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.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

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).

---

† hornung@cs.rwth-aachen.de
‡ kobbelt@cs.rwth-aachen.de

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 cre-
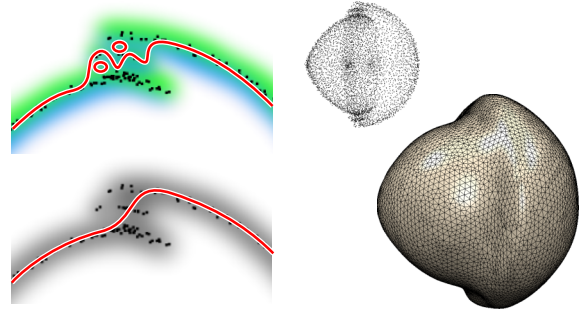

Figure 1: *The fundamental problem of surface reconstruction methods based on a signed distance function is the fact that local inconsistencies of the input data caused, e.g., by unreliable normal estimations, lead to frequent sign changes. This generally results in reconstructions of undesirably high genus, with significant topological artifacts (upper left). In contrast, our method is based on an unsigned distance function, which gracefully handles such inconsistencies (lower left). The right images show a corresponding point cloud and our genus 0 reconstruction.*

ating 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, CBC*01, ABCO*01, DMGL02, OBA*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.
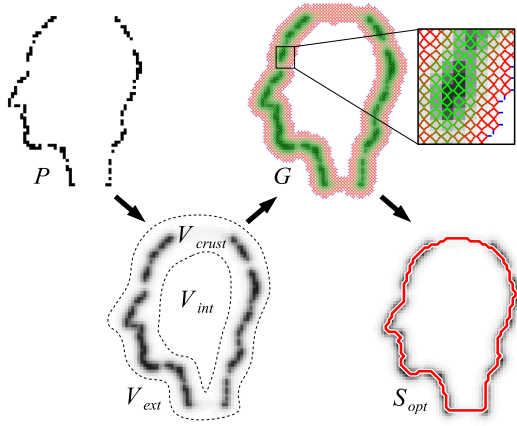
Figure 2: *This figure illustrates the point cloud reconstruction process in 2D. From left to right: Based on an input point cloud P we first compute an unsigned distance function by volumetric diffusion. The unknown surface is supposed to lie in the voxel crust V_{crust} between the outer boundary V_{ext} and the inner boundary V_{int}. We embed a spatial graph structure G within the voxel grid, with small edge weights for high confidence voxels and vice versa. The boundaries are connected a sink and a source node, respectively. Computing the min-cut of this graph yields the surface S_{opt}.*

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 position samples $p \in P$ 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 ϕ 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 $\sum_{v \in S_{opt}} \phi(v) \rightarrow min$. Meth-

ods 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 loosing 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 ϕ over V. To compute ϕ 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,
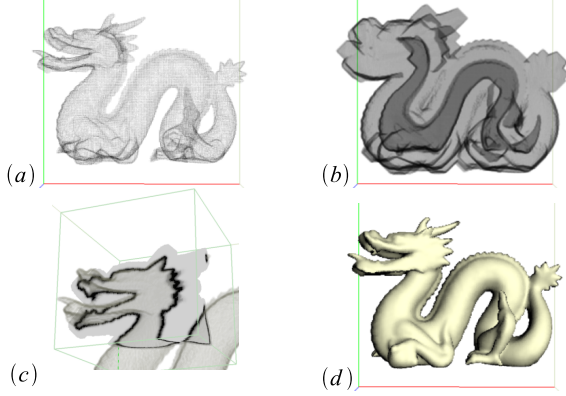
Figure 3: *An example in 3D for point cloud dilation and confidence estimation for the dragon model. We increased the crust size for visualization purposes. Image (a) shows the initial set of occupied voxels which contain a sample of P. In (b) the dilated voxels generate a watertight crust V_{crust} 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_{opt} of surface intersected voxels after the graph-cut computation is shown in (d). The resulting model has the correct genus 1.*

and eventually drops down again to one component when the "interior" of the point cloud P is full of occupied voxels. V_{int} is then simply defined by the voxels conquered during the last dilation steps (e.g., 3 in our experiments). Please note that this repeated flood-filling and dilation process is computationally irrelevant in our overall hierarchical setting (cf. Sect. 6), since we generally start at low volumetric resolutions of 64^3 or 128^3.

For point clouds covering only a part of the surface of an object (cf. Fig. 13), or objects with relatively thin, elongated features and non-uniform sampling density (cf. Fig. 12) it is sometimes not possible to compute a proper interior component V_{int}. In these cases our algorithm computes a fast approximation to the medial axis of the dilated crust by estimating normal orientations on V_{ext}, and propagating them inwards through V_{crust} by an averaging filter. For each voxel we then estimate a normal cone by collecting the normals of all 26 neighboring voxels and label each voxel as V_{int} if the opening angle of this cone lies above a threshold of $\theta = \pi/2$. The actual choice of this threshold however does not have a significant influence on the results, since we basically just want to find discontinuities in the normal field.

Concerning the diffusion process to compute a smooth distance function we first assign distance values $\phi(v) = 0$ to voxels containing surface samples p, and $\phi(v) = 1$ for the

remaining voxels in V_{crust}. The diffusion is then simply performed by iterative averaging over the 6-neighborhood $N(v)$ (in V_{crust}) of a voxel

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

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

$$E(S) = \int_S \phi(x) \ dx + \int_S a \ dS , \qquad (2)$$

with a being a regularizing parameter of the surface tension. Our algorithm efficiently computes a set of surface voxels $S_{opt} \subseteq V_{crust}$, 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.
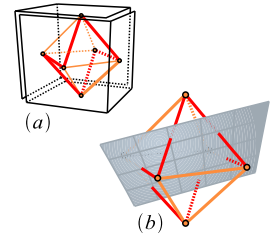
In G, a graph node is associated with each voxel face, and a weighted graph edge is created for each voxel edge, such that each voxel contains an octahedral subgraph (cf. Fig. 4 a). An edge weight $w(v)$,

Figure 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).*

depending on the voxel's unsigned distance value $\phi(v)$ and the constant a, is assigned to all edges of the corresponding subgraph

$$w(v) = \phi(v)^s + a \ . \tag{3}$$

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}^l within a crust V_{crust}^l as described in the previous sections. Then, we compute a new, thinned crust V_{crust}^{l+1} on the next higher resolution level by refining the surface voxels $v \in S_{opt}^l$, and applying a number of morphological dilation steps. This effectively constrains the volumetric region for surface extraction based on our current surface proxy S_{opt}^l. 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
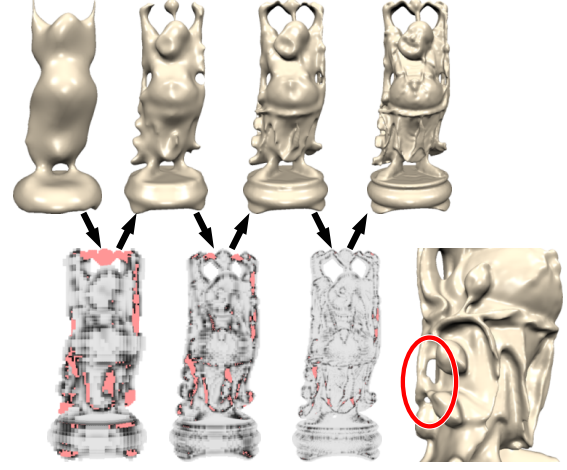

Figure 5: *Starting at a volumetric resolution of 64^3 this image sequence shows a hierarchical reconstruction of the Buddha model from the Stanford 3D Scanning Repository up to 512^3. On each level the images show the reconstructed mesh and the crust of consistency values in alternating order. Re-inserted detail samples are shown in red. Note that reconstructed thin features and the genus of the model do not depend on the respective approximation on the previous levels, but automatically adapt to the current resolution. Our algorithm reduces the original genus of the input model from >100 to 10. A few additional holes are introduced in regions, where the point samples of opposite surface sheets lie very close compared to the volumetric resolution, e.g., as shown in the lower right close-up.*

until they merge with V_{crust}^{l+1} (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_{crust}^{l+1} 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:
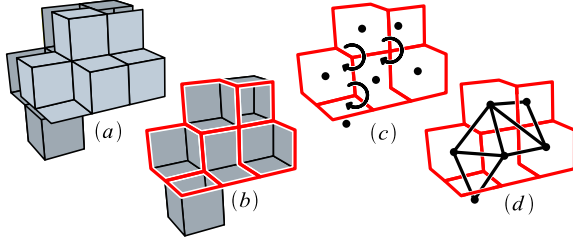
Figure 6: *The computed min-cut of G splits the voxels in S_{opt} into exterior and interior faces (a). The set of cut-edges C defines a loop of split-edges for each voxel, corresponding to a non-planar polygonal face (b). In the dual mesh, the non-planar polygonal faces are defined by cut-corners which 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_{crust}.
   - Compute the unsigned distance function $\phi(v)$ for all $v \in V^l_{crust}$ by volumetric diffusion (cf. Eq. 1).

2. Graph-based surface extraction

   - Generate graph G consisting of octahedral subgraphs for each voxel $v \in V^l_{crust}$ with edge weights according to Eq. 3. Nodes at the boundaries of V^l_{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^l_{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^l_{opt} to the next higher resolution and set $l = l + 1$.
   - Compute the new crust V^l_{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_{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.
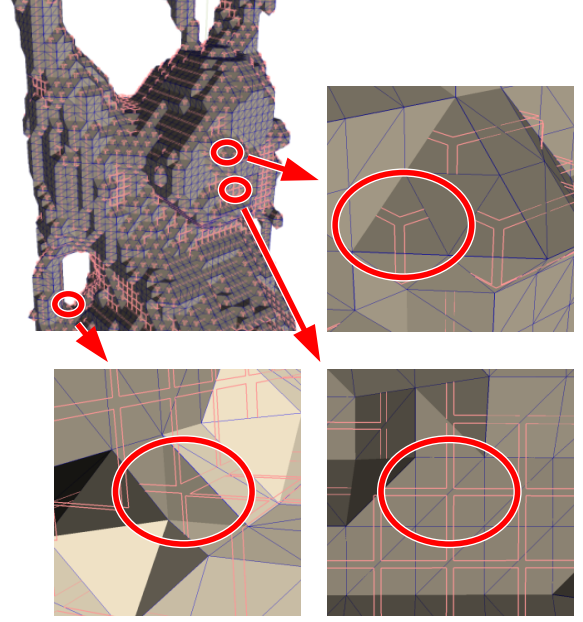

Figure 7: *Different cut-corner configurations for a low resolution mesh of the buddha model with 3, 4, and 5 incident cut-edges (shown in red), respectively.*

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_{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_{opt} defines a polygonal mesh $\mathcal{M}$ with non-planar faces, which corresponds to a closed manifold surface. The vertices of $\mathcal{M}$ 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 $\mathcal{M}'$ which is dual to $\mathcal{M}$, 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 $\mathcal{M}'$ 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 $\mathcal{M}'$ if the block contains at least three voxels from S_{opt} (cf. Fig. 6 c and 7). The edges for a polygonal face of $\mathcal{M}'$ are enumerated by cycling through the voxels of the $2 \times 2 \times 2$ block. Since every voxel in S_{opt} has exactly two cut-edges incident to the corner in the center of B, the ordering is given
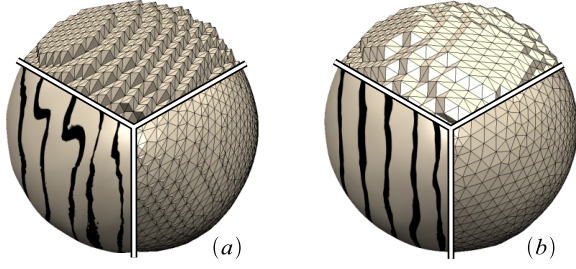
Figure 8: *This figure compares the meshing algorithm presented in [HK06] (a) to our new mesh extraction (b). Both meshes have been generated from the same graph cut of a sphere data set. The upper sectors show the mesh extracted directly from the cut edges with voxel grid discretization artifacts. The lower sectors show the vertex distribution and reflection lines after an equal number of Laplacian smoothing steps. Due to the oversampling of grid artifacts in (a) the surface smoothing for eliminating discretization artifacts converges much slower and more non-uniformly. Since our new meshing algorithm creates exactly one vertex per surface voxel we achieve a significantly better vertex distribution with a more regular mesh topology (i.e., valence 6 vertices), resulting in significantly improved smoothing convergence and lower mesh complexity.*

Figure 9: *Solid genus 0 reconstruction of a statue from non-uniformly sampled 3D points from raw laser scanned data, with significant outliers and holes. The backside of the upper arm and the lower part of the model are only partially sampled from the front, without any samples at the back of the object.*

by cut-edges that two neighboring voxels have in common. The following pseudo-code describes the procedure:

```
for each 2×2×2 block B with at least
    three voxels in S_opt
  pick a starting voxel v from S_opt ∩ 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 S_opt ∩ 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 $S_{opt} \cap B$ 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 ϕ 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
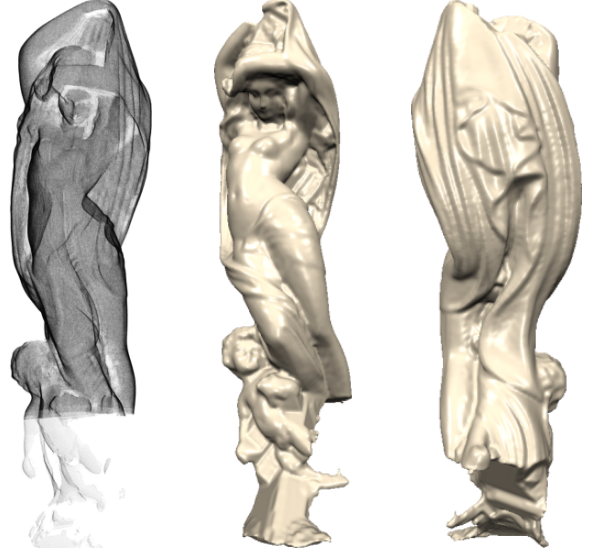
that only grid artifacts are removed, while less confident or noisy parts of the mesh can be smoothed stronger. We implement this algorithm by applying an iterative bi-Laplacian smoothing operator [DMSB99] for each vertex v:

$$v \leftarrow v - \frac{1}{d}\triangle^2 v \ , \quad d = 1 + \frac{1}{n_v}\sum_j n_{v,j} \qquad (4)$$

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. δp is the difference between the original and the smoothed vertex position, δv 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-
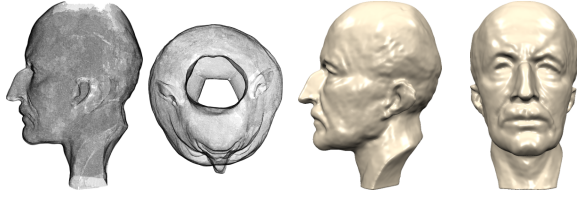
Figure 10: *The point cloud of the Max-Planck consists of a set of circularly acquired laser scans. The second image shows that top and bottom of the bust as well as some smaller areas around the ears do not contain any samples. Our method closes these holes and produces a genus 0 mesh.*
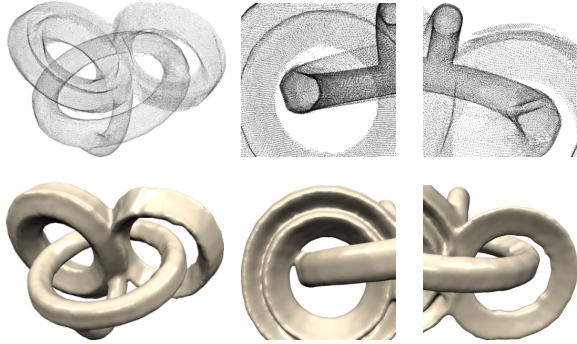

Figure 11: *Solid reconstruction of a genus 3 object from a noisy scan. Due to significant noise, however, two of the rings are merged, resulting in a genus 4 reconstruction. On the other hand, even significantly misaligned parts as shown in the right images are easily handled without producing topological artifacts.*

Figure 12: *The Leo point cloud was generated with a "Voxel Coloring" algorithm for image based 3D reconstruction [SD97]. It is quite noisy and highly non-uniform in terms of sample density and surface "thickness". Nevertheless our method succeeds in reconstructing a proper genus 1 model.*

Figure 13: *The point cloud for the Monkey model was created by an image based stereo reconstruction algorithm. Despite the fact that only samples for the front of the model are available, our algorithm is capable of computing a closed mesh. We computed the interior component with the medial axis approximation described in Sect. 4. The ears however get cut away since they would include too many inconfident voxels in particular on the back of the head, and the resulting energy (cf. Sect. 5) would be higher than the given result.*

formation. We also provide quantitative evaluations in terms of the computation performance and the resulting meshes in Table 1. All experiments were performed on a 3.2 GHz Pentium P4 with 2 Gb of main memory.

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.

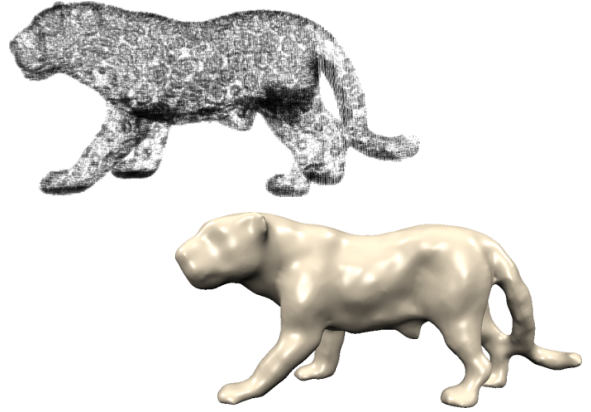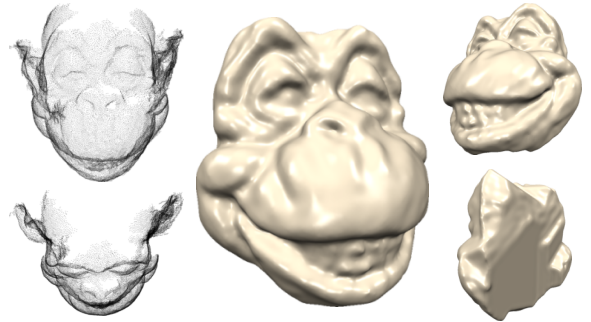The point cloud for the Leo (Fig. 12) as well as for the Monkey (Fig. 13) model have been acquired by image based 3D stereo reconstruction methods. The Leo has a particularly non-uniform and noisy sample distribution, with large clusters of points inside the model, and larger gaps at the tail and the 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)

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

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.*

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 loosing 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.
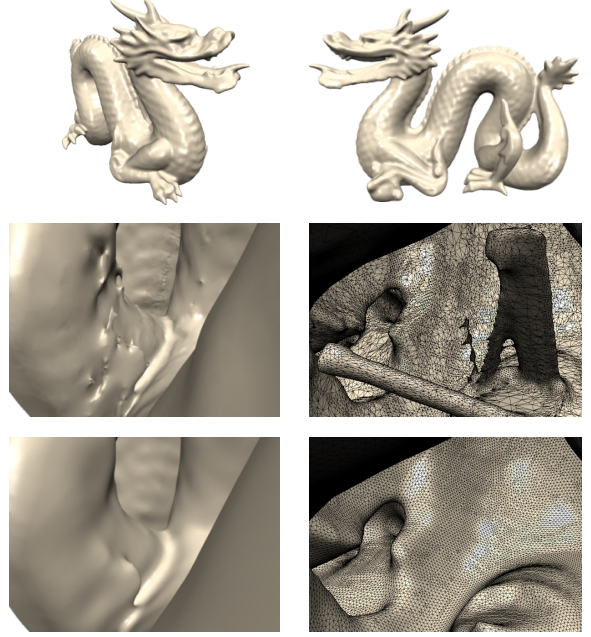
Figure 14: *The original Dragon model in the Stanford 3D Scanning Repository contains numerous small holes and topological artifacts such as bridges, resulting in a very high genus (>400). When using the 3D point samples of this mesh as input to our algorithm, all topological artifacts are removed and the resulting watertight mesh has genus 1. The close-ups compare the original backside of one of the legs and a view from inside the model to our result.*

References

[ABCO*01] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Point set surfaces. In *IEEE Visualization* (2001).

[ABK98] AMENTA N., BERN M. W., KAMVYSSELIS M.: A new voronoi-based surface reconstruction algorithm. In *SIGGRAPH* (1998), pp. 415–421.

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *Symposium on Solid Modeling and Applications* (2001), pp. 249–266.

[BC02] BOISSONNAT J.-D., CAZALS F.: Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Comput. Geom. 22*, 1-3 (2002), 185–203.

[BK03] BOYKOV Y., KOLMOGOROV V.: Computing geodesics and minimal surfaces via graph cuts. In *ICCV* (2003), pp. 26–33.

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 9 (2004), 1124–1137.

[BMR*99] BERNARDINI F., MITTLEMAN J., RUSH-

MEIER H. E., SILVA C. T., TAUBIN G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Vis. Comput. Graph. 5*, 4 (1999), 349–359.

[BNK02] BORODIN P., NOVOTNI M., KLEIN R.: Progressive gap closing for mesh repairing. In *Advances in Modelling, Animation and Rendering*, Vince J., Earnshaw R., (Eds.). Springer Verlag, July 2002, pp. 201–213.

[BPK05] BISCHOFF S., PAVIC D., KOBBELT L.: Automatic restoration of polygon models. *ACM Trans. Graph. 24*, 4 (2005), 1332–1352.

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH* (2001), pp. 67–76.

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *SIGGRAPH* (1996), pp. 303–312.

[DG03] DEY T. K., GOSWAMI S.: Tight cocone: a watertight surface reconstructor. In *Symposium on Solid Modeling and Applications* (2003), pp. 127–134.

[DMGL02] DAVIS J., MARSCHNER S. R., GARR M., LEVOY M.: Filling holes in complex surfaces using volumetric diffusion. In *3DPVT* (2002), pp. 428–438.

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH* (1999), pp. 317–324.

[EBV05] ESTEVE J., BRUNET P., VINACUA A.: Approximation of a variable density cloud of points by shrinking a discrete membrane. *Comput. Graph. Forum 24*, 2 (2005), 791–807.

[ESV97] EL-SANA J., VARSHNEY A.: Controlled simplification of genus for polygonal models. In *IEEE Visualization* (1997), pp. 403–412.

[FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. *ACM Trans. Graph. 24*, 3 (2005), 544–552.

[GW01] GUSKOV I., WOOD Z. J.: Topological noise removal. In *Graphics Interface* (2001), pp. 19–26.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J. A., STUETZLE W.: Surface reconstruction from unorganized points. In *SIGGRAPH* (1992), pp. 71–78.

[HK06] HORNUNG A., KOBBELT L.: Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *CVPR* (2006).

[Ju04] JU T.: Robust repair of polygonal models. *ACM Trans. Graph. 23*, 3 (2004), 888–895.

[Kaz05] KAZHDAN M. M.: Reconstruction of solid models from oriented point sets. In *Symposium on Geometry Processing* (2005), pp. 73–82.

[KB05] KOLMOGOROV V., BOYKOV Y.: What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *ICCV* (2005), pp. 564–571.

[LC87] LORENSEN W., CLINE H.: Marching cubes: A high resolution 3d surface reconstruction algorithm. In *SIGGRAPH* (1987), pp. 163–169.

[LSGX05] LOMBAERT H., SUN Y., GRADY L., XU C.: A multilevel banded graph cuts method for fast image segmentation. In *ICCV* (2005), pp. 259–265.

[MAVdF05] MEDEROS B., AMENTA N., VELHO L., DE FIGUEIREDO L. H.: Surface reconstruction for noisy point clouds. In *Symposium on Geometry Processing* (2005), pp. 53–62.

[NT03] NOORUDDIN F. S., TURK G.: Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph. 9*, 2 (2003), 191–205.

[OBA*03] OHTAKE Y., BELYAEV A. G., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. *ACM Trans. Graph. 22*, 3 (2003), 463–470.

[OBS04] OHTAKE Y., BELYAEV A. G., SEIDEL H.-P.: 3d scattered data approximation with adaptive compactly supported radial basis functions. In *SMI* (2004), pp. 31–39.

[PMG04] PAULY M., MITRA N., GUIBAS L.: Uncertainty and variability in point cloud surface data. In *Eurographics Symposium on Point-Based Graphics* (2004).

[SD97] SEITZ S. M., DYER C. R.: Photorealistic scene reconstruction by voxel coloring. In *CVPR* (1997), pp. 1067–1073.

[SFS05] SCHEIDEGGER C. E., FLEISHMAN S., SILVA C. T.: Triangulating point set surfaces with bounded error. In *Symposium on Geometry Processing* (2005), pp. 63–72.

[SLS*06] SHARF A., LEWINER T., SHAMIR A., KOBBELT L., COHEN-OR D.: Competing fronts for coarse-to-fine surface reconstruction. In *Eurographics* (2006), p. to appear.

[VTC05] VOGIATZIS G., TORR P., CIPOLLA R.: Multi-view stereo via volumetric graph-cuts. In *CVPR* (2005), pp. 391–398.

[WHDS04] WOOD Z. J., HOPPE H., DESBRUN M., SCHRÖDER P.: Removing excess topology from isosurfaces. *ACM Trans. Graph. 23*, 2 (2004), 190–208.