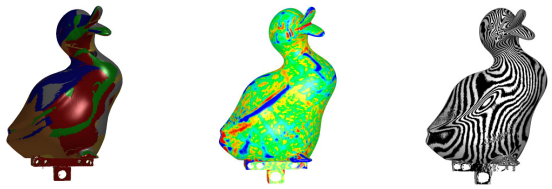
Adaptive Real-Time Grid Generation from 3D Line Scans for fast Visualization and Data Evaluation

Christian Teutsch, Dirk Berndt, Erik Trostmann and Bernhard Preim
Fraunhofer IFF Magdeburg, Germany and O.-v.-G. University Magdeburg, Germany
{christian.teutsch,dirk.berndt,erik.trostmann}@iff.fraunhofer.de, preim@isg.cs.uni-magdeburg.de

Abstract

This paper presents a method for the real-time generation of grids from 3D line scan data for in-line scan previews and the evaluation of large point clouds derived from different 3D-scanners. By exploiting the underlying measuring principles, we generate regular grids for each scan operation even if the sensor movement is non-linear. These grids are finally used for NURBS patch approximations, which enable the determination of higher order features, e.g. curvature and quality evaluations. Experimental results at the example of different point clouds illustrate the effectiveness of our methods in practice.

Keywords— **grid generation, NURBS reconstruction, scan quality evaluation**



(a) (b) (c)

Figure 1: The processing pipeline: Illustration of the generated grids from multiple scan operations (a), each color represents the result of one operation. By using these grids, NURBS patches are approximated for detecting surface features (curvature and direction) (b), and noise is visualized by mapping regular patterns (c).

1 Introduction

Capturing real 3D objects is a common method for the conservation of cultural heritage, reverse engineering or industrial quality inspection. Most 3D scanners use the structured light and triangulation principle to compute 3D positions on the object surface. An often occurring problem when scanning complex objects is that the user or the machine cannot ensure that all surface parts have been captured completely and correctly. Due to shadowing effects, some gaps still remain and outliers or unfavorable surface properties significantly degrade the quality of the result. A post-processing is usually applied to allow for robust measurements and modeling operations on the data.

In contrast to established triangulation methods, we present an alternative algorithm that consequently employs the measuring principles and known system properties. The approach might be adapted to similar 3D acquisition procedures. The basis is a straightforward approach for a real-time grid generation which is, in particular, useful for the in-process control by the user. It allows for a fast result preview and helps the user to check for gaps or interfering reflections. These grids are additionally used to easily approximate NURBS patches which enable us to evaluate surface features with differential geometry methods and check for the existence and quality of important surface features. Important properties are, for example, edges and sampling density as well as the quality of the measurements. As a result, the scanning procedure for large and complex objects is simplified, becomes faster and the results are even better. The typical application for our methods is capturing complex objects with laser triangulation and non-linear scanning paths consisting of several lateral and rotational movements and uneven sampling.

Our main assumption is that nearly no point cloud is unstructured. The underlying measuring principle, the movements, positions and directions of the different sensors yield additional information that can be used for faster algorithms. In this work, we mainly consider the meshing of scanlines, which seems to be a trivial task. However, using the Euclidean coordinates only, does not allow for unique assignments between points and grid. Points on neighboring scanlines must not be topological neighbors, even if some of the spatial coordinates are nearly equal. Therefore, we apply a straightforward approach which employs the known sorting and projection parameters. We transform the 3D "round" scans into a 2.5D projection space. In this space, each 3D position is described by a laser plane and a unique projection angle, which results in a regular row/column grid structure.

The remainder of this paper is organized as follows.

Section 2 describes previous work on integrating additional scan system information and 3D meshing. Section 3 covers the basic principles of our laser scanner and the projection and registration procedures with exemplary results. Section 4 presents the scan quality evaluation and the NURBS based surface measurements. Section 5 summarizes this paper and discusses future work.

2 Previous Work

3D data acquisition and processing are areas of active research. Real world objects have been captured with different measuring principles and many algorithmic tools have been developed to produce high quality triangular representations from 3D points. There is a great deal of published work on mesh generation and some work on the inclusion of measuring principle specific information, particularly in the vision literature. Our literature review only covers work on 3D meshing and principles for including system properties from active and passive range scanners.

2.1 Generation of 3D Meshes

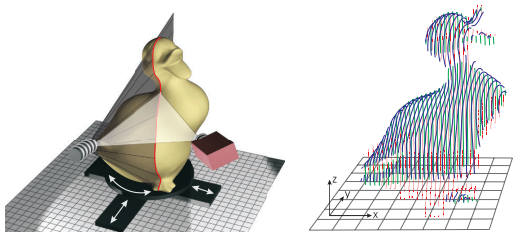
A set of single points can geometrically hardly be interpreted and evaluated without a-priori information about shape or neighborhoods. Therefore, a topological description is usually needed to interpolate the given points. Different fundamental principles exist to generate surface descriptions from a set of 3D points. Typically, triangle meshes are used, which can be derived by using marching cubes algorithms [11]. They are often used for providing initial meshes, which are relaxed in following processes [4]. However, most approaches use the Delaunay tetrahedrization and a filtering for constructing a triangular mesh. These methods are robust and have proven reconstruction properties concerning sampling density and quality guarantees [2]. However they tend to be costly, because of the time-consuming computation of the Delaunay tetrahedrization or its dual Voronoi diagram [7]. The visualization of point clouds without polygonal meshes, but with point set surfaces, is discussed by Alexa et al. in [1]. Levoy et al. applied the entire process pipeline to the statues of Michelangelo [8]. They present procedures for capturing large objects including texture. In addition, Bernardini et al. use a photometric system to subsequently scan Michelangelos' Florentine Pietà [3].

2.2 Exploiting System Information

When considering system specific information, the properties of 2.5D range images are typically discussed. The inclusion of system specifics for an optimized point cloud processing is also discussed in [9] to improve the iterative closest point (ICP) registration of point sets from different scan positions. Although we create grids prealigned by the system calibration, we apply the mesh registration introduced in [5], which is based on thin-plate splines (see Sect. 3.2). In a range image, the measured positions usually are organized, since the reference image induce a natural parametrization of the corresponding surface. This fact is employed for combining normal vector orientations and 3D positions for increased model precision [10]. A method that resamples range images to align scanlines with a voxel grid is discussed in [6].

3 3D Measuring Principle

For the generation of point clouds, we have built a flexible 3D scanning system. It consists of a multi-axis locomotor system and two structured light sensors with digital cameras and line lasers. The object is rotated and translated in front of the sensors. For each position, both sensors project and measure a contour on the surface. Capturing large or complex objects often requires multiple scan operations. For each, the object is non-linearly moved (translation and/or rotation) and the resulting scanlines (and sublines) are stored as one operation.



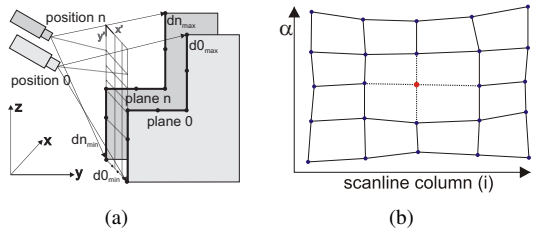
(a) (b)

Figure 2: The flexible laser triangulation system, consisting of three motion devices for lateral and rotational movements (a). One 3D sensor consists of a line laser and a digital camera (we use two sensors, one on bottom and one on top). The result is a structured point cloud, consisting of scanlines and sublines (b) of different operations.

The grid generation requires the 3D data to be projected to 2D by evaluating the laser position for each scanline. A similar problem for computing triangular meshes is discussed in [14]. To build a mesh, they create triangles from four points that are in adjacent rows and columns. We seize this idea for line scan operations on non-linear scanning paths and discontinuous scan lines.

3.1 Grid Projection

Our approach aims at constructing regular grids without pre-calculating triangulated meshes. We use a technique, which is known as mapped meshing. Therefore, a grid and its dimensions are automatically defined. Actually the horizontal dimension (x') of the grid is equal to the number of scanlines. The vertical sampling (y') depends on the object height. Therefore we use the mean distance between two laser planes in relation o the object height. The operation between scanlines is a known system parameter (translation or rotation). Depending on the scanning direction and



(a) (b)

Figure 3: Grid generation: illustration of the min./max. projection angles with respect to one scan operation (a) and derived regular row/column grid structure (b).

the number of sampling steps, we compute the sorting and the distances between each two scanlines per scan. Additionally, the sorting of the points on a scanline is checked, depending on the laser position.

The main procedure is the projection of grid onto the points of a scan operation from laser position. This is a "safe" location for each scanline from which all points have been seen without shadowing effects. We assign a unique direction of projection and the projection angle to each scanline point (Fig. 3) to attain the row/column grid structure. The next step finds a corresponding neighbor for each point with respect to all scanlines of an operation. We assume, that points that have been seen from the same direction of projection, are at least near or neighbors. This is valid, if the surface is slightly changing from line to line. In practice, neighboring scanlines have different point densities and their lengths and shapes may change strongly.

We ensure, that all points from the same direction of projection for all scanlines of one scan are mapped to one horizontal line in the grid. Therefore, we use the projection angle. Considering all scanlines of one operation, there is a minimum and a maximum angle. These angles limit the projection space. All grid positions $g(i,\alpha_j)$ are calculated, based on the defined dimensions by linear interpolation between those two angles. For each point $p(i,k)$, the projection angle α_k is computed, which is unique for its scanline i Eq.(1). Finally, p is back projected to the grid, and the 3D coordinate is assigned to the corresponding grid position. In contrast to terrain modeling, we use angular relations, because there is no common background plane and the back projection of rotational scans could not be handled and would destroy the scanline structure.

$$g(i,\alpha_j) = p(i,\alpha_k) \quad with, \qquad (1)$$
$$|\alpha_j - \alpha_k| < |\alpha_j - \alpha_{k-1}| \wedge |\alpha_j - \alpha_k| < |\alpha_j - \alpha_{k+1}|$$

Gaps cause knots without valid 3D positions, and degenerate cases may arise at positions where scanlines (of one rotational operation) overlap. These are detected by computing the intersection line of the laser planes. To keep the grid consistent, we assign them a dummy point with a zero weight. This case can also be avoided by choosing the rotation axis outside the measuring range.

The discretization of the measured points may result in single gaps within the computed grid. Thus, the initial grids possibly have to be repaired and smoothed. Gaps are located as unused dummy knots in the grid structure (Fig.3(b)). Small gaps are "closed" by bilinear interpolation. Because of the regular grid structure fast algorithms are easily applied, e. g. a Laplacian smoothing is performed Eq.(2). For each grid point, all n topological neighbors are considered. The influence of these points on its new position is controlled with the weight parameter λ. As a result, the grid structure is more regular, and noise is reduced. Typically, four iterations ($\lambda = 0.5$) are sufficient to achieve a fair grid.

$$x_0^{new} = x_0^{old} + \lambda \sum (\frac{x_i^{old} - x_0^{old}}{n}), \quad 0 < \lambda < 1 \qquad (2)$$

3.2 Grid Registration

Because of uncertainties from the calibration procedures, there are small deviations between overlaying grids from different sensors, thus, they do not match exactly. Therefore, we used the ICP algorithm [12] for matching the grids. We start the first iterations with a pre-orientation based on a point-to-point query. This is reached by building a kd-tree from the points of the static grid. The nearest neighbor for the dynamic point cloud is efficiently found by searching the tree. Once the correspondence is established, the transformation is computed with the least-squares method. The distances between overlaying grids are small and thus, only a few (< 10) iterations are needed.

After the fast generation of the grids, we are able to produce a high quality preview of the actual scan operations (Fig. 4) in real-time. This supports the user to detect gaps or noise and illustrates the correct alignment of several surface patches derived from different scan operations and sensors. The computation of surface features on those grids yields rather coarse results, because the grid spacing is not equidistant. To compensate for this, we use the preview grids for a NURBS approximation in the next step and compute surface curvatures more robustly. Due to the sampling theorem the grid dimensions should not exceed half of the number of scanlines and points per line.

4 Surface Measurements

Feature measurements on the interpolated grids require an analytical surface description. Our meshing approach and triangulation methods (see section 2) generate piecewise linear surfaces. The optimal solution is a parametric surface description, which gives the possibility to measure at arbitrary positions. Thus, we use rational B-spline surfaces with a non-uniform knot distribution (NURBS). We

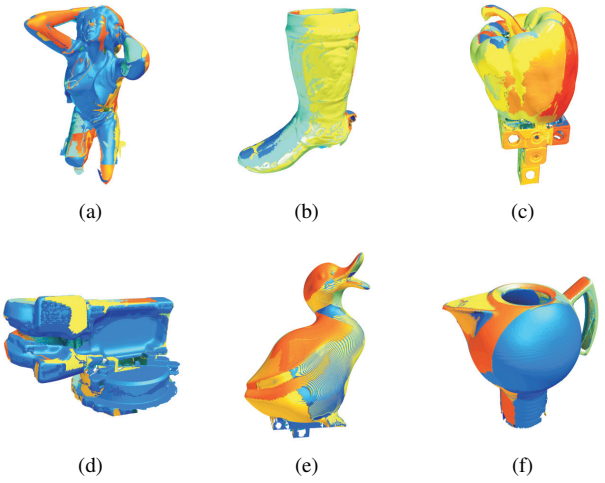


Figure 4: Illustration of resulting grids in different colors at the example of point clouds from the models of woman(a), boot(a), pepper(c), casting(d), duck(e) and a can(f). All transformations are first computed for the grids and then applied to the point clouds.

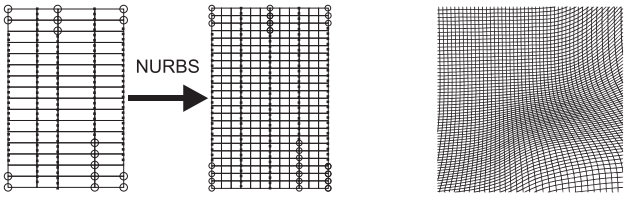


Figure 5: Approximating grids (marked unused knots) (a) by a NURBS patch (b) (even spacing with non-uniform knot vector), example of a resulting NURBS grid (c).

simply use the grids as control net. Because of some unused areas and existing gaps in the underlying grid, we define weights. Grid positions with unused knots get a zero weight. The influence of these points on their environment becomes zero and the consistency of the control net is kept.

4.1 NURBS-Approximation

A non-uniform rational B-spline surface of degree (p,q) is defined by the basis functions N, the weights w and the points of the control net P:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}}. \qquad (3)$$

The basis functions are recursively defined by:

$$\begin{aligned} N_{i,p}(u) &= \frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1} + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u), \\ N_{i,0}(u) &= 1 \text{ if } u_i \le u \le u_{i+1} \text{ or } 0 \text{ otherwise.} \end{aligned} \qquad (4)$$

Typically, 3rd degree surfaces are used in CAD applications and our analysis with differential geometry methods also requires the calculation of at least the 1st and the 2nd derivative of the surface. Additionally, we compensate the non-equidistant structure control grid by using non-uniform knot vectors. The principle is shown in Figure 5.

4.2 Computing the NURBS Surface Weights

The initial preview grid serves as NURBS control mesh, which consists of initial weights which are either 1 or 0, depending on used/unused grid positions. But an approximation with a rational B-spline patch allows for a more sensitive weighting. Therefore, we take a closer look at the quality of each measured 3D point by evaluating the scanline. This section discusses the use of system specific information to compute the data quality and the approximation with B-spline curves to determine significant locations, like edges. Both results are combined to get a weight for each surface point as proposed in [13]. The weight is then assigned to the control grid.

Because the scan process uses optical sensors, the quality directly depends on the viewing and projection properties. The smaller the angle between surface normal and direction of projection or viewing, α_p and α_c respectively, the better the surface was seen. In addition, the triangulation between projection vector $\vec{p}$ and the camera viewing vector $\vec{c}$ is optimal when the angle $(\alpha_p + \alpha_c)$ defined by them is $\frac{\pi}{2}$. The more $(\alpha_p + \alpha_c)$ deviates from $\frac{\pi}{2}$ and the larger the angles themselves, the worse the viewing conditions for camera and/or laser. The normal vectors for a cell are bilinearly interpolated by the four knots forming that cell.

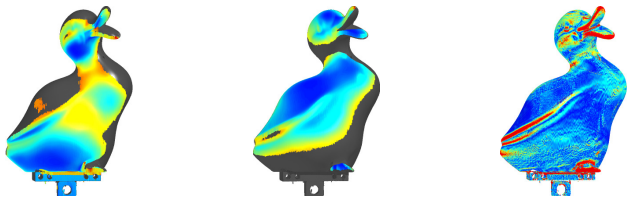
To minimize the smoothing at sharp edges, we use the curvature of each point on a subline as a weight in the grid. High curvatures are detected by analyzing approximated B-spline curves. A B-spline $x(t)$ of order k is defined over an ordered knot vector T as vectorial polynomial with basis functions $N_{i,k,T}(t)$ and control points d_i:

$$x(t) = \sum_{i=0}^{n} d_i N_{i,k,T}(t), \qquad t \in [t_{k-1}, t_{n+1}] \quad . \qquad (5)$$

To ensure that a B-spline curve approximates given points in an optimal way, control points may be generated from the measured points. Therefore, the squared Euclidean distance between the curve points X_i and the measured points M_i is minimized. Control points D_i are computed depending on the basis functions N_i and the M_i:

$$\mathbf{D} = ((\mathbf{N}^T \cdot \mathbf{N})^{-1} \cdot \mathbf{N}^T) \cdot \mathbf{M} \quad . \qquad (6)$$

For performance reasons, we directly use the measured sublines as control polygon, which is possible since the definition of B-spline curves defines the points to stay



(a) (b) (c)

Figure 6: Computing NURBS weights for a scanned object. Shaded point clouds showing: the quality depending on projection/viewing angle for the lower (a) and the upper (b) sensor (0=blue to $\frac{\pi}{2}$=red) and curvature with highlighted significant edges ($\kappa > 0.2$) derived from the B-spline curve approximation per single scanline (c) (blue:$\kappa = 0$ to red:$\kappa > 0.2$).

within the (noisy) control polygon. Because we are approximating the sublines of a scanline, no significant smoothing is brought into the data. We applied 4^{th} order B-spline curves and found that curvatures of $\kappa > 0.2$ indicate sharp edges reliably. The curvature of parameterized 3D curves is given by:

$$\kappa = \frac{\|\vec{r}'(t) \times \vec{r}''(t)\|}{\|\vec{r}'(t)\|^3}. \qquad (7)$$

To determine the weights w for the NURBS patch from the curve approximation of each scanline, we propose the following rules to weight the control points depending on the obtained curvature properties and the normalized quality values from the scanline quantification:

$$w = \begin{cases} 1 & \text{not rateable point} \\ 1 + w_c + w_l + 0.5w_t + 1 & \text{for } \kappa > 0.2 \\ 1 + w_c + w_l + 0.5w_t + \kappa & \text{otherwise} \end{cases}. \qquad (8)$$

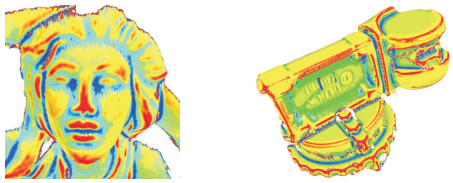
w_c, w_l - camera viewing angle/laser projection angle.
w_t - triangulation angle (lower due to w_l/w_c).

The angle normalization is achieved by scaling with their maximum, which is $\frac{\pi}{2}$. The total weight must no be normalized, since B-splines just evaluate the ratios.

Finally, the NURBS patches are computed, and the following section discusses the visualization of surface features. Using the original data as control grid is possible, since B-splines always stay within the control grid and the smoothing error is smaller than the noise.

4.3 Calculation of Surface Features

Surface features, especially edges, are indicators for the completeness, accuracy and correctness of the measured surface points and allow for a visual inspection of the result. The analysis of parameterized surfaces is usually



(a) (b)

Figure 7: Gaussian curvature of a face (a) and a casting (b) of NURBS patches(blue: K=-100 to red: K=+100).

based on Gauss' fundamental theorem of surface theory. A surface is defined by the relation r=r(u,v) with the real parameters u,v and the relations x=x(u,v), y=y(u,v) and z=z(u,v). For the determination of metric properties of a surface, there are three types of fundamental forms. The most important are the first and second. The first Gaussian fundamental theorem for curved surface is explicitly given by the Riemannian metric Eq.(9). The second fundamental form is the symmetric bilinear form with respect to the tangent space of the first, and is given by relation Eq.(10).

$$ds^2 = Edu^2 + Fdudv + Gdv^2 \qquad (9)$$
$$-dNdr = Ldu^2 + 2Mdudv + Ndv^2 \qquad (10)$$

The Gaussian curvature K and the mean curvature H are one of the most important surface features. A positive curvature indicates bumps and negative values pits. The curvatures are defined based on the parameters of the first (E,F,G) and second (L,M,N) fundamental forms:

$$K = (LN - M^2)/(EG - F^2) \qquad (11)$$
$$H = (LG - 2FM + EN)/(2(EG - F^2)). \qquad (12)$$

Two exemplary illustrations for the curvature based feature detection are given in Fig. 7. The positions of the contours of the models have been evaluated. We additionally display surface discontinuities with the projection (mapping) of stripes (Fig. 1(c)).

5 Summary and Future Work

We presented a straightforward approach for the fast representation of scanned 3D points by regular row/column grids. The methods are fast and adaptive, because they exploit the particular scan device properties and the measuring principle. We described an automated method to project the generated 3D points to a regular grid. The resulting grids are used for a fast preview and quality display to control the scanning process. Additionally, the grids are registered to minimize and balance the uncertainties and noise caused by different sensor calibrations. Since, our approach is based on the structure of neighboring scanlines, the grid construction procedure can also be applied

model	woman	boot	pepper
total points:	2.057.610	1.246.994	2.135.745
grids:	8	8	16
time/grid:	322 ms	210 ms	363 ms
model	**casting**	**duck**	**can**
total points:	1.656.360	1.628.515	1.681.967
grids:	8	8	10
time/grid:	306 ms	418 ms	271 ms

Table 1: Performance evaluation for the construction of the initial grids with respect to the point clouds illustrated in Fig. 4. The NURBS patch computation takes additional time between 3.2 and 4.7 seconds per model.

to other scan systems, which are based on structured light. The performance and feasibility of the presented methods are evaluated in Table 1 with respect to Figure 4(standard PC Pentium4 with 512MB RAM). Furthermore, we applied piecewise NURBS patches using the initial grids as control mesh. This allows for more robust computations of surface features (e.g. curvature) to evaluate the captured surfaces. The methods is adaptive, because it uses weights depending on the point quality. Our method is particularly useful for generating previews when scanning large objects or using non-linear scanning paths.

This work continues by applying fast methods to "close" small gaps in the initial grids by nonlinear surface interpolation. Additional algorithms, that connect the grids of different operations while preserving the grid structure, are also required. Furthermore, the grids serve as an excellent base for fast divide-and-conquer triangulations of the entire point cloud. The limitation of the proposed procedures is that scanlines of one operation should not overlap, because it can cause self-overlays in the grid. This case may not be nice but does not disturb for our previews and cannot occur for lateral scans, fringe projection and photogrammetric methods.

References

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Trans. on Visualization and Computer Graphics*, 9(1):3–15, January 2003.

[2] N. Amenta, S. Choi, and R. K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Comput. Geom., Theory and Applications*, 19:201–210, 2001.

[3] F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, and G. Taubin. Building a Digital Model of Michelangelo's Florentine Pietà. *IEEE Comp. Graphics & App.*, 22(1):59–67, January 2002.

[4] M. Bertram, G. Reis, R. H. van Lengen, S. Koehn, and H. Hagen. Non-manifold mesh extraction from time-varying segmented volumes used for modeling a human heart. In *Eurovis 05*, pages 1–10, 2005.

[5] B. J. Brown and S. Rusinkiewicz. Non-rigid range-scan alignment using thin-plate splines. In *Proc. 3D Data Processing, Visualization, and Transmission*, pages 759–765, Washington, USA, 2004. IEEE CS.

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proc. 23rd conf. on computer graphics and interactive techniques*, pages 303–312, New York, USA, 1996. ACM Press.

[7] T. K. Dey, J. Giesen, and J. Hudson. Delaunay based shape reconstruction from large data. In *Proc. IEEE Symp. Parallel and Large Data Visualization and Graphics (PVG 2001)*, pages 19–27, 2001.

[8] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D Scanning of Large Statues. In *Proc. SIGGRAPH 2000*, pages 131–144. ACM, 2000.

[9] A. S. Mian, M. Bennamoun, and R. A. Owens. From unordered range images to 3d models: A fully automatic multiview correspondence algorithm. In *Theory and Practice of Computer Graphics 2004 (TPCG'04)*, pages 162–166, 2004.

[10] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. In *SIGGRAPH '05*, pages 536–543, New York, USA, 2005. ACM Press.

[11] G. M. Nielson. On marching cubes. *IEEE Trans. on Vis. and Comp. Graphics*, 9(3):283–297, 2003.

[12] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[13] C. Teutsch, T. Isenberg, E. Trostmann, M. Weber, T. Strothotte, and D. Berndt. Evaluation and correction of laser-scanned point clouds. In *Proc. Videometrics VIII (Electronic Imaging 05)*, volume 5665, pages 172–183. SPIE/IS&T, 2005.

[14] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proc. SIGGRAPH '94*, pages 311–318. ACM, 1994.