Efficient RANSAC for Point-Cloud Shape Detection

R. Schnabel, R. Wahl and R. Klein

Universität Bonn, Computer Graphics Group
{schnabel|wahl|rk}@cs.uni-bonn.de

Abstract

In this paper we present an automatic algorithm to detect basic shapes in unorganized point clouds. The algorithm decomposes the point cloud into a concise, hybrid structure of inherent shapes and a set of remaining points. Each detected shape serves as a proxy for a set of corresponding points. Our method is based on random sampling and detects planes, spheres, cylinders, cones and tori. For models with surfaces composed of these basic shapes only, for example, CAD models, we automatically obtain a representation solely consisting of shape proxies. We demonstrate that the algorithm is robust even in the presence of many outliers and a high degree of noise. The proposed method scales well with respect to the size of the input point cloud and the number and size of the shapes within the data. Even point sets with several millions of samples are robustly decomposed within less than a minute. Moreover, the algorithm is conceptually simple and easy to implement. Application areas include measurement of physical parameters, scan registration, surface compression, hybrid rendering, shape classification, meshing, simplification, approximation and reverse engineering.

Keywords: large point-clouds, geometry analysis, shape fitting, localized RANSAC, primitive shapes

ACM CCS: I.4.8: *Scene Analysis Shape; Surface Fitting*, I.3.5: *Computational Geometry and Object Modeling Curve, surface, solid, and object representations*.

1. Introduction

Due to the increasing size and complexity of geometric data sets there is an ever-growing demand for concise and meaningful abstractions of this data. Especially when dealing with digitized geometry, for example, acquired with a laser scanner, no handles for modification of the data are available to the user other than the digitized points themselves. However, in order to be able to make use of the data effectively, the raw digitized data has to be enriched with abstractions and possibly semantic information, providing the user with higher-level interaction possibilities. Only such handles can provide the interaction required for involved editing processes, such as deleting, moving or resizing certain parts and hence can make the data more readily usable for modeling purposes. Of course, traditional reverse engineering approaches can provide some of the abstractions that we seek, but usually reverse engineering focuses on finding a reconstruction of the underlying geometry and typically involves quite tedious user interaction. This is not justified in a setting where a com-

plete and detailed reconstruction is not required at all, or shall take place only after some basic editing operations have been applied to the data. On the other hand, detecting instances of a set of primitive geometric shapes in the point sampled data is a means to quickly derive higher levels of abstraction. For example, in Figure 1 patches of primitive shapes provide a coarse approximation of the geometry that could be used to compress the point-cloud very effectively.

Another problem arising when dealing with digitized geometry is the often huge size of the data sets. Therefore, the efficiency of algorithms inferring abstractions of the data is of utmost importance, especially in interactive settings. Thus, in this paper we focus especially on finding an *efficient* algorithm for point-cloud shape detection, in order to be able to deal even with large point-clouds. Our work is a high performance RANSAC [FB81] algorithm that is capable to extract a variety of different types of primitive shapes, while retaining such favorable properties of the RANSAC paradigm as robustness, generality and simplicity. At the

(a) Original (b) Approximation

Figure 1: *The 372 detected shapes in the choir screen define a coarse approximation of the surface.*

heart of our algorithm are a novel, hierarchically structured sampling strategy for candidate shape generation as well as a novel, lazy cost function evaluation scheme, which significantly reduces overall computational cost. Our method detects planes, spheres, cylinders, cones and tori, but additional primitives are possible. The goal of our algorithm is to reliably extract these shapes from the data, even under adverse conditions such as heavy noise.

As has been indicated above, our method is especially well suited in situations where geometric data is automatically acquired and users refrain from applying surface reconstruction methods, either due to the data's low quality or due to processing time constraints. Such constraints are typical for areas where high level model interaction is required, as is the case when measuring physical parameters or in interactive, semi-automatic segmentation and postprocessing.

Further applications are, for instance, registering many scans of an object, where detecting corresponding primitive shapes in multiple scans can provide good initial matches. High compression rates for point-clouds can be achieved if primitive shapes are used to represent a large number of points with a small set of parameters. Other areas that can benefit from primitive shape information include hybrid rendering and shape classification. Additionally, a fast shape extraction method as ours can serve as building block in applications such as meshing, simplification, approximation and reverse engineering and bears the potential of significant speed up.

2. Previous Work

The detection of primitive shapes is a common problem encountered in many areas of geometry related computer science. Over the years a vast number of methods have been proposed which cannot all be discussed here in depth. Instead, here we give a short overview of some of the most important algorithms developed in the different fields. We treat the previous work on RANSAC algorithms separately in Section 2.1 as it is of special relevance to our work.

Vision. In computer vision, the two most widely known methodologies for shape extraction are the RANSAC paradigm [FB81] and the Hough transform [Hou62]. Both have been proven to successfully detect shapes in 2D as well as 3D. RANSAC and the Hough transform are reliable even in the presence of a high proportion of outliers, but lack of efficiency or high memory consumption remains their major drawback [IK88]. For both schemes, many acceleration techniques have been proposed, but no one on its own, or combinations thereof, have been shown to be able to provide an algorithm as efficient as ours for the 3D primitive shape extraction problem.

The Hough transform maps, for a given type of parameterized primitive, every point in the data to a manifold in the parameter space. The manifold describes all possible variants of the primitive that contain the original point, that is, in practice each point casts votes for many cells in a discretized parameter space. Shapes are extracted by selecting those parameter vectors that have received a significant amount of votes. If the parameter space is discretized naively using a simple grid, the memory requirements quickly become prohibitive even for primitives with a moderate number of parameters, such as, for instance, cones. Although several methods have been suggested to alleviate this problem [IK87,XO93] its major application area remains the 2D domain where the number of parameters typically is quite small. A notable exception is Ref. [VGS*04], where the Hough transform is used to detect planes in 3D data sets, as 3D planes still have only a small number of parameters. They also propose a two-step procedure for the Hough based detection of cylinders that uses estimated normals in the data points.

In the vision community many approaches have been proposed for segmentation of range images with primitive shapes. When working on range images these algorithms usually efficiently exploit the implicitly given connectivity information of the image grid in some kind of region growing or region merging step [FEF97,GBS03]. This is a fundamental difference to our case, where we are given only an unstructured cloud of points that lacks any explicit connectivity information. In Refs. [LGB95] and [LJS97] shapes are found by concurrently growing different seed primitives from which a suitable subset is selected according to an MDL criterion (coined the recover-and-select paradigm). Gotardo

et al. [GBS03] detect shapes using a genetic algorithm to optimize a robust MSAC fitness function (see also Section 2.1). Marshall *et al.* [MLM01] introduce involved non linear fitting functions for primitive shapes that are able to handle geometric degeneracy in the context of recover-and-select segmentation.

Another robust method frequently employed in the vision community is the tensor voting framework [MLT00] which has been applied to successfully reconstruct surface geometry from extremely cluttered scenes. While tensor voting can compete with RANSAC in terms of robustness, it is, however, inherently model-free and therefore cannot be applied to the detection of predefined types of primitive shapes.

Reverse engineering. In reverse engineering, surface recovery techniques are usually based on either a separate segmentation step or on a variety of region growing algorithms [VMC97–BGV*02]. Most methods call for some kind of connectivity information and are not well equipped to deal with a large amount of outliers [VMC97]. Also these approaches try to find a shape proxy for every part of the processed surface with the intent of loading the reconstructed geometry information into a CAD application. Banko *et al.* [BMV01] describe a system which reconstructs a boundary representation that can be imported into a CAD application from an unorganized point-cloud. However, their method is based on finding a triangulation for the point-set, whereas the method presented in this work is able to operate directly on the input points. This is advantageous as computing a suitable tessellation may be extremely costly and becomes very intricate or even ill-defined when there is heavy noise in the data. We do not, however, intend to present a method implementing all stages of a typical reverse engineering process.

Graphics. In computer graphics, Cohen-Steiner *et al.* [CAD04] have recently proposed a general variational framework for approximation of surfaces by planes, which was extended to a set of more elaborate shape proxies by Wu and Kobbelt [WK05]. Their aim is not only to extract certain shapes in the data, but to find a globally optimal representation of the object by a given number of primitives. However, these methods require connectivity information and are, due to their exclusive use of least-squares fitting, susceptible to errors induced by outliers. Also, the optimization procedure is computationally expensive, which makes the method less suitable for large data sets. The output of our algorithm, however, could be used to initialize the set of shape proxies used by these methods, potentially accelerating the convergence of the optimization procedure. While the Hough transform and the RANSAC paradigm have been mainly used in computer vision some applications have also been proposed in the computer graphics community. Décoret *et al.* [DDS*03] employ the Hough transform to identify planes for billboard clouds for triangle data. They propose an extension of the standard Hough transform to include a compactness criterion, but due to the high computational demand of the Hough

transform, the method exhibits poor runtime performance on large or complex geometry. Wahl *et al.* [WGK05] proposed a RANSAC-based plane detection method for hybrid rendering of point-clouds. To facilitate an efficient plane detection, planes are detected only in the cells of a hierarchical space decomposition and therefore what is essentially one plane on the surface is approximated by several planar patches. While this is acceptable for their hybrid rendering technique, our method finds maximal surface patches in order to yield a more concise representation of the object. Moreover, higher order primitives are not considered in their approach. Gelfand and Guibas [GG04] detect so-called slippable shapes which is a superset of the shapes recognized by our method. They use the eigenvalues of a symmetric matrix derived from the points and their normals to determine the slippability of a point-set. Their detection is a bottom-up approach that merges small initial slippable surfaces to obtain a global decomposition of the model. However, the computation of the eigenvalues is costly for large models, the method is sensitive to noise and it is hard to determine the correct size of the initial surface patches. A related approach is taken by Ref. [HOP*05]. They also use the eigenvalues of a matrix derived from line element geometry to classify surfaces. A RANSAC based segmentation algorithm is employed to detect several shapes in a point-cloud. The method is aimed mainly at models containing small numbers of points and shapes as no optimizations or extensions to the general RANSAC framework are adopted.

2.1. RANSAC

The RANSAC paradigm extracts shapes by randomly drawing minimal sets from the point data and constructing corresponding shape primitives. A minimal set is the smallest number of points required to uniquely define a given type of geometric primitive. The resulting candidate shapes are tested against all points in the data to determine how many of the points are well approximated by the primitive (called the *score* of the shape). After a given number of trials, the shape which approximates the most points is extracted and the algorithm continues on the remaining data. RANSAC exhibits the following, desirable properties:

- It is conceptually simple, which makes it easily extensible and straightforward to implement.

- It is very general, allowing its application in a wide range of settings.

- It can robustly deal with data containing more than 50% of outliers [RL93].

Its major deficiency is the considerable computational demand if no further optimizations are applied.

Bolles and Fischler [BF81] apply RANSAC to extract cylinders from range data, Chaperon and Goulette [CG01]

use RANSAC and the gaussian image to find cylinders in 3D point-clouds. Both methods, though, do not consider a larger number of different classes of shape primitives. Roth and Levine [RL93] describe an algorithm that uses RANSAC to detect a set of different types of simple shapes. However, their method was adjusted to work in the image domain or on range images, and they did not provide the optimization necessary for processing large unstructured 3D data sets.

A vast number of extensions to the general RANSAC scheme have been proposed. Among the more recent advances, methods such as MLESAC [TZ00] or MSAC [TZ98] improve the robustness of RANSAC with a modified score function, but do not provide any enhancement in the performance of the algorithm, which is the main focus of our work. Nonetheless, the integration of a MLESAC scoring function is among the directions of our future work. Nister [Nis05] proposes an acceleration technique for the case that the number of candidates is fixed in advance. As it is a fundamental property of our setup that an unknown large number of possibly very small shapes has to be detected in huge point-clouds, the amount of necessary candidates cannot, however, be specified in advance.

3. Overview

Given a point-cloud $\mathcal{P} = \{p_1 \ldots, p_N\}$ with associated normals $\{n_1, \ldots, n_N\}$ the output of our algorithm is a set of primitive shapes $\Psi = \{\psi_1, \ldots, \psi_n\}$ with corresponding disjoint sets of points $\mathcal{P}_{\psi_1} \subset \mathcal{P}, \ldots, \mathcal{P}_{\psi_n} \subset \mathcal{P}$ and a set of remaining points $\mathcal{R} = \mathcal{P} \setminus \{\mathcal{P}_{\psi_1}, \ldots, \mathcal{P}_{\psi_n}\}$. Similar to Refs. [RL93] and [DDS*03], we frame the shape extraction problem as an optimization problem defined by a score function. The overall structure of our method is outlined in pseudocode in algorithm 1. In each iteration of the algorithm, the primitive with maximal score is searched using the RANSAC paradigm. New shape candidates are generated by randomly sampling minimal subsets of $\mathcal{P}$ using our novel sampling strategy (see Section 4.3). Candidates of *all* considered shape types are generated for *every* minimal set and all candidates are collected in the set $\mathcal{C}$. Thus no special ordering has to be imposed on the detection of different types of shapes. After new candidates have been generated the one with the highest score m is computed employing the efficient lazy score evaluation scheme presented in Section 4.5. The best candidate is only accepted if, given the size $|m|$ (in number of points) of the candidate and the number of drawn candidates $|\mathcal{C}|$, the probability $P(|m|, |\mathcal{C}|)$ that no better candidate was overlooked during sampling is high enough (see Section 4.2.1). We provide an analysis of our sampling strategy to derive a suitable probability computation. If a candidate is accepted, the corresponding points $\mathcal{P}_m$ are removed from $\mathcal{P}$ and the candidates $\mathcal{C}_m$ generated with points in $\mathcal{P}_m$ are deleted from $\mathcal{C}$. The algorithm terminates as soon as $P(\tau, |\mathcal{C}|)$ for a user defined minimal shape size τ is large enough.

In our implementation we use a standard score function that counts the number of compatible points for a shape candidate [GBS03,RL93]. The function has two free parameters: ϵ specifies the maximum distance of a compatible point while α restricts the deviation of a points' normal from that of the shape. We also ensure that only points forming a connected component on the surface are considered (see Section 4.4).

Algorithm 1 Extract shapes in the point cloud $\mathcal{P}$

$\Psi \leftarrow \emptyset$ {extracted shapes}
$\mathcal{C} \leftarrow \emptyset$ {shape candidates}
repeat
   $\mathcal{C} \leftarrow \mathcal{C} \cup$ newCandidates() {see sec. 4.1 and 4.3}
   $m \leftarrow$ bestCandidate($\mathcal{C}$) {see sec. 4.4}
   if $P(|m|, |\mathcal{C}|) > p_t$ **then**
      $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{P}_m$ {remove points}
      $\Psi \leftarrow \Psi \cup m$
      $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{C}_m$ {remove invalid candidates}
   end if
until $P(\tau, |\mathcal{C}|) > p_t$
return Ψ

---

4. Our Method

4.1. Shape estimation

As mentioned above, the shapes we consider in this work are planes, spheres, cylinders, cones and tori which have between three and seven parameters. Every 3D-point p_i sample fixes only one parameter of the shape. In order to reduce the number of required points we compute an approximate surface normal n_i for each point [HDD*92], so that the orientation gives us two more parameters per sample. That way it is possible to estimate each of the considered basic shapes from only one or two point samples. However, always using one additional sample is advantageous, because the surplus parameters can be used to immediately verify a candidate and thus eliminate the need of evaluating many relatively low scored shapes [MC02].

Plane. For a plane, $\{p_1, p_2, p_3\}$ constitutes a minimal set when not taking into account the normals in the points. To confirm the plausibility of the generated plane, the deviation of the plane's normal from n_1, n_2, n_3 is determined and the candidate plane is accepted only if all deviations are less than the predefined angle α.

Sphere. A sphere is fully defined by two points with corresponding normal vectors. We use the midpoint of the shortest line segment between the two lines given by the points p_1 and p_2 and their normals n_1 and n_2 to define the center of the sphere c. We take $r = \frac{\|p_1 - c\| + \|p_2 - c\|}{2}$ as the sphere radius. The sphere is accepted as a shape candidate only if all three points are within a distance of ε of the sphere and their normals do not deviate by more than α degrees.

Cylinder. To generate a cylinder from two points with normals we first establish the direction of the axis with $a = n_1 \times n_2$. Then we project the two parametric lines $p_1 + tn_1$ and $p_2 + tn_2$ along the axis onto the $a \cdot x = 0$ plane and take their intersection as the center c. We set the radius to the distance between c and p_1 in that plane. Again the cylinder is verified by applying the thresholds ϵ and α to distance and normal deviation of the samples.

Cone. Although the cone, too, is fully defined by two points with corresponding normals, for simplicity we use all three points and normals in its generation. To derive the position of the apex c, we intersect the three planes defined by the point and normal pairs. Then the normal of the plane defined by the three points $\{c + \frac{p_1 - c}{\|p_1 - c\|}, \ldots, c + \frac{p_3 - c}{\|p_3 - c\|}\}$ gives the direction of the axis a. Now the opening angle ω is given as $\omega = \frac{\sum_i \arccos((p_i - c) \cdot a)}{3}$. Afterwards, similar to above, the cone is verified before becoming a candidate shape.

Torus. Just as in the case of the cone we use one more point than theoretically necessary to ease the computations required for estimation, that is, four point and normal pairs. The rotational axis of the torus is found as one of the up to two lines intersecting the four point-normal lines $p_i + \lambda n_i$ [MLM01]. To choose between the two possible axes, a full torus is estimated for both choices and the one which causes the smaller error in respect to the four points is selected. To find the minor radius, the points are collected in a plane that is rotated around the axis. Then a circle is computed using three points in this plane. The major radius is given as the distance of the circle center to the axis.

4.2. Complexity

The complexity of RANSAC is dominated by two major factors. The number of minimal sets that are drawn and the cost of evaluating the score for every candidate shape. As we desire to extract the shape that achieves the highest possible score, the number of candidates that have to be considered is governed by the probability that the best possible shape is indeed detected, that is, that a minimal set is drawn that defines this shape.

4.2.1. Probabilities

Consider a point cloud $\mathcal{P}$ of size N and a shape ψ therein consisting of n points. Let k denote the size of a minimal set required to define a shape candidate. If we assume that any k points of the shape will lead to an appropriate candidate shape then the probability of detecting ψ in a single pass is:

$$P(n) = \binom{n}{k} \bigg/ \binom{N}{k} \approx \left(\frac{n}{N}\right)^k. \qquad (1)$$

The probability of a successful detection $P(n, s)$ after s candidates have been drawn equals the complementary of s consecutive failures:

$$P(n, s) = 1 - (1 - P(n))^s. \qquad (2)$$

Solving for s tells us the number of candidates T required to detect shapes of size n with a probability $P(n, T) \geq p_t$:

$$T \geq \frac{\ln(1 - p_t)}{\ln(1 - P(n))}. \qquad (3)$$

For small $P(n)$ the logarithm in the denominator can be approximated by its Taylor series $\ln(1 - P(n)) = -P(n) + O(P(n)^2)$ so that:

$$T \approx \frac{-\ln(1 - p_t)}{P(n)}. \qquad (4)$$

Given the cost C of evaluating the cost function, the asymptotic complexity of the RANSAC approach is $O(TC) = O(\frac{1}{P(n)}C)$.

4.3. Sampling strategy

As can be seen from the last formula, the runtime complexity is directly linked to the success rate of finding good sample sets. Therefore, we will now discuss in detail how sampling is performed.

4.3.1. Localized sampling

Since shapes are local phenomena, the *a priori* probability that two points belong to the same shape is higher the smaller the distance between the points. In our sampling strategy we want to exploit this fact to increase the probability of drawing minimal sets that belong to the same shape. Myatt *et al.* [MTN*02] have shown that non uniform sampling based on locality leads to a significantly increased probability of selecting a set of inliers. From a ball of given radius around an initially unrestrainedly drawn sample the remaining samples are picked to obtain a complete minimal set. This requires to fix a radius in advance, which they derive from a known (or assumed) outlier density and distribution. In our setup however, outlier density and distribution vary strongly for different models and even within in a single model, which renders a fixed radius inadequate. Also, in our case, using minimal sets with small diameter introduces unnecessary stability issues in the shape estimation procedure for shapes that could have been estimated from samples spread farther apart. Therefore, we propose a novel sampling strategy that is able to adapt the diameter of the minimal sets to both, outlier density and shape size.

We use an octree to establish spatial proximity between samples very efficiently. When choosing points for a new candidate, we draw the first sample p_1 without restrictions among all points. Then a cell C is randomly chosen from any
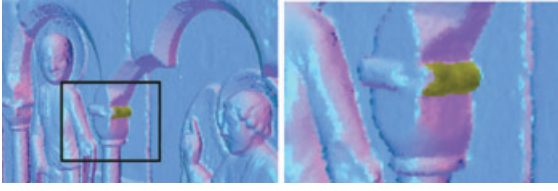
Figure 2: *A small cylinder that has been detected by our method. The shape consists of 1066 points and was detected among 341,587 points. That corresponds to a relative size of 1/3000.*

level of the octree such that p_1 is contained in C. The $k - 1$ other samples are then drawn only from within cell C.

The effect of this sampling strategy can be expressed in a new probability $P_{local}(n)$ for finding a shape ψ of size n:

$$P_{local}(n) = P(p_1 \in \psi)P(p_2 \dots p_k \in \psi | p_2 \dots p_k \in C). \tag{5}$$

The first factor evaluates to n/N. The second factor obviously depends on the choice of C. C is well chosen if it contains mostly points belonging to ψ. The existence of such a cell is backed by the observation that for most points on a shape, except on edges and corners, there exists a neighborhood such that all of the points therein belong to that shape. Although in general it is not guaranteed that this neighborhood is captured in the cells of the octree, in the case of real-life data, shapes have to be sampled with an adequate density for reliable representation and, as a consequence, for all but very few points such a neighborhood will be at least as large as the smallest cells of the octree. For the sake of analysis, we assume that there exists a C for every $p_i \in \psi$ such that ψ will be supported by half of the points in C, which accounts for up to 50% local noise and outliers. We conservatively estimate the probability of finding a good C by $\frac{1}{d}$, where d is the depth of the octree (in practice a path of cells starting at the highest good cell to a good leaf will be good as well). The conditional probability for $p_2, p_3 \in \psi$ in the case of a good cell is then described by $\frac{\binom{|C|/2}{k-1}}{\binom{|C|}{k-1}} \approx (\frac{1}{2})^{k-1}$. And substituting yields:

$$P_{local}(n) = \frac{n}{Nd2^{k-1}}. \tag{6}$$

As large shapes can be estimated from large cells (and with high probability this will happen), the stability of the shape estimation is not affected by the sampling strategy.

The impact of this sampling strategy is best illustrated with an example. The cylinder depicted in Figure 2 consists of 1066 points. At the time that it belongs to one of the largest shapes in the point-cloud, 341,547 points of the original 2 million still remain. Thus, it then comprises only three thousandth of the point-cloud. If an ordinary uniform sampling strategy were to be applied, 151,522,829 candidates would

have to be drawn to achieve a detection probability of 99%. With our strategy only 64,929 candidates have to be generated for the same probability. That is an improvement by three orders of magnitude, that is, in this case that is the difference between hours and seconds.

4.3.1.1. Level weighting. Choosing C from a proper level is an important aspect of our sampling scheme. Therefore we can further improve the sampling efficiency by choosing C from a level according to a non uniform distribution that reflects the likelihood of the respective level to contain a good cell. To this end, the probability P_l of choosing C from level l is first initialized with $\frac{1}{d}$. Then for every level l, we keep track of the sum σ_l of the scores achieved by the candidates generated from a cell on level l. After a given number of candidates has been tested, a new distribution for the levels is computed. The new probability $\hat{P}_l$ of the level l is given as

$$\hat{P}_l = x \frac{\sigma_l}{w P_l} + (1 - x) \frac{1}{d} \tag{7}$$

where $w = \sum_{i=1}^{d} \frac{\sigma_i}{P_i}$. We set $x = .9$ to ensure that at all times at least 10% of the samples are spread uniformly over the levels to be able to detect when new levels start to become of greater importance as more and more points are removed from $\mathcal{P}$.

4.3.2. Number of candidates

In Section 4.2, we gave a formula for the number of candidates necessary to detect a shape of size n with a given probability. However, in our case, the size n of the largest shape is not known in advance. Moreover, if the largest candidate has been generated early in the process we should be able to detect this lucky case and extract the shape well before achieving a precomputed number of candidates while on the other hand we should use additional candidates if it is still unsure that indeed the best candidate has been detected. Therefore, instead of fixing the number of candidates, we repeatedly analyze small numbers t of additional candidates and consider the best one ψ_m generated so far each time. As we want to achieve a low probability that a shape is extracted which is not the real maximum, we observe the probability $P(|\psi_m|, s)$ with which we would have found another shape of the same size as ψ_m. Once this probability is higher than a threshold p_t (we use 99%) we conclude that there is a low chance that we have overlooked a better candidate and extract ψ_m. The algorithm terminates as soon as $P(\tau, s) > p_t$.

4.4. Score

The score function $\sigma_{\mathcal{P}}$ is responsible for measuring the quality of a given shape candidate. We use the following aspects in our scoring function:

- To measure the support of a candidate, we use the number of points that fall within an ε-band around the shape.

- To ensure that the points inside the band roughly follow the curvature pattern of the given primitive, we only count those points inside the band whose normals do not deviate from the normal of the shape more than a given angle α.

- Additionally, we incorporate a connectivity measure: Among the points that fulfill the previous two conditions, only those are considered that constitute the largest connected component on the shape.

More formally, given a shape ψ whose fidelity is to be evaluated, $\sigma_{\mathcal{P}}$ is defined as follows:

$$\sigma_{\mathcal{P}}(\psi) = |\mathcal{P}_{\psi}|,$$

that is, we count the number of points in $\mathcal{P}_{\psi}$. $\mathcal{P}_{\psi}$ is defined in the following two steps:

$$\hat{\mathcal{P}}_{\psi} = \{p \mid p \in \mathcal{P} \land d(\psi, p)$$
$$< \varepsilon \land \arccos(|n(p) \cdot n(\psi, p)|) < \alpha\}$$

$$\mathcal{P}_{\psi} = maxcomponent(\psi, \hat{\mathcal{P}}_{\psi})$$

where $d(\psi, p)$ is the euclidian distance between the point p and the shape primitive ψ, $n(p)$ is the normal in p and $n(\psi, p)$ is the normal of ψ in p's projection on ψ. $maxcomponent(\psi, \hat{\mathcal{P}}_{\psi})$ extracts the group of points in $\hat{\mathcal{P}}_{\psi}$ whose projections onto ψ belong to the largest connected component on ψ.

4.4.1. Connected components

We find connected components in a bitmap located in the parameter domain of the shape. A pixel in the bitmap is set if a point is projected into it. Ideally, the size β of the pixels in the bitmap should correspond to the distance between neighboring points in the data, that is, the sampling resolution. If the data is irregularly sampled, β should be chosen as the minimal sampling resolution satisfied everywhere in the data.

The parametrization of the bitmap is straightforward for the plane and the cylinder. In case of the sphere we use one bitmap for every hemisphere. For the hemispheres we use the low distortion parametrization given by [SC97]. We choose between two different parameterizations when constructing a bitmap on a cone. Flat cones are parameterized by angle and distance to the apex, yielding a circle in the plane. For acute opening angles this produces overly large bitmaps if the points are far from the apex, therefore we parametrize the cone by distance to the apex and arc length in such cases, resulting in a triangular domain in the bitmap.

4.5. Score evaluation

The second major performance factor of the RANSAC scheme is the score function evaluation. In our case, in a naïve implementation, the distance to all points in $\mathcal{P}$ would

have to be computed together with a normal at a corresponding position on the shape for each candidate. Then the largest connected component has to be found among all compatible points.

4.5.1. Random subsets

Obviously, the cost of evaluation would be prohibitive without any optimizations. But since in each run we are only interested in the candidate that achieves the highest score, using the entire point-cloud $\mathcal{P}$ when computing $\sigma_{\mathcal{P}}(\psi)$ is not necessary for every shape candidate. We significantly reduce the number of points that have to be considered in the evaluation of $\sigma_{\mathcal{P}}(\psi)$ by splitting the point-cloud $\mathcal{P}$ into a set of disjoint random subsets: $\mathcal{P} = \{\mathcal{S}_1 \ldots \mathcal{S}_r\}$.

After a shape candidate was generated and successfully verified, the candidate is only scored against the first subset $\mathcal{S}_1$ and no connected component is extracted yet. From the score on the first subset $\sigma_{\mathcal{S}_1}(\psi)$ an estimate $\hat{\sigma}_{\mathcal{P}}(\psi)$ for the score $\sigma_{\mathcal{P}}(\psi)$ on all points can be extrapolated using the well known induction from inferential statistics:

$$\hat{\sigma}_{\mathcal{P}}(\psi) = -1 - f(-2 - |\mathcal{S}_1|, -2 - |\mathcal{P}|, -1 - \sigma_{\mathcal{S}_1}(\psi)) \tag{8}$$

where

$$f(N, x, n) = \frac{xn \pm \sqrt{\frac{xn(N-x)(N-n)}{N-1}}}{N} \tag{9}$$

is the mean plus/minus the standard deviation of the hypergeometric distribution. $\hat{\sigma}_{\mathcal{P}}(\psi)$ is a confidence interval $[a, b]$ that describes a range of likely values for the true score $\sigma_{\mathcal{P}}(\psi)$. The expected value $E(\sigma_{\mathcal{P}}(\psi))$ is given by $\frac{a+b}{2}$. With this extrapolation the potentially best candidate ψ_m can be quickly identified by choosing the one with the highest expected value. Since the uncertainty of the estimation is captured in the confidence intervals, the truly maximal candidate can be found by comparing the confidence intervals of the candidates.

If the confidence intervals of ψ_m and another candidate ψ_i overlap, the score on an additional subset is evaluated for both candidates and new extrapolations are computed, now taking into account the scores on all subsets that have already been computed:

$$\hat{\sigma}_{\mathcal{P}}(\psi) = -1$$
$$- f\left(-2 - \sum_i |\mathcal{S}_i|, -2 - |\mathcal{P}|, -1 - \sum_i \sigma_{\mathcal{S}_i}(\psi)\right).$$

The more subsets have been considered, the smaller becomes the range of the confidence intervals, since the uncertainty in the estimation decreases. Further subsets are included until the confidence intervals of ψ_i and ψ_m no longer overlap and it can be decided if either ψ_i or ψ_m is better.

To include the effect of the connectedness condition in the extrapolation, every time an additional subset has been

Table 1. *Statistics on processed models. ϵ is given as ratio of maximum bounding box width. Results have been averaged over 5 runs and rounded.*

Model	$|\mathcal{P}|$	ε	α	τ	$|\Psi|$	$|\mathcal{R}|$	sec
Fandisk	12k	0.01	10	50	24	38	0.57
Rocker arm	40k	0.003	20	50	73	1k	6.5
Carter	546k	0.001	20	200	138	47k	29.1
Rolling stage	606k	0.003	20	300	61	16k	15.1
Oil pump	542k	0.0015	30	100	202	15k	30.9
Master cyl.	418k	0.003	35	300	37	7k	12.1
House	379k	0.002	20	100	130	19k	10.7
Church	1,802k	0.002	20	1000	160	690k	40.7
Choir screen	1,922k	0.002	20	4,000	81	543k	20.8
					500	372	236k	61.5

evaluated, the maximal connected component is found among all the compatible points that have been discovered so far. The resolution of the bitmap that is used to find the components has to be adapted to reflect the lower sampling rate of the subsets. If $\frac{1}{x} = \frac{\sum_i |\mathcal{S}_i|}{|\mathcal{P}|}$ is the fraction of points in $\mathcal{P}$ that have been tested so far, then the bitmap resolution is adjusted to $x\beta$.

4.5.2. Octree

For each of the subsets an octree is constructed, so that during the cost function evaluation only the points lying in cells within ε distance to the shape have to be considered.

4.6. Refitting

When a candidate shape ψ has been selected for extraction, a refitting step is executed before the shape is finally accepted. As is the standard in RANSAC based algorithms, we use a least-squares approach [Sha98]. This optimizes the geometric error of the candidate shape. In refitting and extraction we include all compatible points within a distance of 3ε from the shape, as this removes unnecessary clutter from the point-cloud [GBS03].

5. Results

We have run extensive tests of our algorithm on different kind of geometry. The results show that basic shapes are reliably detected if they are present in the data. For parts of a surface that closely resemble a basic shape, a well approximating representation is obtained. More involved areas are partitioned into basic shapes in a reasonable manner and the number of remaining points reflects the complexity of the surface. The algorithm exhibits high performance as is shown in the timings of Table 1.

To illustrate the effect of the two major optimizations employed by our algorithm, we disabled them independently
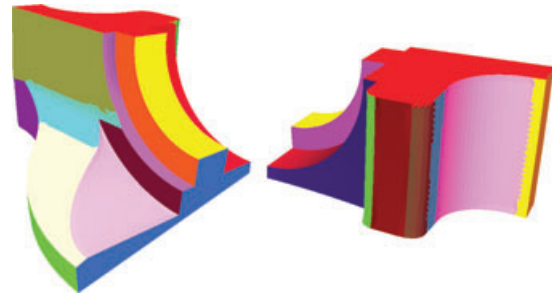

Figure 3: *To generate this image our algorithm was applied to the barycenters of the triangles. The triangles were than colored according to the shape of their barycenter and the vertices were projected onto the shape. The jagged lines appear because the triangulation does not contain the edges of the shapes.*
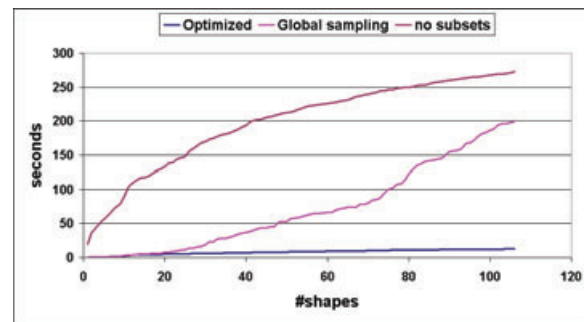

Figure 4: *The chart shows the times of detection of the shapes found in the oil pump model when either subset evaluation or the localized sampling is disabled. For comparison also the timings of the fully optimized version are plotted. Total runtime for the version without subsets was 272.5s, 199.1s without local sampling and 12.3s with both optimizations activated.*

and compare the resulting timings to the optimized version in Figure 4. The timings were obtained with the parameters given in Table 1. The gain of the localized sampling strategy depends on the relative sizes of the shapes with respect to the model. The localized sampling improves the performance especially for smaller shapes detected later on in the process, as their relative size decreases and a global sampling strategy requires far more candidates. Conversely, the score evaluation on subsets has a greater impact early on in the process when the point-cloud and the shapes are both very large and thus many distance evaluations on octree cells and points have to be performed. In general, the gain of the score evaluation on subsets increases with the size of the model. This way these two optimizations complement one another leading to significant performance gains at all stages of the detection.
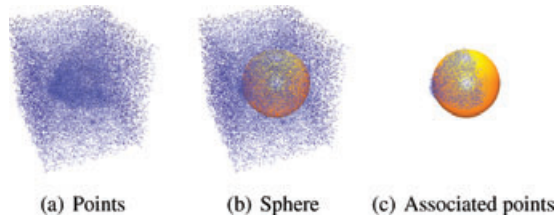
Figure 5: *Points on an octant of a sphere distorted by synthetic gaussian noise with a σ of 10% relative to the sphere diameter and 80% outliers. Our algorithm is able to robustly detect the sphere, see also Table 2.*
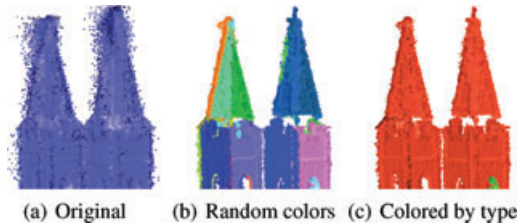
Figure 6: *(a) A part of the church model containing heavy noise. (b) Points belonging to shapes in random colors (c) Points colored by type of shape as in Figure 7.*

The choir screen in Figure 1 consists of a number of basic shapes, for example, large planar areas, cylindrical or conical pillars, and more detailed parts such as the persons. The data was obtained with a laser range scanner and consists of several registered scans. In addition to some noise there also are some registration errors. The planar regions and the pillars are well detected but even for the persons good approximating shapes are found. Note that the small casks in the hands of the two leftmost persons are reliably detected despite their small size relative to the whole data set. The image was generated by projecting all points on their corresponding basic shape. For parameter values see Table 1.

In contrast, the surface of the fandisk in Figure 3 is entirely composed of basic shapes. As expected, our algorithm is able to find these shapes and decomposes the surface into the constituting parts without leaving any remaining points. Note that the result is practically identical to that obtained by Ref. [WK05]. The oil pump on the other hand does not only consist of the basic shapes detected by our method. However, as is shown in Figure 7, existent basic shapes are well detected and only areas of blending patches or small-scale details are ignored. The main characteristics of the model are captured concisely.

In Figure 9 additional results are presented. Outliers are ignored successfully and fine detail geometry is well recognized.

Table 2. *Parameter errors for the fitted spheres under different noise conditions compared to ground truth. All values are given in percent of the sphere diameter. The values are, from left to right, the level of gaussian noise σ, the percentage of outliers o, the mean values μ_r for radius and μ_c for center deviation, and the standard deviations σ_r of the radius and σ_c of the center.*

|		Before refitting				After refitting			
σ	o	μ_r	μ_c	σ_r	σ_c	μ_r	μ_c	σ_r	σ_c
0	0	0.02	0.02	0.02	0.02	0.00	0.00	0.00	0.00
1	25	1.73	2.24	1.16	0.95	0.07	0.07	0.004	0.004
2	25	1.79	2.60	1.43	1.47	0.31	0.31	0.021	0.023
5	50	2.25	3.96	1.65	2.19	0.35	0.26	0.054	0.025
10	50	8.11	11.87	4.15	4.44	4.32	7.20	0.53	0.91
10	80	7.17	10.58	5.18	4.68	5.12	5.99	1.65	1.96

5.1. Noise

We have performed a simple experiment to demonstrate the ability of our method to handle noisy data. We use points of an octant of a sphere with increasing amount of synthetic gaussian noise and outliers as test cases. The points' normals have been obtained by locally fitting a least-squares plane to each sample's neighbors, and the radius in which the neighbors are collected is increased in accordance to the noise ratio. Note that no shape types have been deactivated during the detection, but we did not allow tori with a major radius smaller than the minor one because these form a superset of spheres. In Figure 5 an example with 10% noise and 80% outliers is depicted. Outliers are generated with a uniform distribution over the bounding box. Table 2 lists additional results together with the respective noise ratios. As can be seen the original sphere parameters are reliably detected even for a large amount of noise as well as outliers. For a noise degree higher than 20% the detection starts to become unstable and the algorithm sometimes returns false shape types. Up to 95% outliers are tolerated for a noise degree of 2%.

Figure 8b and c show the behavior of our algorithm on noisy data for a more complex model. The oil-pump model was distorted by synthetic Gaussian noise with σ equalling 1% of the bounding box diagonal. For such heavy noise we observe that only small or very narrow shapes (e.g. some of the screw heads) can no longer be reliably segmented since not enough support can be gathered for them. Another reason is that the estimated normals become too unreliable for these shapes, as the number of neighbor points used for the estimation needs to be increased in order to smooth out the effect of the noise. Therefore points belonging to adjoining shapes adversely influence the estimated normals for such small shapes. The larger shapes, however, are unaffected and therefore are successfully and stably detected despite the heavy noise. In Figure 8d) in addition to the noise 10% of the points have been randomly repositioned as outliers (again by a uniform distribution over the bounding box). As expected,

(a) Original *(b) Random colors* *(c) Colored by type* *(d) Bitmaps*
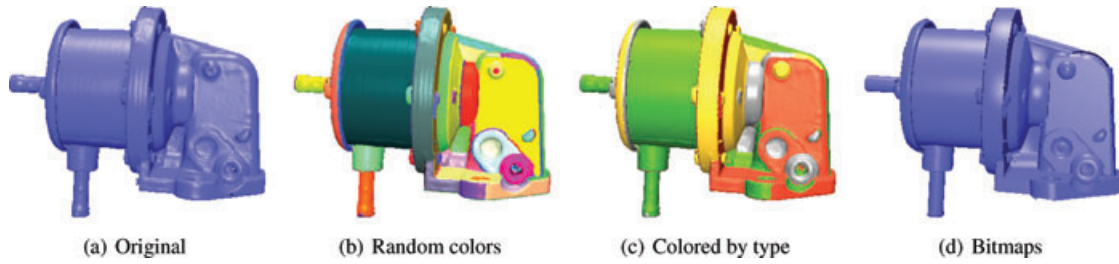
Figure 7: *(a) The original scanned model with ca. 500k points. (b) Points belonging to shapes in random colors. (c) Points of the shapes colored according to the type of the shape they have been assigned to: planes are red, cylinders green, spheres yellow, cones purple and tori are grey. No remaining points are shown. (d) The bitmaps constructed for connected component computations provide a rough reconstruction of the object.*

(a) Noisy original *(b) Random colors* *(c) Colored by type* *(d) Random colors*
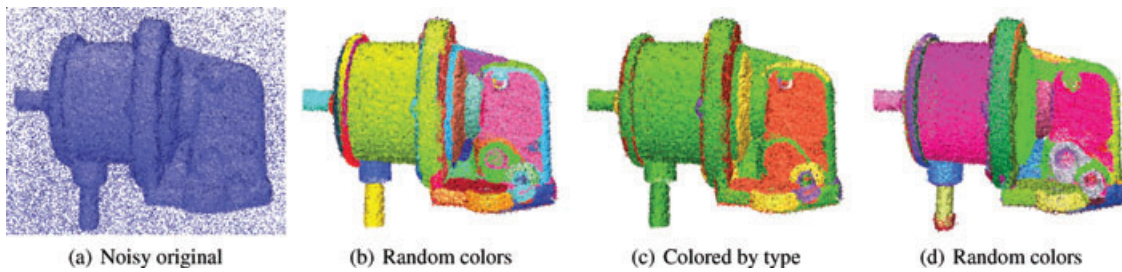
Figure 8: *(a) Distorted model with Gaussian noise and outliers (b–c) Results of the detection on the model with Gaussian noise but without added outliers. (d) In addition to the Gaussian noise, 10% outliers were added (see a)).*
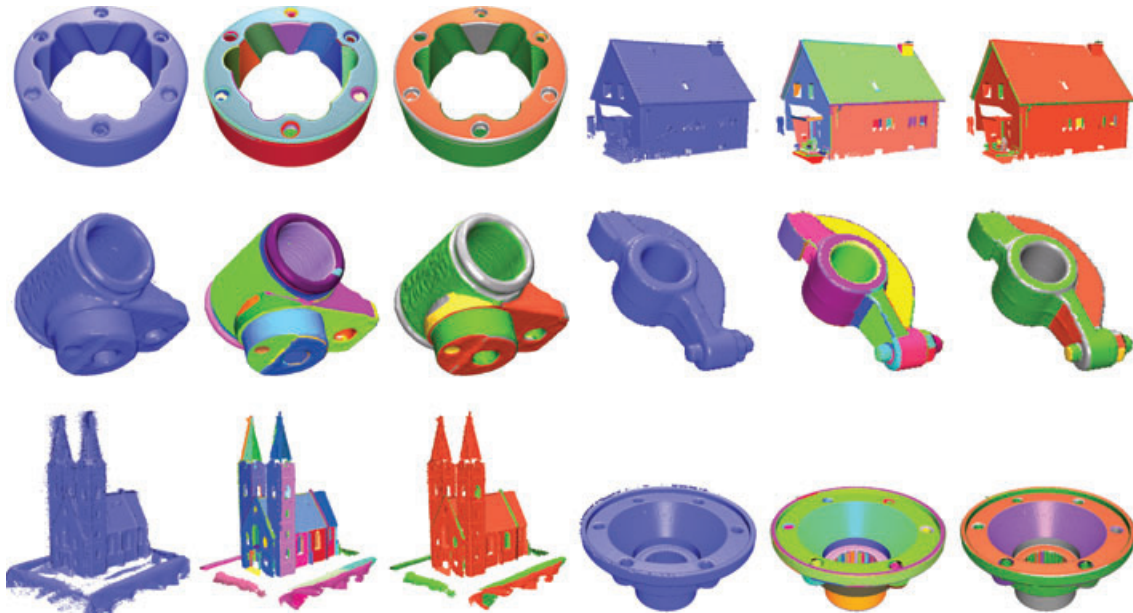
Figure 9: *First column: Original point-clouds. Second column: Shapes colored randomly. Last column: Shapes colored by type as in Figure 7. Models are from top to bottom and left to right: rolling stage, house, master cylinder, rocker arm, church and carter. For parameters and timings see Table 1.*

Table 3. *Average percentage of correctly detected regions on the 40 test range images of the Segmentation Comparison Project. The threshold controls the ratio of required overlap between ground truth and machine segmented regions.*

Threshold	[BJ88]	[JB97]	Our
0.7	16.65%	66.93%	71.06%
0.8	16.12%	66.50%	68.04%
0.9	16.14%	62.42%	62.35%

the detection successfully ignores these outliers and the result is equivalent to that without outliers.

A real world example of heavy noise is demonstrated in Figure 6. Our method is able to successfully detect the planes comprising the roofs of the two spires despite the numerous outliers and noise artifacts.

5.2. Comparison

Comparing our algorithm to other existing methods operating on 3D point-clouds is difficult. Most point-cloud segmentation algorithms do not explicitly detect instances of primitive shapes but find areas that are consistent in some other, predefined way. Moreover, there are hardly any publicly available implementations or results. However, a benchmark for range image segmentation does exist (http://marathon. csee.usf.edu/seg-comp/SegComp.html) [PBJ*98]. Therefore, we resort to this benchmark to provide some comparisons of our method with other existing algorithms. Please note though, that the original intent of our method is not range image segmentation and the benchmark can therefore reflect only a subset of our method's abilities. The benchmark provides test images in conjunction with manually obtained ground truth segmentations as well as a tool for automatic evaluation. Due to the low quality of the used scanner the test images contain heavy systematic noise. Range image segmentation, of course, is a very special and simpler case of our far more general problem formulation that would allow for many extra optimizations and assumptions in the algorithm. Nonetheless, we have tested our method *as is* on the test images provided by the benchmark. Table 3 lists our detection results as well as those of the other two state-of-the-art methods for which results are available. Our method is able to achieve similar or even slightly higher rates of correctly detected regions as the UB segmenter [JB97], while clearly outperforming the BJ segmenter [BJ88]. The results are encouraging in the sense that our method can easily compete with these native range image segmenters without any further adaptations to this special case.

6. Conclusion

After having outlined the potential in the well-known RANSAC paradigm we have developed several fundamental

extensions and derived new assessments for the probabilities involved therein. We were able to present a randomized shape detection algorithm that is not dominated by the imponderability of chance, but is extremely robust and fast, and can be applied to a large variety of data from different sources and of different quality. Although our algorithm, in contrast to previous methods in computer graphics, does not find shape proxies for every part of the surface, we stress that this is not a requirement in many applications. The speed of the method, the quality of the results and its few requirements on the data render our algorithm a practical choice for shape detection in many areas.

Besides obvious extensions, such as the recognition of additional shapes and the inclusion of a more sophisticated model selection stage (possibly based on the MDL principle), in future work we plan to further explore the potential of the hybrid shape representations obtained with the algorithm. Currently, we are investigating the use of the shape information for compression as well as registration and already have achieved very promising results. Further directions of future research include surface repairing, surface reconstruction and object recognition. Also the benefit of such higher level information in interactive applications such as modeling, segmentation and classification can be studied.

References

[BF81] BOLLES R. C., FISCHLER M. A.: A ransac-based approach to model fitting and its application to finding cylinders in range data. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, (1981), 637–643.

[BGV*02] BENKO P., GÉZA K., VÁRADY T., ANDOR L., MARTIN R.: Constrained fitting in reverse engineering. *Comput. Aided Geom. Des. 19*, 3 (2002), 173–205.

[BJ88] BESL P. J., JAIN R. C.: Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Anal. Mach. Intell. 10*, 2 (1988), 167–192.

[BMV01] BENKO P., MARTIN R. R., VÁRADY T.: Algorithms for reverse engineering boundary representation models. *Comput.-Aided Des. 33*, 11 (2001), 839–851.

[CAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. *ACM Trans. Graph. 23*, 3 (2004), 905–914.

[CG01] CHAPERON T., GOULETTE F.: Extracting cylinders in full 3-d data using a random sampling method and the gaussian image. In *VMV01*, (2001), 35–42.

[DDS*03] DÉCORET X., DURAND F., SILLION F., DORSEY J.: Billboard clouds for extreme model simplification. In *Proceedings of the ACM Siggraph*, ACM Press, 2003.

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (1981), 381–395.

[FEF97] FITZGIBBON A. W., EGGERT D. W., FISHER R. B.: High-level CAD model acquisition from range images. *Comput.-Aided Des. 29*, 4 (1997), 321–330.

[GBS03] GOTARDO P. F. U., BELLON O. R. P., SILVA L.: Range image segmentation by surface extraction using an improved robust estimator. *cvpr 02*, 33 (2003).

[GG04] GELFAND N., GUIBAS L. J.: Shape segmentation using local slippage analysis. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, ACM Press, New York, 2004, 214–223.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MC DONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, 1992, 71–78.

[HOP*05] HOFER M., ODEHNAL B., POTTMANN H., STEINER T., WALLNER J.: 3d shape recognition and reconstruction based on line element geometry. In *Tenth IEEE International Conference on Computer Vision 2*, IEEE Computer Society, (2005), 1532–1538.

[Hou62] HOUGH P.: Method and means for recognizing complex patterns. In *US Patent*, 1962.

[IK87] ILLINGWORTH J., KITTLER J.: The adaptive hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence 9*, 5 (September 1987), 690–698.

[IK88] ILLINGWORTH J., KITTLER J.: A survey of the hough transform. *Comput. Vision Graph. Image Process. 44*, 1 (1988), 87–116.

[JB97] JIANG X., BUNKE H.: Range image segmentation: Adaptive grouping of edges into regions. In *ACCV '98: Proceedings of the Third Asian Conference on Computer Vision-Volume II*, Springer-Verlag, London, 1997, 299–306.

[LGB95] LEONARDIS A., GUPTA A., BAJCSY R.: Segmentation of range images as the search for geometric parametric models. *Int. J. Comput. Vision 14*, 3 (1995), 253–277.

[LJS97] LEONARDIS A., JAKLIČ A., SOLINA F.: Superquadrics for segmenting and modeling range data. *IEEE Trans. Pattern Anal. Mach. Intell. 19*, 11 (1997), 1289–1295.

[MC02] MATAS J., CHUM O.: Randomized RANSAC with t(d, d) test. In *Proceedings of the British Machine Vision Conference 2002 (BMVC)*, 2002.

[MLM01] MARSHALL D., LUKACS G., MARTIN R.: Robust segmentation of primitives from range data in the presence of geometric degeneracy. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 3 (2001), 304–314.

[MLT00] MEDIONI G., LEE M. S., TANG C. K.: *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

[MTN*02] MYATT D., TORR P., NASUTO S., BISHOP J., CRADDOCK R.: Napsac: High noise, high dimensional robust estimation —it's in the bag. In *BMVC*, p. Computer Vision Tools, 2002.

[Nis05] NISTR D.: Preemptive RANSAC for live structure and motion estimation. *Mach. Vision Appl. 16*, 5 (2005), 321–329.

[PBJ*98] POWELL M. W., BOWYER K. W., JIANG X., BUNKE H.: Comparing curved-surface range image segmenters. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, 1998, 286.

[RL93] ROTH G., LEVINE M. D.: Extracting geometric primitives. *CVGIP: Image Underst. 58*, 1 (1993), 1–22.

[SB95] SAPIDIS N. S., BESL P. J.: Direct construction of polynomial surfaces from dense range images through region growing. *ACM Trans. Graph. 14*, 2 (1995), 171–200.

[SC97] SHIRLEY P., CHIU K.: A low distortion map between disk and square. *J. Graph. Tools 2*, 3 (1997), 45–52.

[Sha98] SHAKARJI C. M.: Least-squares fitting algorithms of the NIST algorithm testing system. *J. Res. Nat. Inst. Stand. Techn. 103*, 6 (1998), 633–641.

[TZ00] TORR P., ZISSERMAN A.: MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding 78*, (2000), 138–156.

[TZ98]   TORR P., ZISSERMAN A.: Robust computation and parametrization of multiple view relations. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, IEEE Computer Society, Washington, DC, 1998, 727.

[VGS*04]   VOSSELMAN G., GORTE B., SITHOLE G., RABBANI T.: Recognising structure in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 46* 8/W2 (2004), 33–38.

[VMC97]   VÁRADY T., MARTIN R. R., COX J.: Reverse engineering of geometric models—an introduction. *Comput.-Aided Des. 29*, 4 (1997), 255–268.

[WGK05]   WAHL R., GUTHE M., KLEIN R.: Identifying planes in point-clouds for efficient hybrid rendering. In *The 13th Pacific Conference on Computer Graphics and Applications*, October 2005.

[WK05]   WU J., KOBBELT L.: Structure recovery via hybrid variational surface approximation. In *Computer Graphics Forum 24* (2005), pp. 277–284.

[XO93]   XU L., OJA E.: Randomized hough transform (RHT): basic mechanisms, algorithms, and computational complexities. *CVGIP: Image Underst. 57*, 2 (1993), 131–154.