

On-line CAD Reconstruction with Accumulated Means of Local Geometric Properties

Klaus Denker¹, Bernd Hamann², and Georg Umlauf¹

¹ Institute for Optical Systems, University of Applied Science Constance, Germany,
kdenker|umlauf@htwg-konstanz.de

² Institute for Data Analysis and Visualization (IDAV), Department of
Computer Science, University of California, Davis, CA 95616, U.S.A.,
hamann@cs.ucdavis.edu

Abstract. Reconstruction of hand-held laser scanner data is used in industry primarily for reverse engineering. Traditionally, scanning and reconstruction are separate steps. The operator of the laser scanner has no feedback from the reconstruction results. On-line reconstruction of the CAD geometry allows for such an immediate feedback.

We propose a method for on-line segmentation and reconstruction of CAD geometry from a stream of point data based on means that are updated on-line. These means are combined to define complex local geometric properties, e.g., to radii and center points of spherical regions. Using means of local scores, planar, cylindrical, and spherical segments are detected and extended robustly with region growing. For the on-line computation of the means we use so-called accumulated means. They allow for on-line insertion and removal of values and merging of means. Our results show that this approach can be performed on-line and is robust to noise. We demonstrate that our method reconstructs spherical, cylindrical, and planar segments on real scan data containing typical errors caused by hand-held laser scanners.

1 Introduction

Hand-held laser scanners in industry are primarily used for reverse engineering. A human operator moves the laser scanner along the surface of a physical object to sample its geometry. Usually the point set generated by the scanning process is shown to the operator. The reconstruction of geometric objects is computed afterward in a separate step, where a manual extraction of the geometric parameters is done. When data is missing in the scan the complete process fails. Thus, the whole scanning and reconstruction process must be repeated.

When using an on-line reconstruction algorithm, the reconstruction is done simultaneously with the acquisition of the scan data. Missing data can immediately be detected and corrected by the operator: one simply scans the critical region again. Thus, all data structures of the reconstruction are updated with the new data to improve the quality of the reconstruction immediately.

In this context *on-line computation* means that the data is already processed while the operator is scanning the object. Therefore, the operator can interact

with the results of this reconstruction process. In contrast to real-time computing, no strict time constraints have to be guaranteed.

We present an on-line algorithm for segmentation and reconstruction of CAD geometry from a stream of point data. It is based on accumulated means of local geometric quantities and classification scores for different surface types. Accumulated means allow for on-line insertion, removal, and merging of data.

Data structures for local geometry, edge detection, and segmentation are described in Section 3. Accumulated means of local geometric properties and classification scores are presented in Section 4. A segmentation strategy using the advantages of accumulated means is described in Section 5. The performance of the method with simulated and real scan data is analyzed in Section 7.

2 Related work

One of the first approaches of reconstructing data from triangulation laser scanners is presented in [1], where cylinders are fitted to point data from a laser scanner. To reconstruct complex objects a segmentation of the surface is necessary. An extensive overview of surface segmentation methods is provided in [24]. The underlying data structures are either meshes, e.g. [29], or a k -nearest neighbor graph, e.g. [30], for an efficient local surface analysis. For CAD applications usually a segmentation by surface type is computed. The Gaussian sphere is used in [3] to segment surfaces by their dimensionality, whereas variational surfaces are used in [29].

Early methods for the reconstruction of geometry are discussed in [27]. They are based on the reconstruction of polygonal boundary models and the fitting of simple surfaces and free-form geometry. The method presented in [2] focuses on the reconstruction of rotational and translational surfaces and blends. In [29] variational surfaces are used to reconstruct implicit representations of geometric primitives. These methods do not work for on-line computations of point streams.

Iterative methods for geometry reconstruction often apply stochastic algorithms like Random Sample Consensus (RANSAC) fitting. They are used for the reconstruction of geometric primitives [23] or super-quadrics [4]. These RANSAC based methods are applied to random sub-sets of the data, whereas our segmentation approach is based on all available data. Iteration between segmentation and reconstruction is used in [26]. Here, based on quadric fitting the segmentation of an unorganized point cloud is iteratively improved. Complete data sets are required for each iteration.

Stream processing of point data is discussed in [18, 6]. A sweep line approach sequentially processes large point sets with a set of geometric operators. Thus, the point sets have to be pre-sorted along one spatial direction.

Hand-held laser scanners generate unorganized streams of point data. Such streams can be triangulated on-line [5]. The data is reduced to a set of vertices that are almost uniformly distributed. A surface mesh is generated on-line connecting these vertices. This approach was extended by [10, 11] with a multi-level data structure. Thus, the mesh can be adapted to non-uniform point densities.

An incremental computation of means is introduced in [28, 22]. These methods are compared with other non-incremental methods to compute means and variances with respect to numerical performance in [16, 8]. The application of these methods for variance computations is also discussed in [15]. These incremental means are constructed to add single values to the mean. The removal of values or merging of multiple means is not considered.

Clustering methods like k-means [19] or *unweighted pair group method with arithmetic mean* (UPGMA) [25] use mean positions of multi-dimensional data. On-line k-means methods like the ones covered in [31] allow the incremental addition of data to the mean centroids of the clusters. New values are added with a fixed or decreasing learning rate. Data changing the cluster is not removed from the old cluster centroids. We note that on-line k-means is not an on-line method in the sense we define in the Introduction. Efficient implementations of UPGMA, as in [12] based on arithmetic distance means, use a reduction formula to merge means. Though, no higher order geometric properties are used.

A method that combines local geometric properties to more stable estimations is presented in [17]. Discrete estimates for normals and curvatures are combined over an area using a voting algorithm. However, this is not an on-line method and no geometric parameters derived from the local geometry are used.

The method presented in this paper is partially based on [9]. It differs in the used methods for segmentation and reconstruction. In [9] the segmentation is based only on local geometric properties and the reconstruction of geometric primitives is based on quadric fitting.

In this paper we use a segmentation based on accumulated means. The segmentation also provides the information for the reconstruction of geometric primitives. Thus, there is no separate reconstruction step or quadrics fitting.

3 Data structures

The on-line reconstruction is based on a ball tree data structure as proposed in [11]. It is used to store and process the data stream from the laser scanner and to prepare it for the determination of surface segments.

3.1 Ball tree

The data stream generated by a laser scanner consists of noisy raw point data in three-dimensional (3d) space and orientation data of the laser probe. The raw points are assigned to *n(neighborhood)-balls* β that are defined as

$$\beta(\mathbf{c}, r) = \{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x} - \mathbf{c}\| < r\},$$

where \mathbf{c} denotes the center and r the radius of the n-ball. The n-balls might overlap. Every new raw point \mathbf{q} is added to the n-ball $\beta(\mathbf{c}, r)$ that contains \mathbf{q} with minimal distance $\|\mathbf{q} - \mathbf{c}\|$. If no such n-ball exists, a new n-ball $\beta(\mathbf{q}, r)$ is constructed. The radius r is the maximum r such that $\beta(\mathbf{q}, r)$ does not contain any other n-ball centers. For detail we refer to [11].

For fast access, the n-balls are organized in a *ball tree*, which is an octree data structure. An n-ball is added to the octree region that contains its center. The edge length of this region is the radius of its n-balls. Thus, n-ball radii are dyadic fractions of the initial scan area's edge length e_O , i.e., $r = e_O/2^n$, $n \in \mathbb{N}$.

Each n-ball β stores estimates for the local normal and principal curvatures. The *principal component analysis* (PCA) [14] is used to compute the normal \mathbf{n} of the raw points of a neighborhood of n-balls in the ball tree. The Weingarten map [7] of a quadratic approximation P_β to these raw points based on the local tangent plane defined by \mathbf{n} is used to compute estimates on the principal curvatures κ_1 , κ_2 and principal directions \mathbf{d}_1 , \mathbf{d}_2 . These estimates are only used, when the ratio of the smaller eigenvalues of the PCA is less than $1/2$.

For each n-ball β the arithmetic mean of its raw points is computed and projected onto P_β . The resulting point \mathbf{p} represents the local geometry of β and is used for the reconstruction. Additionally, each n-ball holds a list of closest neighbor points, similar to [30].

3.2 Sharp feature detection

The quadratic polynomial P_β approximates the geometry of the raw points of β accurately only when the geometry is sufficiently smooth. Sharp features cannot be approximated accurately by a single quadratic polynomial. Thus, we use two polynomial approximations to reconstruct sharp features. More than two approximations are possible but their reconstruction is not sufficiently stable.

To compute P_β we use a *least absolute deviation approximation* instead of a least squares approximation. It is based on the L^1 norm instead of the L^2 norm and is better suited for partial reconstructions. For the computation *iteratively re-weighted least squares* (IRWLS) [13] are used. For IRWLS compute the weighted least squares approximation with weights $w_i = 1/\sqrt{d_i}$, where d_i is the distance of the approximation to the raw points. These weights are adjusted in every iteration. The weight adjustment compensates for the difference between the L^1 and L^2 norm. Usually, after approximately five iterations P_β is sufficiently close to a least absolute deviation approximation.

During the IRWLS computation the number of outliers with distance $d_i > \varepsilon_{sd}r$ is recorded. If there are more than 20% outliers, we assume a sharp feature in the scanned geometry and approximate a second polynomial P_β^2 . This second polynomial is initially IRWLS approximated to the outliers. Both polynomials $P_\beta^1 = P_\beta$ and P_β^2 are IRWLS approximated to all raw points with weights

$$w_i^1 = \max(d_i^2 - \varepsilon_{sd}, 0) / \sqrt{d_i^1}, \quad \text{and} \quad w_i^2 = \max(d_i^1 - \varepsilon_{sd}, 0) / \sqrt{d_i^2}.$$

Thus, raw points close to one of the approximations obtain a small weight for the other one. The superscript k refers to quantities related to P_β^k , $k = 1, 2$. The ratio of outliers not close to any surface to the number of approximated raw points defines a quality score q_{ls} for the approximation.

For sharp features, the point \mathbf{p} is projected onto the intersection curve of P_β^1 and P_β^2 by iterative projection to the intersection line of local tangent planes.

3.3 Segment data structure

A *segment* is a set of n-balls of approximately the same geometric type. The implementation is based on a hash table [20]. Thus, look-up and insertion operations have amortized computational complexity $O(1)$. For the on-line CAD reconstruction segments are classified by their geometry. Currently the supported *segment types* are planar, cylindrical, spherical, and unknown geometry.

Each segment maintains a set of its neighbor segments. This set contains all segments containing n-balls that are neighbors to n-balls in the segment. Neighbor segments are added to this set when n-balls are added or modified. At the same time a back link is added to the neighbor segment's set. Neighbor entries are only removed upon segment deletion.

For the association of n-balls to segments also a hash table is used. Furthermore, a hash table for each n-balls is used to maintain all geometric quantities that are associated to this n-ball in order to guarantee consistently accumulated means, as described in the next section.

4 Accumulated means

The advantage of using means of geometric properties instead of purely local geometric properties for the detection of surface segments is the increased stability in the computations. Thus, we use means throughout our on-line approach.

Weighted and unweighted arithmetic means of a set of n data x_i with constant weights w_i are defined as

$$\bar{x}_n = \frac{\sum_{i=1}^n x_i w_i}{\sum_{i=1}^n w_i} \quad \text{and} \quad \tilde{x}_n = \frac{1}{n} \sum_{i=1}^n x_i.$$

For on-line computations means are computed by incremental addition of data. Besides, it is necessary to remove data from a mean or to merge two means. We call a quantity *accumulated* when it supports these three *accumulation operations*, e.g., \bar{x}_n is an accumulated weighted arithmetic mean.

4.1 Accumulated arithmetic means

The incremental computation of an unweighted arithmetic mean \tilde{x}_n when adding a value $x_n, n > 1$, is defined as

$$\tilde{x}_n = \tilde{x}_{n-1} + \frac{1}{n} (x_n - \tilde{x}_{n-1}), \quad \tilde{x}_1 = x_1.$$

For numerical reasons we avoid adding $(n-1)\tilde{x}_{n-1}$, see [16]. To compute a weighted arithmetic mean \bar{x}_n , the average weight \tilde{w}_n is used

$$\bar{x}_n = \bar{x}_{n-1} + \frac{w_n}{(n-1)\tilde{w}_{n-1} + w_n} (x_n - \bar{x}_{n-1}), \quad \bar{x}_1 = x_1.$$

A value $x_{n+1}, n \geq 1$, can be removed from an unweighted arithmetic mean by

$$\tilde{x}_n = \tilde{x}_{n+1} - \frac{1}{n} (x_{n+1} - \tilde{x}_{n+1})$$

and from a weighted arithmetic mean by

$$\bar{x}_n = \bar{x}_{n+1} - \frac{w_{n+1}}{(n+1)\tilde{w}_{n+1} - w_{n+1}} (x_{n+1} - \bar{x}_{n+1}).$$

Two arithmetic means \tilde{a}_n and \tilde{b}_m with $m \leq n$ are merged to one combined arithmetic mean \tilde{x}_{n+m} by

$$\tilde{x}_{n+m} = \tilde{a}_n + \frac{m}{n+m} (\tilde{b}_m - \tilde{a}_n). \quad (1)$$

The combined weighted arithmetic mean is computed using the arithmetic means of the weights \tilde{u}_n, \tilde{v}_m of the partial weighted arithmetic means \bar{a}_n and \bar{b}_m

$$\bar{x}_{n+m} = \bar{a}_n + \frac{m\tilde{v}_m}{n\tilde{u}_n + m\tilde{v}_m} (\bar{b}_m - \bar{a}_n).$$

The weight mean \tilde{w}_{n+m} for the combined mean \bar{x}_{n+m} is computed from \tilde{u}_n, \tilde{v}_m by Equation (1). We note that the computational complexity of all accumulation operations is $O(1)$, which is optimal for an on-line algorithm.

4.2 Accumulated arithmetic means in the ball tree

For each n-ball in a ball tree we obtain sets of values that are combined in accumulated arithmetic means. Unweighted arithmetic means are used for position \mathbf{p} , normal \mathbf{n} , and ball radius r . Their means $\tilde{\mathbf{p}}, \tilde{\mathbf{n}}, \tilde{r}$ are used to detect planar segments. Accumulated weighted arithmetic means $\bar{\kappa}_1, \bar{\kappa}_2, \bar{H}$ are computed for principal curvatures κ_1, κ_2 , and mean curvature H . The respective weights are given by the quality score q_s of the least squares approximation.

4.3 Indirect accumulated means

For each segment accumulated weighted arithmetic means for both principal curvatures and the mean curvature yield *indirect accumulated weighted means* for the hypothetical cylinder and sphere radii \bar{r}_1, \bar{r}_2 , and \bar{r}_s

$$\bar{r}_i = -\bar{\kappa}_i^{-1}, i = 1, 2, \quad \text{and} \quad \bar{r}_s = -\bar{H}^{-1}.$$

Other geometric quantities are defined by normalized, non-oriented, vector-valued data, e.g., principal curvature directions $\mathbf{d}_1, \mathbf{d}_2$. Thus, arithmetic means cannot be used to combine principal directions. Due to normalization, this data lies on the unit-sphere. Their distribution on the unit-sphere is estimated by a PCA step. For both principal curvature directions $\mathbf{d}_i, i = 1, 2$, the 3×3 covariance matrix \mathbf{C}_i is computed. The eigenvalues $\lambda_{i,j}, j = 1, 2, 3$, of \mathbf{C}_i characterize these distributions. The eigenvectors of the largest eigenvalues $\lambda_{i,3}$ are used

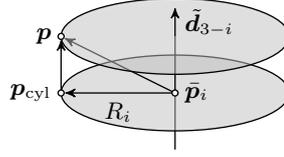


Fig. 1: Points and directions of a cylindrical segment.

as unweighted means $\tilde{\mathbf{d}}_i$ for the non-oriented directions. Co-variance matrices \mathbf{C}_i are accumulated, because data \mathbf{d} can be added or removed by

$$\mathbf{C}_i^{\text{new}} = \mathbf{C}_i^{\text{old}} \pm \mathbf{d}_i \cdot \mathbf{d}_i^t,$$

and two co-variance matrices $\mathbf{C}_i^1, \mathbf{C}_i^2$ are merged by

$$\mathbf{C}_i = \mathbf{C}_i^1 + \mathbf{C}_i^2,$$

yielding the co-variance of the union of both sets of non-oriented directions. Thus, $\tilde{\mathbf{d}}_i, i = 1, 2$, are *indirect accumulated means*. They provide information for cylinder detection: the cylinder axis has the same direction as the principal direction of the smaller absolute principal curvature.

The complexity for the accumulation operations for a 3×3 co-variance matrix \mathbf{C} is $O(1)$. Due to the constant size of \mathbf{C} , the computation of its eigenvalues and eigenvectors has also complexity $O(1)$. Thus, the overall complexity is $O(1)$.

4.4 Accumulated means depending on means

For each segment hypothetical cylinder and sphere center points are estimated by locally approximated center points

$$\mathbf{p}_i = \mathbf{p} - \bar{r}_i \cdot \mathbf{n}, i = 1, 2, \quad \text{and} \quad \mathbf{p}_s = \mathbf{p} - \bar{r}_s \cdot \mathbf{n}.$$

These local estimates accumulate to the weighted arithmetic means $\bar{\mathbf{p}}_1, \bar{\mathbf{p}}_2$, and $\bar{\mathbf{p}}_s$ with weights $|\bar{\kappa}_1|, |\bar{\kappa}_2|$, and $|\bar{H}|$, respectively. We use weighted arithmetic means to account for the non-linear distribution of center points and the fact that there is no multi-dimensional harmonic mean. Since the local estimates and the weights vary with time, they are stored for the accumulation operations.

Using $\bar{\mathbf{p}}_1, \bar{\mathbf{p}}_2$, and $\bar{\mathbf{p}}_s$ a more stable local estimate for the radii can be computed, see Figure 1,

$$\begin{aligned} R_i &= \|(\mathbf{p} - \bar{\mathbf{p}}_i) - ((\mathbf{p} - \bar{\mathbf{p}}_i)\tilde{\mathbf{d}}_{3-i})\tilde{\mathbf{d}}_{3-i}\|, & i = 1, 2, \\ R_s &= \|\mathbf{p} - \bar{\mathbf{p}}_s\|. \end{aligned}$$

The accumulated weighted means of corrected radii \bar{R}_1, \bar{R}_2 , and \bar{R}_s are computed again with weights $|\bar{\kappa}_1|, |\bar{\kappa}_2|$, and $|\bar{H}|$.

4.5 Scores for geometric primitives

For segmentation we define scores to measure the fitting quality of an n-ball to planar, spherical, cylindrical, or unknown geometry. We use multiplicative scores that are centered at 1 using a normalization weight ω , see Table 1. Values smaller than 1 denote high fitting quality. Partial scores s_i are combined by weighted multiplication using weights w_i to yield a combined score

$$s = \prod_{i=1}^n ((s_i - 1) w_i + 1) \text{ with } w_i \in (0, 1).$$

For each segment all scores are maintained at all times. Currently we compute scores for planar, cylindrical, spherical and unknown geometries. Each of these scores consists of a partial score for the segment size, a distance score, an angle score, and a curvature score. The latter is used only for cylindrical and spherical geometries. In the computations all means belong to the same segment and all local data belongs to the corresponding n-ball. The weights in the multiplicative combination are 3/4 for the distance and angle scores and 1/2 otherwise.

The score for planar segments s_{pln} is computed based on these quantities:

- Distance score:** Normal distance $\tilde{n}(\tilde{\mathbf{p}} - \mathbf{p})/(\tilde{r}\omega_{\text{nd}})$.
Angle score: $\angle(\mathbf{n}, \tilde{\mathbf{n}})$.
Size score: Reciprocal of number of n-balls in the segment.

The score for cylindrical segments s_{cyl} is based of the direction of the cylinder curvature $\tilde{\kappa}_\delta$ that is orthogonal to the cylinder axis $\tilde{\mathbf{d}}_\Delta$ with

$$\delta = \underset{i=1,2}{\operatorname{argmin}} |\lambda_{i,3}| \quad \text{and} \quad \Delta = \underset{i=1,2}{\operatorname{argmax}} |\lambda_{i,3}|.$$

Thus, the index δ is an indirect accumulated quantity characterizing the type of the cylinder. To simplify radius computations we compute a point on the cylinder surface in the plane normal to $\tilde{\mathbf{d}}_\Delta$ containing $\tilde{\mathbf{p}}_\delta$, see Figure 1,

$$\mathbf{p}_{\text{cyl}} = \mathbf{p} - \left((\mathbf{p} - \tilde{\mathbf{p}}_\delta) \tilde{\mathbf{d}}_\Delta \right) \tilde{\mathbf{d}}_\Delta.$$

The partial scores for cylindrical segments are:

- Distance score:** Radius difference $\left| \|\mathbf{p}_{\text{cyl}} - \tilde{\mathbf{p}}_\delta\| - |\tilde{R}_\delta| \right| / (\omega_{\text{cyl}} |\tilde{R}_\delta|)$.
Angle score: Average of $\angle(\tilde{\mathbf{d}}_\Delta, \tilde{\mathbf{d}}_\Delta)$ and $\angle(\mathbf{n}, \mathbf{p}_{\text{cyl}} - \tilde{\mathbf{p}}_\delta)$.
Curvature score: $\max(|\kappa_\delta|, |\tilde{\kappa}_\delta|) / \min(|\kappa_\delta|, |\tilde{\kappa}_\delta|)$.
Size score: Reciprocal of number of n-balls in the segment.

The score for spherical segments s_{sph} is computed from:

- Distance score:** Radius difference $\left| \|\mathbf{p} - \tilde{\mathbf{p}}_s\| - |\tilde{R}_s| \right| / (\omega_{\text{sph}} |\tilde{R}_s|)$.
Angle score: $\angle(\mathbf{n}, \mathbf{p} - \tilde{\mathbf{p}}_s)$.
Curvature score: $\max(|H|, |\tilde{H}|) / \min(|H|, |\tilde{H}|)$.
Size score: Reciprocal number of n-balls in the segment.

For unknown geometry a score s_{unk} is computed based mostly on local values of two neighbored n-balls β and β_{nb} :

Distance score: Normal distance $\mathbf{n}_{\text{nb}}(\mathbf{p}_{\text{nb}} - \mathbf{p})/(\bar{r}\omega_{\text{nd}})$.
Angle score: $\angle(\mathbf{n}, \mathbf{n}_{\text{nb}})$.
Size score: Reciprocal of number of n-balls in the segment.

Since the scores are combined from different geometric quantities, the ranges for the overall scores s_{pln} , s_{cyl} , s_{sph} , and s_{unk} differ significantly, e.g., good values of s_{pln} tend to be much smaller than good values of s_{cyl} . To compensate for this effect and to avoid a time-consuming re-balancing of the partial scores, the overall scores are re-balanced by correction factors. These factors are 2.5 for s_{pln} , 0.7 for s_{cyl} , 0.9 for s_{sph} , and 6.0 for s_{unk} .

For s_{pln} , s_{cyl} , and s_{sph} accumulated means \bar{s}_{pln} , \bar{s}_{cyl} , and \bar{s}_{sph} are computed for each segment. These means are weighted by the quality q_{ls} and are used to determine the overall segment type. The smallest mean score determines the segment type. If it is larger than 1 the segment is classified as unknown geometry.

5 Segmentation strategy

Our segmentation algorithm is implemented as a separate layer on top of the ball tree described in Section 3. The n-balls can be added, removed or modified at any time. For each of these updates we need to update the segmentation.

5.1 Changing single n-balls

When a new n-ball is generated, it is assigned to the best-fitting segment. To find the best-fitting segment the score s_{pln} , s_{cyl} , s_{sph} , or s_{unk} of this n-ball for each hypothetical segment that contains an n-ball in the local neighborhood is computed. The segment type of the hypothetical segment determines which score for the n-ball is computed. Then it is added to the segment with the minimal score by adding its data and scores to the means of the segment. The n-ball is also inserted into the segment data structures.

To remove an n-ball its old values are removed from the means of its segment. The n-ball is removed from the segment data structure. If a segment contains no n-balls after a removal, the segment is deleted.

For modifications of n-balls the same minimal scores as for adding n-balls are computed. When a score indicates a change of segments, the n-ball is removed from the old segment and added to the new one. Otherwise, the new values of the n-ball replace the old values in the means of its segment.

Accumulating the means of a segment is of complexity $O(1)$. Assuming that the size of the local neighborhood of an n-ball is constant, the search for the best-fitting segment has also complexity $O(1)$. Updating the set of n-balls of a segment and updating the global hash table of n-balls have complexity $O(1)$.

5.2 Merging segments

After several updates for one segment, a hypothetical merged segment with each of its neighbor segments is computed. This hypothetical segment contains the

data of both segments using the accumulation formula for the means and covariance matrices. We note that in the sequel quantities with superscript $k = 1, 2$ refer to data associated with one of the merged segments and quantities without superscript to data of the resulting hypothetical segment.

The segment type of the hypothetical segment is determined by the accumulated mean scores \bar{s}_{pln} , \bar{s}_{cyl} , \bar{s}_{sph} . Depending on the resulting segment type further merge conditions based on thresholds ε are evaluated, see Table 1. For planar segments the normal distance between the center points $\tilde{\mathbf{p}}^1$, $\tilde{\mathbf{p}}^2$ of both original segments and the angle between their normals $\tilde{\mathbf{n}}^1$, $\tilde{\mathbf{n}}^2$ are evaluated, i.e.

$$|\tilde{\mathbf{n}}^1(\tilde{\mathbf{p}}^1 - \tilde{\mathbf{p}}^2)| + |\tilde{\mathbf{n}}^2(\tilde{\mathbf{p}}^2 - \tilde{\mathbf{p}}^1)| < \varepsilon_{\text{nds}}(\tilde{r}^1 + \tilde{r}^2) \quad \text{and} \quad \angle(\tilde{\mathbf{n}}^1 \tilde{\mathbf{n}}^2) < \varepsilon_{\angle}.$$

For cylindrical segments the angle between the cylinder axes, the deviation of the cylinder axes are evaluated, and the difference of radii, i.e.

$$\begin{aligned} \angle(\tilde{\mathbf{d}}_{\Delta}^1 \tilde{\mathbf{d}}_{\Delta}^2) < \varepsilon_{\text{ca}}, \quad & \|(\tilde{\mathbf{p}}_{\delta}^2 - \tilde{\mathbf{p}}_{\delta}^1) - ((\tilde{\mathbf{p}}_{\delta}^2 - \tilde{\mathbf{p}}_{\delta}^1) \tilde{\mathbf{d}}_{\Delta}^1) \tilde{\mathbf{d}}_{\Delta}^2\| < \varepsilon_{\text{cc}} |\bar{R}_{\delta}|, \quad \text{and} \\ |\bar{R}_{\delta}^1 - \bar{R}_{\delta}^2| < \varepsilon_{\text{cr}} |\bar{R}_{\delta}|. \end{aligned}$$

For spherical segments the difference of radii and the distance of their center points are evaluated, i.e.

$$|\bar{R}_s^1 - \bar{R}_s^2| < \varepsilon_{\text{sr}} |\bar{R}_s| \quad \text{and} \quad \|\tilde{\mathbf{p}}_s^1 - \tilde{\mathbf{p}}_s^2\| < \varepsilon_{\text{sc}} |\bar{R}_s|.$$

Segments classified as unknown geometry cannot be merged.

When the hypothetical segment satisfies the combination of the merge conditions, the merge operation is realized: The means and data of the smaller segment are merged to the means and the data of the larger segment, the neighbor information is updated, and then the smaller segment is deleted. After a merge operation, further checks for merges of the involved segments are dropped.

The complexity to compute a hypothetical segment is $O(1)$. Hypothetical segments are computed for each of the k neighbors, so the complexity of checking for merges is $O(k)$. The complexity of a merge operation is $O(1)$ for merging the means. Merging the sets of n-balls of the segments has an amortized complexity of $O(m)$, where m is the number of n-balls in the smaller segment. The global hash table associating n-balls to segments is updated with complexity $O(m)$. Thus, the overall complexity to merge two segments is $O(k + m)$.

5.3 Consistency checks

Every time a segment changes its type, all its n-balls are checked for consistency. This is also done when a cylindrical segment changes the direction of its axis. For each n-ball in the segment the segment with the minimal score s_{pln} , s_{cyl} , s_{sph} , or s_{unk} is determined. If it differs from the current segment, the n-ball is removed from the current segment and added to the new segment.

The computational complexity of this consistency check is $O(m)$ where m is the number of n-balls in the segment. Each of these n-balls determines its best-fitting neighboring segment with $O(1)$ time complexity. It is removed from the current segment with $O(1)$ and added to another one with $O(1)$, if necessary.

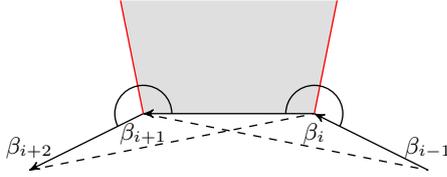


Fig. 2: Sector (gray) between half-angle lines (red) of a boundary edge $\langle \beta_i, \beta_{i+1} \rangle$.

6 Segment boundaries

In CAD applications one must handle trimmed surfaces. A boundary curve is constructed that represents the outer contour of the surface. To render the results of the on-line CAD reconstruction a similar approach is used. Polygonal boundaries are constructed surrounding the areas of the primitives. A polygonal boundary B is an ordered sequence of n-balls β_i with $1 \leq i \leq n_B$. By definition, it is ordered counterclockwise when seen from the outside.

The segmentation in Section 5 uses a region growing approach. A segment starts with a single n-ball and is extended by adding n-balls. When it consists of three n-balls, a first initial boundary B with $n_B = 3$ can be constructed. A local planar projection is used to choose a counterclockwise orientation of B .

6.1 Boundary expansion

All the following operations use a local planar projection based on the segment data, e.g., in the direction towards the center point of a spherical segment. The orientation of a segment is defined by the sign of its average normal $\tilde{\mathbf{n}}$ or radius \tilde{r}_1 , \tilde{r}_2 , or \tilde{r}_s .

For each n-ball β_{new} that is added, modified or removed from a segment, B is updated. When a newly added n-ball is outside of B , it is inserted into the closest edge $\langle \beta_i, \beta_{i+1} \rangle$ of B . The neighbors β_i of β_{new} on the boundary B are determined. Both edges $\langle \beta_{i-1}, \beta_i \rangle$ and $\langle \beta_i, \beta_{i+1} \rangle$ are candidates for inserting β_{new} . In the projection, β_{new} has to lie within the sector bounded by the half-angle lines to its two neighboring edges, see Figure 2. The edge that fulfills this condition and whose center point is closest to β_{new} is chosen for insertion. After the insertion, consistency checks improve the shape of the boundary.

Changes of the values of an n-ball on the boundary trigger only the consistency checks. If it is not on the boundary, it is treated like a newly added n-ball. When removing an n-ball from the segment, it is also removed from the boundary. If it was on the boundary, we attempt to re-connect the boundary by inserting neighboring n-balls to the boundary. In the projection space, the neighboring n-ball β_{nb} with the smallest sum of inner angles to the half-angle lines next to the deleted n-ball is searched, see Figure 2. If such a β_{nb} is found and it lies outside of the chosen boundary edge, it is inserted into the boundary.

Symbol	Threshold	Value	Symbol	Threshold	Value
ε_{sd}	Distance to surface	0.14	ω_{nd}	Normal distance	0.8
ω_{cyl}	Cylinder radius difference	0.1	ω_{sph}	Sphere radius difference	0.1
ε_{nds}	Normal distance between $\tilde{\mathbf{p}}$	0.4	$\varepsilon_{\triangleleft}$	Angle between $\tilde{\mathbf{n}}$	20°
ε_{ca}	Angle between $\tilde{\mathbf{d}}_{\Delta}$	20°	ε_{cr}	Difference between \bar{R}_{δ}	0.2
ε_{cc}	Deviation of cylinder axes	0.4	ε_{sr}	Difference between \bar{R}_s	0.2
ε_{sc}	Distance of $\tilde{\mathbf{p}}_s$	0.4			

Table 1: Used scale invariant thresholds.

6.2 Multiple boundaries

A segment can have multiple boundaries B_i , e.g., a complete cylinder mantle has two boundaries around the caps. When scanning such a cylindrical segment, a single boundary B_1 is created that is split into two boundaries B_1, B_2 , when a full scan around the cylinder was completed. Whenever two outsides of the boundary are close to each other, such a split operation is performed.

Merging of segments, see Section 5.2, requires a merging of their boundaries. After the segments are merged, all boundaries of the segment are checked for neighboring n-balls on different boundaries of the same segment. If a part of both boundaries is close together and runs in opposite direction, the boundaries are merged. The boundary parts that are close to each other are removed and their end points are connected to create a single boundary.

6.3 Local consistency checks

After connections or n-balls of B are modified, a local consistency check is performed for a single n-ball β_i . If the outer angle between both its edges $\langle \beta_{i-1}, \beta_i \rangle$ and $\langle \beta_i, \beta_{i+1} \rangle$ is smaller than 150°, β_i is removed from the boundary. For all neighbors β_{nb} of β_i on the same boundary, intersections are computed between all connected boundary edges. If two intersections are found, the boundary is split by connecting the two segments before and after the intersections.

If in the local neighborhood of β_i a point on another boundary on the same segment is found, a merge operation of these boundaries is initiated.

Finally, for segments of two edges on B that are close to the edges next to β_i with have opposite directions a split operation is done, see Section 6.2.

7 Results

For our method we demonstrate the effectiveness, robustness to noise, and performance for real scan data. It runs at interactive speed, where the segmentation takes only little processing power. All results are computed using the thresholds in Table 1, which are scale invariant because they are angles or relative distances.

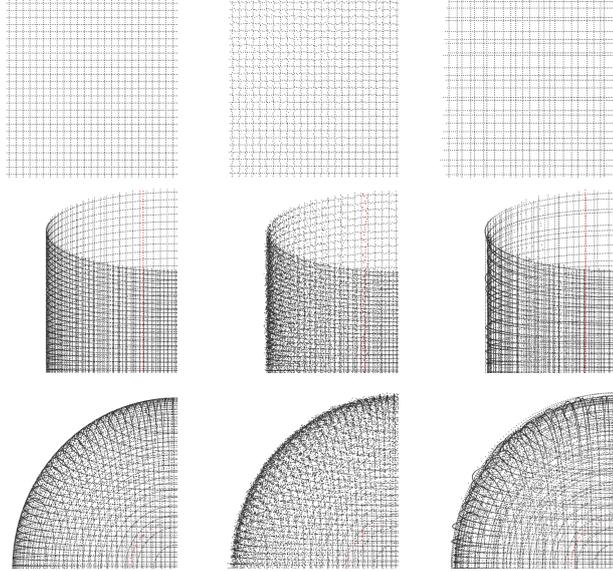


Fig. 3: Sections of simulated scans without noise (left), with laser noise (middle), and with tracking noise (right).

7.1 Synthetic scan data

To demonstrate the robustness to noise, we use synthetic scan data, where the true segment types are known. These data sets are shown in Figure 3. The simulated plane has an edge length of 20 cm. The cylinder height is 20 cm and the radius is 10 cm; the sphere radius is 10 cm. The added noise is generated from normal-distributed random numbers. It is scaled according to different levels of standard deviation σ . We use two different types of noise:

Laser noise: Noise of the laser scanner is directed along the scanning direction. It is applied to each raw point separately.

Tracking noise: Noise of scanner tracking is added to the scanner position and the raw points in all three spatial directions. It is applied to each scan line.

The plots in Figures 4a to 4c show the results of our experiments with synthetic planar, cylindrical, and spherical data, respectively. For each level of noise with standard deviation $\sigma = i/10[mm]$, $i = 0, \dots, 10$, we have computed 25 experimental segmentations and reconstructions. For each experiment new noise is generated. The plots show the average distance of the parameters of plane, cylinder, and sphere to the known ground truth parameters. For planes the average normal distance is plotted. For cylinders and spheres the average radius difference and the average distances of the center lines or points are plotted.

The evaluation shows that tracking noise has the largest effect on the reconstruction. The plane reconstruction even improves when some laser noise is

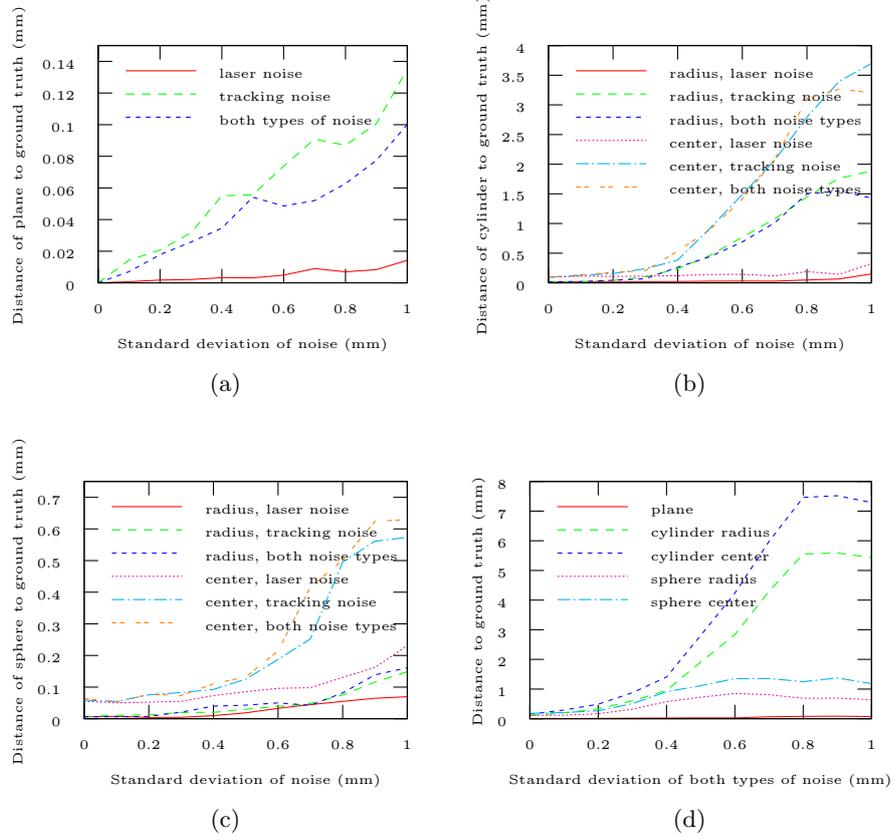


Fig. 4: Plots of the average normal distance of 25 experiments for noise levels $\sigma = i/10[\text{mm}]$, $i = 0, \dots, 10$, for plane, cylinder, and sphere reconstruction. Plot (d) shows the result for all segment types using a 6DoF input controller.

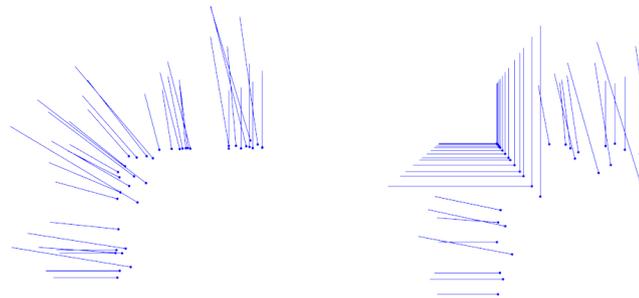


Fig. 5: N-ball points \mathbf{p} and normals \mathbf{n} near a sharp edge computed without and with sharp feature detection.

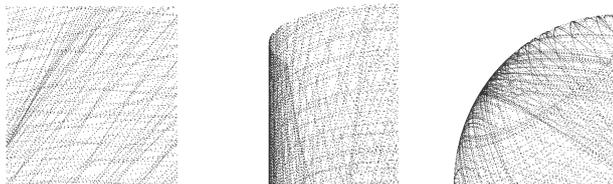


Fig. 6: Sections of simulated scans scanned interactively with tracking of hand movements.

added to the tracking noise. Cylindrical and spherical reconstructions lead to better results for the radius than for the center line or point. Planar reconstruction is least affected by noise followed by spherical reconstruction. The added noise has the most influence on the cylindrical reconstruction.

Planar reconstruction is based on positions \mathbf{p} and normals \mathbf{n} that are quite robust to noise. Spheres also use the mean curvatures H for the radius. The cylinder radius depends on principal curvatures κ_1, κ_2 and principal directions $\mathbf{d}_1, \mathbf{d}_2$. Especially the principal directions are effected substantially by local noise. The reconstruction errors of cylinder radii are around 10 times larger than the errors of sphere radii for the same levels of tracking noise. Both segment types use radii $\bar{r}_1, \bar{r}_2, \bar{r}_s$ and normals \mathbf{n} for center reconstruction. Here, the error for cylinders is six times larger than for spheres in the presence of tracking noise.

While the precision of the reconstruction is affected by the noise none of the reconstructions has failed. No outliers with extremely high error rates appear in the data of these plots. Noise with standard deviation $\sigma > 1$ mm would cause the segments to split into smaller segments, reconstructing parts of the data. Most of these would be planar as a curvature based reconstruction would fail.

The sharp feature detection is shown in Figure 5. The n-balls near the edge have two different normals and the points \mathbf{p} are projected onto the feature line.

7.2 Hand-tracked synthetic scan data

The synthetic scan data simulates linear movements of the scanner that cause a rasterized structure of the scan lines. To avoid this effect it is necessary to track hand movements. We use a magnetic *six degrees of freedom* (6DoF) input controller called *Razer Hydra* [21] to track interactive scans of the simulated objects, see Figure 6. The sphere and the cylinder were scanned only from one side to demonstrate the reconstruction of incomplete objects.

For all synthetic data-sets with noise level $\sigma < 0.5$ mm the segmentation produces perfect results: a single segment containing all n-balls is reconstructed.

For the evaluation of these scans we plot in Figure 4d the same parameters as in Figures 4a to 4c for both types of noise. The non-uniform distribution of the scan lines does not influence the noise robustness for planar reconstruction much. The partial scans of the cylinder and sphere are much more affected by noise. Their reconstructions have errors roughly twice as large as those for the

complete data-set. Since only one half of the objects is scanned, the corrected radii R_i , R_s do not have the same stabilizing effect as for complete scans.

7.3 Real scans

We use point streams from a real laser scanner to evaluate the on-line reconstruction method. For the presented experiments, we scanned objects with a *Faro Edge Laser ScanArm*.

We scanned the toy fire truck shown in Figure 7. The length of the truck is 0.6 meters. Its material is coated wood with stripes of black rubber as tires around the wheels. The surface is composed of basic geometric primitives. There are many flat areas that are reconstructed with planar segments. Cylindrical regions can be found at the rounded edges, the wheels, and the holes at the ladder. The front lights, the center of the wheels, and the head of the driver figure are spherical. First we scanned the driver's cabin of the toy truck. Most segments are detected correctly, see Figure 8. Because of the dark material, the tires are not detected as single cylindrical segments. They split up into multiple smaller flat and cylindrical segments. At the rims, flat and spherical segments are correctly reconstructed. Cylinders with small radii are reconstructed at the rounded edges of the driver's cabin. Even a small spherical segment can be seen at the front left corner of the hood. At the radiator grill, a flat area is reconstructed that contains two holes for the lights. The reconstruction shows some remaining problems with the boundary reconstruction that result in incomplete and jagged shapes. The floor below the toy truck in Figure 8d shows that missing regions are covered by n-ball vertices. They are correctly segmented but not enclosed by the reconstructed boundary.

For our second example, we scanned the rear end and the ladder of the toy fire truck, see Figure 9. The bottom of the ladder platform consists of large cylinders. They demonstrate the capabilities of the accumulated means reconstruction to correctly classify segments with large radii. Four cylindrical holes at the upper part of the ladder are scanned from the inside. They are correctly reconstructed and rendered as cylinders with surface normals pointing to the inside. The signs of principal curvatures are preserved during the accumulated means reconstruction. The black rubber tire visible in this scan was captured better than the one in the last scan and is reconstructed by a single cylinder mantle. Another cylinder mantle with slightly smaller radius reconstructs the visible part of the wooden wheel. Figures 9c and 9d show the tire in purple and the outside of the wheel in blue. The narrow green cylinder in Figures 9b to 9d shows a typical error of the boundary rendering. Temporarily, the outer parts of the cylinder contained boundary points. Therefore, it is assumed that these parts lie within the boundary. Instead of just a small area that really is inside the boundary, almost the complete cylinder mantle is rendered.



Fig. 7: Photograph of a wooden fire truck toy.

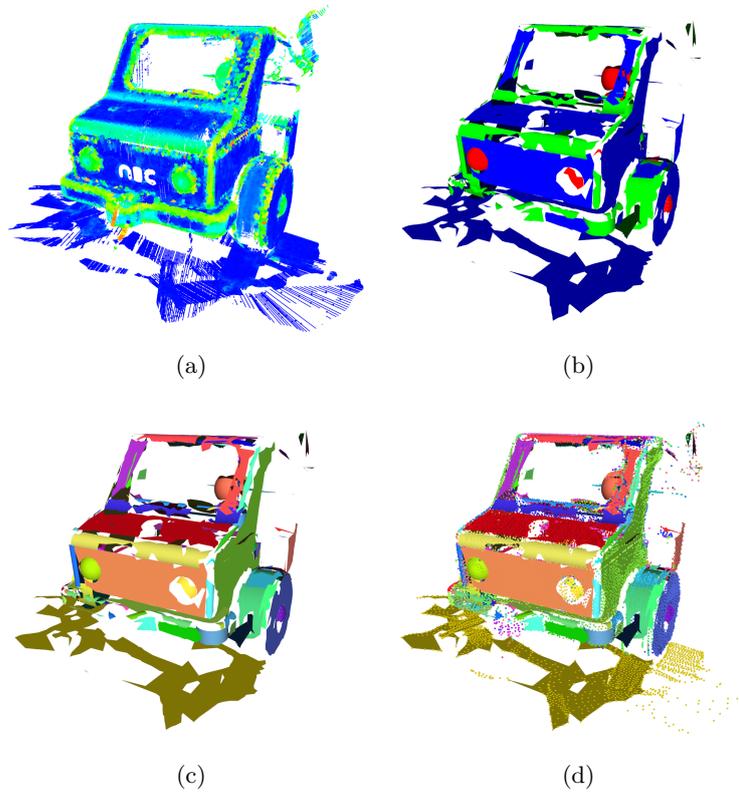


Fig. 8: Raw data and reconstruction from a scan of the driver's cabin of the wooden fire truck toy: 1 271 342 points, 12 325 scan lines, 15 013 n-balls, 1 011 segments. Raw data colored by curvature magnitude (a), segments colored by segment type: **planar**, **cylindrical**, and **spherical** (b), or each segment individually (c), additionally the n-ball vertices are shown in the same colors in (d).

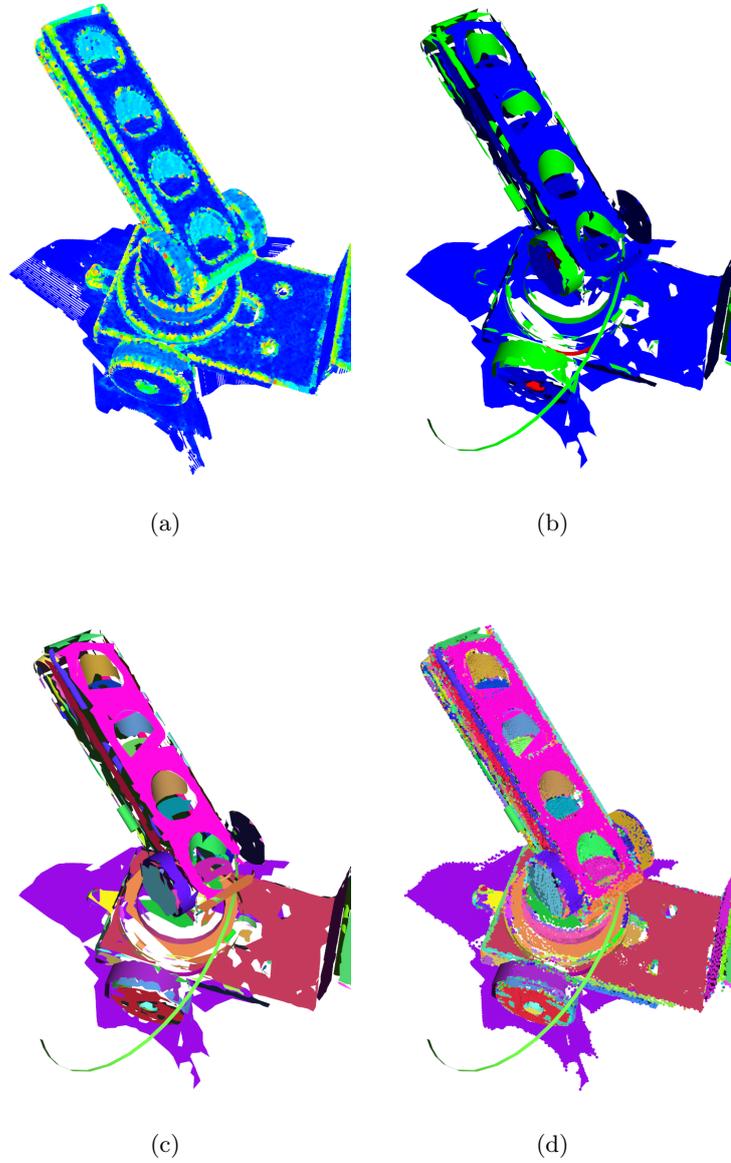


Fig. 9: Raw data and reconstruction from a scan of the ladder of the wooden fire truck toy: 3 128 055 points, 26 749 scan lines, 32 002 n-balls, 1 733 segments. Raw data colored by curvature magnitude (a), segments colored by segment type: **planar**, **cylindrical**, and **spherical** (b), or each segment individually (c), additionally the n-ball vertices are shown in the same colors in (d).

7.4 Performance

A large number of segments was reconstructed for these scans. Though, the implementation always performs at interactive speed, processing 30 scan lines with up to 5 640 points per second. We used an Intel Core 2 Quad Q6600 2.4 GHz computer with 8 GB of RAM. The most processing power is used to update the ball tree data structure. The thread responsible for the computation of the accumulated means and the segmentation only uses 18% on average and 28% at maximum of a single processor core.

8 Conclusions

We presented an on-line segmentation and reconstruction method for CAD geometry from a stream of point data. This method is based on accumulated means that are used for on-line computation of means of geometric properties. An advantage of this approach is that the results can be evaluated immediately by the operator of the hand-held laser scanner.

Currently the approach is limited to planar, cylindrical, and spherical shapes. We plan to extend the approach to other geometric primitives like cones, ellipsoids, tori, and rolling ball blends. Furthermore, an on-line boundary reconstruction of segments could be used to improve the results of the geometry reconstruction. Such boundaries are necessary to reconstruct trimming curves and intersections between geometric primitives.

Acknowledgments

This work was supported by DFG grant UM 26/5-1.

References

1. Agin, G., Binford, T.: Computer description of curved objects. *IEEE Trans. on Computers* C-25(4), 439–449 (1976)
2. Benkő, P., Martin, R.R., Várady, T.: Algorithms for reverse engineering boundary representation models. *Computer-Aided Design* 33(11), 839–851 (2001)
3. Benkő, P., Várady, T.: Direct segmentation of smooth, multiple point regions. In: *Geometric Modeling and Processing*. pp. 169–178. *IEEE* (2002)
4. Biegelbauer, G., Vincze, M.: Efficient 3D object detection by fitting superquadrics to range image data for robot’s object manipulation. In: *International Conference on Robotics and Automation*. pp. 1086–1091. *IEEE* (2007)
5. Bodenmüller, T., Hirzinger, G.: Online surface reconstruction from unorganized 3d-points for the DLR hand-guided scanner system. In: *2nd Symp. on 3D Data Processing, Visualization and Transmission*. pp. 285–292 (2004)
6. Boesch, J., Pajarola, R.: Flexible configurable stream processing of point data. In: *WSCG’2009*. pp. 49–56 (2009)
7. do Carmo, M.P.: *Differential Geometry of Curves and Surfaces*. Pearson (1976)

8. Chan, T., Golub, G., Leveque, R.: Algorithms for computing the sample variance: analysis and recommendations. *The American Statistician* 37(3), 242–247 (1983)
9. Denker, K., Hagel, D., Raible, J., Umlauf, G., Hamann, B.: On-line reconstruction of cad geometry. In: *International Conference on 3D Vision*. pp. 151–158 (2013)
10. Denker, K., Lehner, B., Umlauf, G.: Online triangulation of laser-scan data. In: Garimella, R. (ed.) *17th International Meshing Roundtable*. pp. 415–432 (2008)
11. Denker, K., Lehner, B., Umlauf, G.: Real-time triangulation of point streams. *Engineering with Computers* 27(1), 67–80 (2011)
12. Gronau, I., Moran, S.: Optimal implementations of upgma and other common clustering algorithms. *Information Processing Letters* 104(6), 205 – 210 (2007)
13. Holland, P.W., Welsch, R.E.: Robust Regression Using Iteratively Reweighted Least-Squares. *Com. in Statistics: Theory and Methods A6*, 813–827 (1977)
14. Jolliffe, I.: *Principal Component Analysis*. Springer-Verlag, Heidelberg (2002)
15. Knuth, D.E.: *Seminumerical Algorithms, The Art of Computer Programming*, vol. 2, chap. 4.2.2, p. 232. Addison-Wesley, Boston, third edn. (1998)
16. Ling, R.F.: Comparison of several algorithms for computing sample means and variances. *Journal of the American Statistical Association* 69(348), 859–866 (1974)
17. Page, D., Sun, Y., Koschan, A., Paik, J., Abidi, M.: Normal vector voting: Crease detection and curvature estimation on large, noisy meshes. *Graph. Models* 64, 199–229 (2002)
18. Pajarola, R.: Stream-processing points. In: *IEEE Visualization*. pp. 239–246 (2005)
19. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press (2007)
20. Qt Project Hosting: Qt Documentation (2013),
<http://qt-project.org/doc/qt-5/containers.html#algorithmic-complexity>
21. Razer: Hydra Motion Sensing Controller (2013),
<http://www.razerzone.com/gaming-controllers/razer-hydra>
22. van Reeken, A.J.: Letters to the editor: Dealing with Neely’s algorithms. *Commun. ACM* 11(3), 149–150 (1968)
23. Schnabel, R., Wahl, R., Klein, R.: Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum* 26(2), 214–226 (2007)
24. Shamir, A.: A survey on mesh segmentation techniques. *Computer Graphics Forum* 27(6), 1539–1556 (2008)
25. Sokal, R.R., Michener, C.D.: A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin* 28, 1409–1438 (1958)
26. Vančo, M., Hamann, B., Brunnett, G.: Surface reconstruction from unorganized point data with quadrics. *Computer Graphics Forum* 27(6), 1593–1606 (2008)
27. Várady, T.: Reverse engineering of geometric models – an introduction. *Comp.-Aided Design* 29(4), 255–268 (1997)
28. Welford, B.P.: Note on a Method for Calculating Corrected Sums of Squares and Products. *Technometrics* 4(3), 419–420 (1962)
29. Wu, J., Kobbelt, L.: Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum* 24(3), 277–284 (2005)
30. Yamazaki, I., Natarajan, V., Bai, Z., Hamann, B.: Segmenting point-sampled surfaces. *The Visual Computer* 26(12), 1421–1433 (2010)
31. Zhong, S.: Efficient online spherical k -means clustering. In: *Proc. 2005 IEEE International Joint Conference on Neural Networks*. vol. 5, pp. 3180–3185 (2005)