent orientation of the faces can be propagated by mesh traversal. The polygonal faces may be converted into triangle fans afterwards and decimated if required.

The resulting mesh is watertight but shows grid artifacts due to the fact that we initially placed the mesh vertices at the voxel centers. However our surface confidence map \( \phi \) computed for each voxel can be applied to the mesh vertices accordingly. We can exploit this information for a confidence weighted smoothing algorithm, which allows for error bounded surface smoothing in confident surface areas, such that only grid artifacts are removed, while less confident or noisy parts of the mesh can be smoothed stronger. We implement this algorithm by applying an iterative bi-Laplacian smoothing operator [DMSB99] for each vertex \( v \):

\[
v ← v - \frac{1}{d} \Delta^2 v , \quad d = 1 + \frac{1}{n_v} \sum_{j} n_{v,j}
\]

with \( n_v \) and \( n_{v,j} \) being the valences of vertex \( v \) and its \( j \)-th one-ring neighbor. The surface confidence values \( \phi(v) \) prescribe how much every vertex is allowed to deviate from its original position during smoothing. We stop the movement of a vertex if \( \delta p < \delta v (\phi(v) + 1)^s \) is violated. \( \delta p \) is the difference between the original and the smoothed vertex position, \( \delta v \) represents the voxel size, and \( s \) allows for emphasized smoothing in inconfident regions. For all of our presented results, however, we simply set \( s = 1 \).

This algorithm computes smooth meshes while preserving the original surface approximation quality of the computed cut surface.

8. Results

In this section we present the results of our method applied to a variety of different data sets such as point clouds acquired from laser scans and stereo vision based point reconstructions as well as model repair. All reconstructions are based purely on 3D sample positions without any normal in-
The point cloud of the Max-Planck consists of a set of circularly acquired laser scans. The second image shows that top and bottom of the bust as well as some smaller areas around the ears do not contain any samples. Our method closes these holes and produces a genus 0 mesh.

The statue shown in Fig. 9 is reconstructed from raw laser-scanning data at a volumetric resolution of $1024^3$. The input point cloud contains significant noise, outliers and large gaps, e.g., at the bottom part or at the backside of the upper arm. Nevertheless our algorithm reconstructs a proper, watertight genus 0 model. This model is particularly difficult to reconstruct because of large regions with completely missing samples on the backside of the statue.

Further reconstructions from raw laser scanned point clouds are shown in Figures 10 and 11, respectively. While the Max-Planck example has a highly non-uniform sample distribution with large holes, especially at the top and the bottom, the Rings example contains significant noise and alignment artifacts.

The point cloud for the Leo (Fig. 12) as well as for the Monkey (Fig. 13) model have been acquired by image based 3D stereo reconstruction methods. The Leo has a particularly non-uniform and noisy sample distribution, with large clusters of points inside the model, and larger gaps at the tail and the legs. The Monkey model has a more uniform sampling but consists only of samples for the front side of the face. Creating watertight models from such models without at least approximate surface normals has been very challenging for previous methods.

As examples for mesh repair we applied our algorithm to the VRIP-reconstructions [CL96] of the Buddha (Fig. 5) and the Max-Planck statue (Fig. 9).
and the Dragon (Fig. 14), available at the Stanford 3D Scanning Repository. Both models have a very high genus due to topological artifacts. Our method faithfully reconstructs watertight models with low genus. For the Dragon model we also show the triangle quality generated by our meshing algorithm.

9. Conclusions

In this paper we presented a robust algorithm to reconstruct a watertight triangle mesh from point clouds without requiring normal information. It generates surfaces of low genus without the topological artifacts produced by many other techniques due to the use of an unsigned distance function and graph cut minimization for surface extraction. Our method supports highly non-uniform sample densities without losing details due to an efficient hierarchical scheme. Finally we showed how to extract a proper triangle mesh from the cut representation with just one vertex per voxel.

The resolution of our output models is currently restricted to $1024^3$ because we explicitly generate the spatial graph structure $G$ using a graph cut library [BK04], resulting in a noticeable memory overhead. However, due to the duality of the voxels and the embedded octahedral subgraphs it should be possible to alleviate this limitation by computing the cut directly on the voxel grid.

Since the surface is reconstructed at voxel accuracy, flat surface areas with a slight slope with respect to the main axes of the volumetric grid can lead to staircase artifacts, which converge only slowly to a smooth surface during our smoothing process. Although this fact does not influence the accuracy of the reconstructed model, we would like to find a solution for visually improved results, e.g., by integrating direct solvers.

Table 1: The time and space complexity for all presented reconstructions. The timings include all processing steps, from confidence estimation to mesh smoothing. However, the most significant processing time is currently used for creating the actual graph structure. As discussed in Sect. 9, we expect a much better performance by computing the cut directly on the voxel grid.

<table>
<thead>
<tr>
<th>Model</th>
<th>Resolution</th>
<th>Timings</th>
<th>Genus</th>
<th>Vertices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rings (Fig. 11)</td>
<td>$256^3$</td>
<td>45 s</td>
<td>4</td>
<td>91 K</td>
</tr>
<tr>
<td>Leo (Fig. 12)</td>
<td>$256^3$</td>
<td>48 s</td>
<td>1</td>
<td>47 K</td>
</tr>
<tr>
<td>Monkey (Fig. 13)</td>
<td>$256^3$</td>
<td>82 s</td>
<td>0</td>
<td>72 K</td>
</tr>
<tr>
<td>Buddha (Fig. 5)</td>
<td>$512^3$</td>
<td>112 s</td>
<td>10</td>
<td>264 K</td>
</tr>
<tr>
<td>Dragon (Fig. 14)</td>
<td>$512^3$</td>
<td>150 s</td>
<td>1</td>
<td>318 K</td>
</tr>
<tr>
<td>Max-Planck (Fig. 10)</td>
<td>$512^3$</td>
<td>199 s</td>
<td>0</td>
<td>320 K</td>
</tr>
<tr>
<td>Statue (Fig. 9)</td>
<td>$1024^3$</td>
<td>269 s</td>
<td>0</td>
<td>448 K</td>
</tr>
</tbody>
</table>

References


[BMR*99] Bernardini F., Mittleman J., Rush-


