Range Tracing

Philipp Jenke and Benjamin Huhle
email: pjenke@gris.uni-tuebingen.de
huhle@gris.uni-tuebingen.de

WSI-2010-01
Februar 2010

Graphisch-Interaktive Systeme
Wilhelm-Schickard-Institut
Universität Tübingen
D-72076 Tübingen, Germany
WWW: http://www.gris.uni-tuebingen.de

ISSN 0946-3852

Abstract

In this report, we tackle the problem of merging an arbitrary number of range scans (depth images) into a single surface mesh. The mesh-based representation is superior to point-based approaches since it contains important connectivity information. Most previous mesh-based merge methods, however, lose surface details by using simplifying intermediate surface representations (e.g. implicit functions). Such details are essential for further processing steps, especially for feature-preserving reconstruction methods. Our method preserves all information (connectivity and the original measurement positions) as edges and vertices of a merged surface mesh. It avoids aliasing and smoothing artifacts, adapts to the local scanner sampling and is independent of the overlap size of the input range scans. The algorithm consists of only two basic operations and is therefore simple to implement. We evaluate the performance of our approach on highly detailed real-world scans acquired with different devices.

1 Introduction

Range scanning devices have become ubiquitous. Each new generation of scanners was smaller, lighter and cheaper than the previous one and usually delivered data with a higher signal-to-noise ratio. One problem, however, remains unsolved: occlusions. Nearly all current range scanners only measure the distance from the eye position of the scanner to the closest surface in different directions. Especially when acquiring complete scenes, scanning from a single viewpoint is therefore insufficient due to occlusions of hidden surfaces. The only way to overcome this problem is to acquire several scans from different viewpoints and merge the resulting data into a unified coordinate system.

We address the problem of merging registered scans into a single consistent representation. Many methods to solve the registration problem are available. The most successful of them build upon the ICP method [BM92]. Unstructured point clouds are then often used in the fusion process because of their conceptual simplicity (see e.g. [GP07]). Not only is it trivial to merge point clouds but also a broad variety of methods exist to further process point clouds: smoothing [ABCO$^+$03], surface reconstruction [OBA$^+$03, KBH06] or even physical simulation [MHTG05], to name only a few.

> *Thus, if you discard mesh connectivity, you are discarding real and possibly useful information about the underlying surface (Marc Levoy, Standford Scanning Repository website)*

However, the main drawback of point-based representations is the lack of topological connectivity. This does not cause problems for sufficiently well-sampled surfaces. Fine sampling, however, is often not available in range scanner datasets. Point-based methods consequently often fail to represent surfaces, especially at fine surface structures or where two parts come close.

The standard data structure to encode topological connectivity is the mesh. For a single depth image, such mesh connectivity is implicitly defined by its regular grid structure [TL94, CL96]. An abundant number of algorithms exist to operate on meshes. For nearly all tasks required to improve the quality of scanned surface meshes, powerful algorithms exist (e.g. surface reconstruction

[FDCO03, JDD03, DTB06], denoising [HP07] or hole filling [DMGL02]). For a set of registered range images, however, it is necessary to first merge the meshes into a single consistent representation.

State-of-the art methods for this task recompute an intermediate surface representation from the different scans – e.g. an implicit function [CL96]. Then, however, the original measurement positions are lost. We argue that consecutive methods to improve the quality need direct access to the original data. Otherwise small features are lost and artificially introduced smoothing corrupts the results. Qualitative conclusions can no longer be made. We present a method here, which does not utilize any prior knowledge about the data and does not use any heuristics on properties of the data. Instead, it only relies on the input range images and preserves the unaltered information.

The central idea of our mesh-based range scan merge algorithm is related to the ray tracing approach. For each measurement, we consider the line-of-sight between the eye position e_i of the corresponding scan and the measurement position m_{ij}. These rays must not intersect any surface in our merged surface mesh. We start with an initial mesh taken from one of the range scans. Then, we iteratively add all other measurements from the other scans and adjust the surface mesh. This is done in two basic operations: *subdivide* to add the detail from each new measurement position and *tunnel* to create corridors for the measurement rays. The result is a single surface mesh (possibly consisting of several components). Its vertices are the original measurement positions and the mesh edges encode the connectivity.

The contribution of our method compared to previous work is that

- we do not require any *intermediate data structure*, such as a voxel grid, in our merging method. Therefore, we avoid artificial aliasing and smoothing artifacts.
- additional, potentially highly detailed *scans* can be *integrated* anywhere as desired later.
- our merged mesh representation is implicitly *adaptive* to the local sampling of the range scans since we use the original measurement positions.
- we introduce *no* data-independent *external knowledge* into the process. Consecutive operations on the data structure have all the information, which is available in the original scans.
- we keep a *consistent representation* at all time – a watertight surface mesh.

We believe that many mesh-based surface processing methods (see for instance [FDCO03, JDD03, DTB06]) can only give reasonable results on either single depth meshes or merged meshes, which consist of this original information only.

The remainder of the report is structured as follows: We first present known methods to merge range scanner data into a unified representation in Section 2. Then, we give the details of our mesh-based merge method (Section 3) with its two central operations *subdivide* and *tunnel*. We describe the implementation of the method and discuss results in Section 4. Section 5 concludes the report.

2 Merging Scanner Data

A survey on merging range images can be found in [RFL02]. We here focus on the part, where a unified consistent surface representation is created.

Point-based: Most merge methods operate on point clouds. The reason for this choice is twofold. First, point clouds are structurally simple and therefore the merge operation is trivial – no topological connectivity has to be updated or can become corrupted. Second, points recently became widely accepted as a full surface representation [GP07]. The point-based representation, however, fails to capture the correct surface structure in regions with low signal-to-noise ratio or where different surface parts come close. Then, only a very dense point-sampling is required, which is usually not available from the scanners. In such cases, mesh-based representations are superior.

Global parameterization: Potmesil [Pot83] extracts depth information with a structured-light technique and then merges patches from multiple views into a global parameterization. Chen and Medioni compute a global parameterization in a cylindrical or spherical domain in [CM92] (similar to [VA86]). The main drawbacks of global parameterizations are that concave surfaces cannot be represented and that the estimated or acquired depth information is often no longer directly available.

Mesh-based: Soucy and Laurendeau [SL95] focus on range scans, which only partially overlap and which are taken from an object observed by a scanner. Their system detects and stitches together the overlapping regions of adjacent range images. No discussion is included how occlusions can be handled. Rutishauser and colleagues [RST94] presented a method to merge depth meshes, which is limited to two fixed sensors. The *zipper* method of Turk and Levoy [TL94] zips meshes together along overlapping parts between different range scans. In the boundary area, they automatically detect and remove overlapping triangles until only a single row of overlapping triangles remains. These triangles are then stitched together by introducing additional vertices and an artificially broadened boundary. The zip operation can cause numerical instabilities since it is basically a 2D method projected to 3D. Our measurement rays as well as the surface mesh representation, in contrast, are true 3D structures. The *zipper* and most other mesh-based methods have a slightly different focus than ours: We consider depth images, which (possibly) completely overlap – scanned scenes instead of scanned objects.

Volumetric: The state-of-the-art method to merge range images is the volume-based system of Curless and Levoy [CL96]. They compute a signed distance function, which is discretized on a voxel grid, from multiple scans. A mesh representation is created in a second step via marching cubes. Sagawa et al. presented an alternative volume-based method in [SNI05] and extended the system in a probabilistic framework in [SNO06]. The grid-discretization, however, introduces smoothing and aliasing artifacts. Too much important information is absorbed by this representation, especially for surfaces with sharp features.

3 Ra(y)nge Tracing

We describe the details of our method in this section. We expect a registered set of watertight meshes $\mathcal{I}_i$ as input to the merge process. Such meshes can easily be extracted from the grid structure in most depth images [TL94, CL96]. We close the boundary of depth images, which do not cover a complete sphere environment, by creating additional triangles between each boundary edge and the eye position.

Our goal is to create a watertight surface triangle mesh $\mathcal{S}$. Its vertices should be the original measurement positions. We initialize the algorithm with any one of the depth images $\mathcal{I}_0$ and iteratively add measurements from other range images $\mathcal{I}_i$. The inclusion of additional measurements requires two basic operations: *subdivide* and *tunnel*. The *subdivide* operation refines the detail and the *tunnel* operation corrects the topology of the surface.

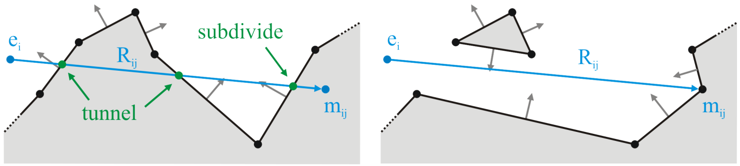
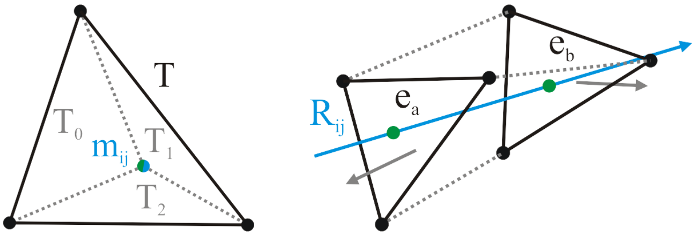


Figure 1: Refinement of the intermediate mesh representation with *subdivide* and *tunnel* operations along a measurement ray $R_{ij}(e_i, m_{ij})$. The *subdivide* operation adds surface detail and the *tunnel* operation corrects the topology.

We cast a ray R_{ij} from the the eye position e_i of the range image $\mathcal{I}_i$ for each measurement position m_{ij}. For now, we assume that the eye e_i is inside the intermediate surface mesh $\mathcal{S}$. The ray then must have an odd number of intersections with triangles inside the mesh. We sort the intersected triangles by their distance to e_i. These cause *tunnel* operations. Additionally, it must have one triangle intersection to exit the volume of $\mathcal{S}$, which causes a *subdivide* operation. This operation inserts the new measurement position m_{ij} into the mesh. The two intersection types are shown in Figure 1. If the eye position is outside $\mathcal{S}$, an additional *subdivide* operation is applied to the triangle, which is closest to e_i. This operation inserts the eye position e_i into the mesh. All vertices, which represent eye positions, can later be removed from the mesh since they do not account for real surface points.

All rays are bounded by the corresponding measurement depth values. We only consider triangle intersections, which are closer than or equal to the distance of the measurement position. If no triangles are intersected, the closest triangle along the elongated ray is accepted for a *subdivide* operation even if its distance is larger than the measurement depth. *Tunnel* operations are only applied to pairs of intersected triangles along the ray, where the first triangle faces towards the eye position e_i and the second backwards. The final triangle used in the *subdivide* operation also has to face towards the eye position. Ray intersections, which do not obey these rules, are postponed. They can for instance occur, if the ray direction is (nearly) perpendicular to the normal of an intersected triangle. Postponed measurements are pushed on a stack and reinserted later. If the inclusion of a measurement still fails in this later step, the measurement is discarded.

3.1 Subdivide



(a) subdivide (b) tunnel

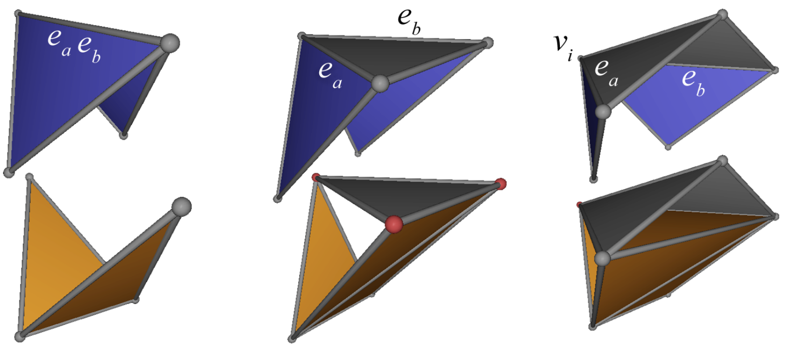
Figure 2: Refinement operations: a) subdivision of a triangle T (solid black lines) into three sub-triangles $T_{1,2,3}$ (dotted gray lines). b) insertion of a tunnel between two triangles, correspondences are established between vertices (gray dotted lines) and edges (e.g. e_a and e_b).

The *subdivide* operation is applied to the triangle, which is last intersected by the measurement ray R_{ij}. This triangle is subdivided into three new triangles. Each new triangle shares one edge with the original triangle and is additionally connected to a new vertex, which is located at the measurement position m_{ij} (Figure 2a).

It is possible that a ray scarcely misses an inner surface part and is assigned to a triangle from a background surface instead. This occurs, if the representation of inner surface structures is still too coarse. It results in a spike created from subdividing the background triangle towards the inner surface structure. We avoid such situations by finding the triangle with the smallest distance to the measurement position. If its distance is smaller than the distance to the background triangle, we instead apply the subdivision operation on this triangle.

3.2 Tunnel

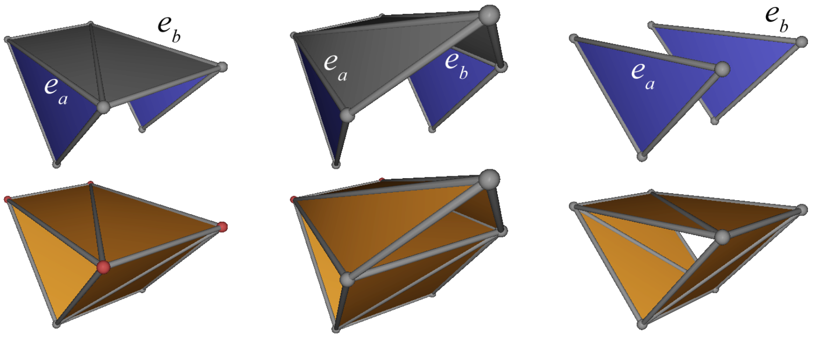
The *tunnel* operation refines the topology of the merged surface mesh. It operates on a pair of triangles. These two are replaced by a tunnel consisting of up to six triangles – an additional hole is introduced into the surface (Figure 1). In order to find the triangulation of the tunnel, we need to establish correspondences between the vertices and in direct consequence the edges of the original triangles (Figure 2b). From all possible triangulations, we select the one with the maximum inner volume. If multiple configurations have the same volume, we chose the configuration with the smaller angles between the normals of triangles, which connect corresponding edges (e.g. e_a and e_b in Figure 2b). We now consider a pair of corresponding edges e_a and e_b. Depending on adjacent triangles in the original configuration (gray triangles in Figure 3), six cases can occur during the triangulation. These cases are shown in the subfigures of Figure 3. Zero, one or two triangles need to be inserted in each case. It is possible that isolated triangle pairs occur (two triangles with same vertices but opposite orientation, cases b and d). Cases c and e require the duplication of vertices to keep the surface mesh $\mathcal{S}$ a manifold surface. Such duplicated vertices are highlighted in red in Figure 3.



(a) e_a equals e_b

(b) e_a and e_b share single vertex, $n = 1$ triangles between

(c) e_a and e_b share a single vertex v_i, $n \neq 1$ triangles between



(d) end vertices of e_a and e_b are connected via edges

(e) e_a and e_b are connected by one edge

(f) e_a and e_b are not connected

Figure 3: Different cases during the *tunnel* operation. The cases are shown for the corresponding upper edges e_a and e_b of the triangles, which were intersected by the ray (blue). These two triangles are removed in any case. Newly created triangles are rendered in orange and adjacent triangles in gray. The top row shows the original configuration and the bottom row the result of the *tunnel* operation. The applied modifications are: a) do nothing. b) create isolated triangle pair with three new vertices. c) duplicate vertex v_i (red), relink intermediate triangles, insert one triangle. d) for each inner triangle: if triangle exists, create isolated triangle pair, otherwise insert triangle. e) duplicate two vertices (red), relink intermediate triangles, create two triangles. f) create two triangles.

Both operations, *subdivide* and *tunnel*, must not be applied if the resulting surface mesh will contain self-intersections. Before we modify the mesh, we therefore check for possible collisions. We push a measurement on a stack if it would cause a collision and try to add it again after all other measurements are included.

The quality of the resulting surface is significantly improved by edge flips: For all edges involved in the two operations, we check if an edge flip leads to smaller edge lengths. We then apply the flip, if it does not cause any self-intersections. This surface improvement does not only lead to visually more pleasing results, but also makes following operations more stable (fewer collisions).

The *tunnel* operations are especially prone to causing collisions. More collisions in turn mean that changes are not applied even though corresponding measurement rays exist. The topological changes from that operation mostly correspond to the low surface frequencies while the details introduced by the *subdivide* operations represent the high frequencies. Consequently, we apply our merge method in a multiresolution manner. We simplify the range images by discarding complete rows and columns leading to a hierarchy of resolution levels. Then, we start with the range images of the lowest resolution and iteratively insert the higher levels. This ensures that large topological changes are robustly applied in the first steps while later measurements fill in the details. Measurements, which were already inserted into the mesh, are discarded in the upper levels.

4 Results

Our method can easily be implemented using well-established tools and methods. We use the half-edge structure of the OpenMesh library[1]. We additionally organize all triangles in an octree. This significantly accelerates all distance queries (self-intersections and closest triangle queries during the *subdivide* operation). Fast methods for all required intersection tests can be found (including code) in [AMHH08].

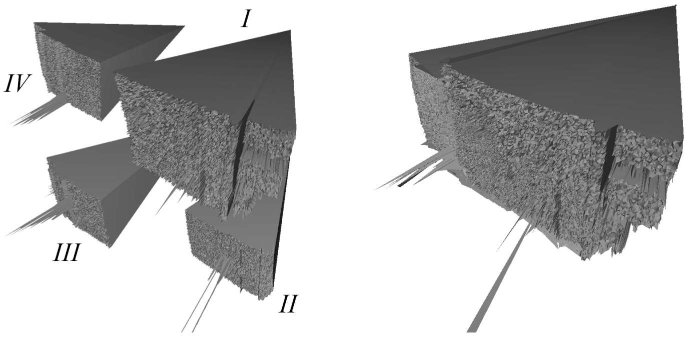
We evaluated our method on different complex range scanner datasets. Figure 4 shows the merge of four range images acquired with a time-of-flight camera. This camera has a very small opening angle and a very low signal-to-noise ratio. Therefore, the resulting datasets are especially challenging for point-based surface reconstruction methods. Most features would immediately be lost. Our mesh-based method preserves the connectivity of the range data and therefore important information for further processing steps.

A reconstruction of a large dataset consisting of three scans acquired with a laser-range scanner is shown in Figure 5. What can be seen in subfigure c is that the large triangles, which incorrectly 'close' the volume of the individual scans due to occlusions, are correctly broken up in the merged surface mesh. Invalid measurements in scans II and III (visible as spikes in subfigure a), which exceed the maximum scanner range, are not used in the merge process.

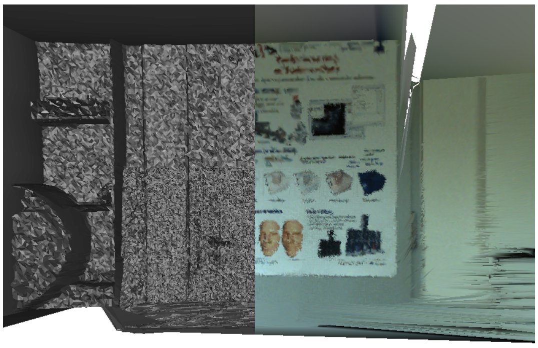
We compared our mesh-based approach with the current state-of-the-art point-based surface reconstruction method in Figure 6. The surface measurements at the tables in the office environment are only given on the upper side due to the scanner positions. This causes such structures to nearly disappear with the Poisson surface reconstruction method [KBH06]. Our result, instead, preserves the surfaces correctly. Note that most point-based surface reconstruction systems require normal directions. Usually, such normals are estimated via PCA. This requires a local influence region as user input (we used the 20-nearest-neighbors in Figure 6a). Our method is independent of such user parameters or user knowledge. We applied one Laplacian smoothing step as implemented in the OpenMesh package to improve the depth perception in this example.

---

[1] http://www.openmesh.org



(a) input range scans (b) merged mesh



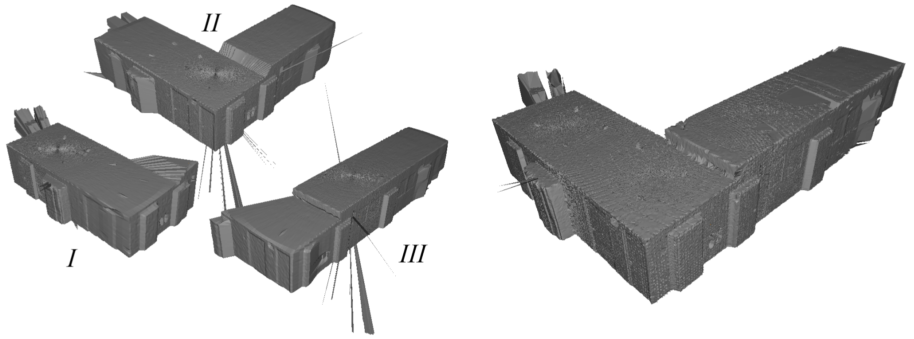
(c) interior view of the merged mesh: colored data (no shading) on the right

Figure 4: Merge of four range scans, which were acquired with a time-of-flight camera. A watertight surface is constructed by connecting the boundary edges of the range images with the scanner eye position (*wall* dataset).

We provide timings for all conducted experiments in Table 1. Our prototype implementation is not optimized for performance even though we use some acceleration techniques as described. Our experiments showed that the performance can sometimes depend on the order in which new measurements are inserted. We currently work on an optimized implementation of the method.

5 Discussion

We presented a method to merge an arbitrary number of range scans (depth images) into a single consistent surface mesh. All vertices of the mesh are original unaltered measurement positions. We only change the connectivity compared to the original scans. The merge process is initialized with one of the range scans, which is assumed to be given as a watertight surface mesh. Afterwards, two basic operations, *subdivide* and *tunnel*, are applied in a multiresolution fashion. The *subdivide* operation introduces detail into the surface, while the *tunnel* operation adjusts the topology. The mesh-based surface representation better preserves the information given in depth images, the standard output of most range scanners, than a point-based representation. Our merging method out-



(a) input range scans (b) merged mesh

(c) interior view of the merged mesh (right half after one Laplacian smoothing step)

Figure 5: Merge operation on three scans acquired with a laser-range scanner, which captures the complete surrounding sphere (*floor* dataset). The artificial walls from scans I (right end) and III (left end) are correctly broken open (c).

performs previous work as it preserves the original depth measurements and as it is parameter-free. An additional advantage of our method is that it is independent of prior knowledge and structurally simple. Since our merge operation is not limited to small overlapping boundary regions, it allows integrating an arbitrary number of range scans of arbitrary complexity.

The approach opens up several additional research directions. We want to investigate how one could better use the connectivity between inserted measurement positions. Also, we would like to try working on range scans with boundaries and therefore discard the requirement for a watertight input.

References

[ABCO$^+$03] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics (TVCG '03)*, 9:3–15, 2003.

[AMHH08] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-Time Rendering 3rd Edition.* A. K. Peters, Ltd., 2008.

(a) point-based (Poisson) surface reconstruction [KBH06]

(b) our merge method (one Laplacian smoothing step applied)

Figure 6: The comparison of the merged data with a state-of-the-art point-based surface reconstruction method shows that we are able to preserve finer details (*office* dataset). The merged data (b) is smoothed with one iteration of a Laplacian smoother.

	#scans (#levels)	*#vertices/#faces*	*time*
floor	3 (3)	213k/421k	40 min.
wall	4 (1)	60k/121k	64 min.
office	3 (3)	203k/377k	48 min.

Table 1: The timings were performed on an Intel Core2 Duo machine (only one core used) with 2.5 GHz and 4 GBs of RAM. *Tunnel* operations on the wall dataset, which operate on triangles adjacent to the eye position, can be very expensive because many triangles might be modified (cases c and e in Figure 3).

[BM92] P. J. Besl and N. D. Mckay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[CL96] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[CM92] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image Vision Comput.*, 10:145–155, 1992.

[DMGL02] J. Davis, S. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *Proceedings First International Symposium on 3D Data Processing, Visualization, Transmission*, 2002.

[DTB06] J. R. Diebel, S. Thrun, and M. Bruenig. A bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics*, 25:39–59, 2006.

[FDCO03] S. Fleishman, I. Drori, and D. Cohen-Or. Bilateral mesh denoising. *ACM Transactions on Graphics*, 22/3:950–953, 2003.

[GP07] M. Gross and H. Pfister, editors. *Point-Based Graphics.* Morgan Kaufmann, 2007.

[HP07] K. Hildebrandt and K. Polthier. Constraint-based fairing of surface meshes. In *Proceedings Eurographics Symposium on Geometry Processing (SGP '07)*, 2007.

[JDD03] T. R. Jones, F. Durand, and M. Desbrun. Non-iterative, feature-preserving mesh smoothing. In *Proceedings ACM Siggraph '03*, 2003.

[KBH06] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings Eurographics Symposium on Geometry Processing (SGP '06)*, 2006.

[MHTG05] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. In *Proceedings of ACM SIGGRAPH '05*, 2005.

[OBA$^+$03] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multilevel partition of unity implicits. In *Proceedings ACM Siggraph '03*, 2003.

[Pot83] M. Potmesil. Generating models of solid objects by matching 3d surface segments. *Proceedings IJCAI*, pages 1089–1093, 1983.

[RFL02] M. Rodrigues, R. Fisher, and Y. Liu. Special issue on registration and fusion of range images. *Comput. Vis. Image Underst.*, 87:1–7, 2002.

[RST94] M. Rutishauser, M. Stricker, , and M. Trobina. Merging range images of arbitrarily shaped objects. In *Proc. IEEE Int. Conf. on Comput. Vision and Pattern Recog.*, 1994.

[SL95] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:344–358, 1995.

[SNI05] R. Sagawa, K. Nishino, and K. Ikeuchi. Adaptively merging large-scale range data with reflectance properties. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:392–405, 2005.

[SNO06] R. Sagawa and Y. Yagi N. Osawa. A probabilistic method for aligning and merging range images with anisotropic error distribution. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 2006.

[TL94] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, 1994.

[VA86] B. C. Vemuri and J. K. Aggarwal. 3d model construction from multiple views using range and intensity data. In *Proceedings CVPR '86*, 1986.