A Minimalist Approach to Type-Agnostic Detection of Quadrics in Point Clouds

Tolga Birdal[1,2]   Benjamin Busam[1,3]   Nassir Navab[1,4]   Slobodan Ilic[1,2]   Peter Sturm[5]

[1] Technische Universität München, Germany   [2] Siemens AG, Munich, Germany

[3] Framos AG, Munich, Germany   [4] Johns Hopkins University, US   [5] INRIA, France

Abstract

This paper proposes a segmentation-free, automatic and efficient procedure to detect general geometric quadric forms in point clouds, where clutter and occlusions are inevitable. Our everyday world is dominated by man-made objects which are designed using 3D primitives (such as planes, cones, spheres, cylinders, etc.). These objects are also omnipresent in industrial environments. This gives rise to the possibility of abstracting 3D scenes through primitives, thereby positions these geometric forms as an integral part of perception and high level 3D scene understanding.

As opposed to state-of-the-art, where a tailored algorithm treats each primitive type separately, we propose to encapsulate all types in a single robust detection procedure. At the center of our approach lies a closed form 3D quadric fit, operating in both primal & dual spaces and requiring as low as 4 oriented-points. Around this fit, we design a novel, local null-space voting strategy to reduce the 4-point case to 3. Voting is coupled with the famous RANSAC and makes our algorithm orders of magnitude faster than its conventional counterparts. This is the first method capable of performing a generic cross-type multi-object primitive detection in difficult scenes. Results on synthetic and real datasets support the validity of our method.

1. Introduction

Quadrics, or quadratic surfaces, are second order implicit representations, including geometric primitives such as planes, spheres, cylinders, ellipsoids, cones and more. Due to their ability to cover a wide variety of shapes, many industrial parts and man-made objects are manufactured using quadric solids after designed with specific CAD software. As a result, our environment consists of these mathematical constructs. This prominent exposure made understanding and reverse engineering of quadrics, where a scene is parsed into primitives, a primary concern in computer vision already in the 80s [36, 7]. While a large part of these works address the problem of recovering quadrics from 2D images [12], in the 3D domain, the research streams still
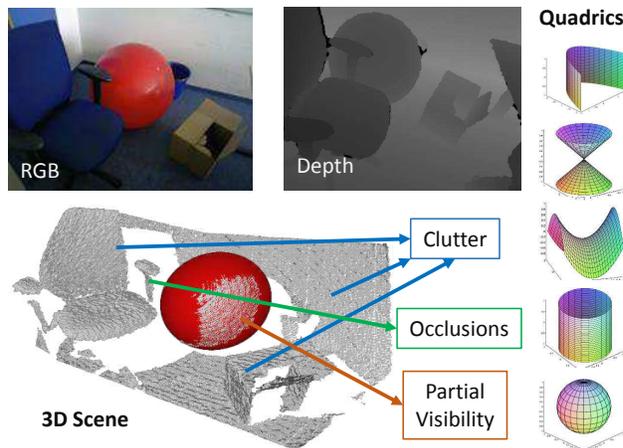

Figure 1. Our algorithm quickly detects quadric shapes in point clouds under occlusions, partial visibility and clutter. Note that, the algorithm has no clue whether a sphere exists or not and is able to detect the plausible primitive using only 3D geometry.

seem to be divided into two branches: *Quadric Fitting* and *Primitive Detection*. The former, *Quadric Fitting*, concerns the fitting of a general quadric to a clutter-free, potentially noisy scene [51, 49, 9]. The latter, *Primitive Detection*, tackles a different problem: the reconstruction of type-specific primitives from cluttered scenes [1, 2, 25]. Yet, to this date, the task of automatic and generic quadric detection within clutter or occlusions without necessitating auxiliary preprocessing steps such as segmentation or manual intervention [39], remain to be unexplored and unsolved. This is partially because unlike rigid 6-DoF free-form detection & pose estimation, quadrics have 9 degrees of freedom, and can deform into various forms. Naturally, this problem is a harder one. Although deep neural networks can learn to segment scenes [37, 23], or show great potential in learning the fitting function and feature extraction [13], jointly solving the detection and surface fitting, to the best of our knowledge, is an open challenge. Moreover, for the problem at hand, data labeling is notoriously exhaustive.

One trivial way to attack our problem would be to use an off-the-shelf quadric fitting method in a RANSAC [18] pipeline. Yet, there is a caveat in doing this: RANSAC re-

quires repeated trials of a minimal-case model fitting. Even though degenerate or special quadrics such as cylinders, planes or spheres can be described by three or two or single oriented 3D point(s) [16], it is not the case for a more complex quadric such as an ellipsoid, paraboloid or hyperboloid and the state of the art requires 9 [51, 49, 9] to 12 [4] points to perform a generic quadric fit. Such large DoF necessity increases the complexity of RANSAC-like approaches significantly. Hence, this seemingly old problem - discovering equations of quadrics in arbitrary and general 3D scenes - awaits an efficient and robust clutter treatment [42, 21].

In this paper, we fill these gaps by showing first a mixed primal/dual-space quadric fit, requiring as low as 4 oriented-points (4 points and associated normals). We analyze this fit and around it, devise a local Hough transform method to reduce the 4-point case to 3-points, establishing the minimalist fit known so far. These developments pave the way to an efficient RANSAC loop. We slightly modify the RANSAC pipeline using a pre-clustering of feasible hypotheses. We provide distance and score metrics on quadric spaces to perform this robust fit. In a nutshell, our contributions are:

1. A novel closed form quadric fitting capable of using as low as 4 oriented points as supported theoretically,
2. An effective local voting strategy, reducing the 4-point requirement to 3-points,
3. A fast & robust scheme to compute voting parameters,
4. A 3-point RANSAC algorithm with an efficient score function to verify the hypotheses,
5. To the best of our knowledge, the first robust pipeline to perform generic type-agnostic multi-object quadric detection in the presence of clutter and occlusion.

Quantitative and qualitative experiments on synthetic and real datasets demonstrate the robustness of our fitting as well as the computational gains. We will make our implementation openly available.

2. Prior Art

Quadrics Quadrics appear in various domains of vision, graphics and robotics. They are found to be one of the best local surface approximators in estimating differential properties [42]. Thus, point cloud normals and curvatures are oftentimes estimated via local quadrics [59, 15, 6, 14]. Other use cases include mesh representation [27], mesh and point-cloud segmentation [56, 33], projective geometry & reconstruction [24, 12], minimal problem solving [31], grasp motion planning [41, 54] and feature extraction [58].

Primitive Detection Discovering primitives in point clouds has kept the vision researchers busy for a long period of time. This category treats the primitive shapes independently [25], giving rise to plane, sphere, cone, cylinder and etc. specific fitting algorithms. Planes, as the simplest

forms, are the primary targets of Hough-family [10]. Yet, to find a larger variety of primitives, RANSAC comes in handy as shown in the prosperous Globfit [32]: a relational local to global RANSAC algorithm. Schnabel et. al. [47] and Tran et. al. [52] also focus on reliable estimation via RANSAC. Qui et.al. extract pipe runs using cylinder fitting [45], and the local Hough transform of Drost and Ilic [16] efficiently detects spheres, cylinders and planes from point clouds. Lopez et. al. [34] devise a robust ellipsoid fitting based on iterative non-linear optimization while Sveier et. al. [48] suggest a conformal geometric algebra to spot planes, cylinders and spheres. Andrews [3] deals with paraboloids and hyperboloids in CAD models.
Methods and algorithms in this category are quite successful in shape detection, yet they handle the primitives separately. This prevents automatic type detection, or generalized modeling of surfaces such as paraboloids or hyperboloids along with cylinders and spheres.

Quadric Fitting Since the 90s, following the pioneering work of Gabriel Taubin [51], quadric fitting is cast as a constrained optimization problem, where the solution is obtained from a Generalized Eigenvalue decomposition of a scatter matrix. This work has then been enhanced by 3L fitting [9], proposing a local, explicit ribbon surface composed of three-level-sets of constant Euclidean distance to the object boundary. This fit implicitly uses the local surface information. Later on Tasdizen [49] improved the local surface properties by incorporating the surface normals to regularize the scatter matrix, which allows for a good and stable fit. Recently, Beale et. al. [4] introduced the use of a Bayesian prior for the fitting pipeline. All of these methods use at least 9 points. Moreover, they only use surface normals as regularizers and not as additional constraints or are unable to deal with outliers in data. There are a few other studies [30, 1], which improve these standard methods, but many of them either involve non-linear optimization [57] or they share the common drawback of requiring 9 independent constraints and no outlier treatment.

Quadric Detection Recovering general quadratic forms from cluttered and occluded scenes is a rather unexplored area. A promising direction was to represent quadrics with spline surfaces [39], but such approaches have to tackle the increased number of control points, i.e. 8 for spheres, 12 for general quadrics [43, 44]. Segmentation is one way to overcome such difficulties [35, 39], but introduces undesired errors especially under occlusions. Other works exploit genetic algorithms [26] but have the obvious drawback of inefficiency. QDEGSAC [22] proposed a 6-point hierarchical RANSAC, but the paper misses out an evaluation or method description for a quadric fit. Petitjean [42] stressed out the necessity of outlier aware quadric fitting however only ends up suggesting M-estimators for future research.

3. Preliminaries

Definition 1
*A **quadric** in 3D Euclidean space is a hypersurface defined by the zero set of a polynomial of degree two:*

$$f(x, y, z) = Ax^2 + By^2 + Cz^2 + 2Dxy + 2Exz \quad (1)$$
$$+ 2Fyz + 2Gx + 2Hy + 2Iz + J = 0.$$

or similarly in the vector notation $\mathbf{v}^T \mathbf{q} = 0$ with:

$$\mathbf{q} = \begin{bmatrix} A & B & C & D & E & F & G & H & I & J \end{bmatrix}^T \quad (2)$$
$$\mathbf{v} = \begin{bmatrix} x^2 & y^2 & z^2 & 2xy & 2xz & 2yz & 2x & 2y & 2z & 1 \end{bmatrix}^T$$

Using homogeneous coordinates, quadrics can be analyzed uniformly. The point $\mathbf{x} = (x, y, z) \in \mathbb{R}^3$ lies on the quadric, if the projective algebraic equation over $\mathbb{RP}^3$ with $d_q(\mathbf{x}) := [\mathbf{x}^T 1]\mathbf{Q}[\mathbf{x}^T 1]^T$ holds true, where the matrix $\mathbf{Q} \in \mathbb{R}^{4 \times 4}$ is defined by re-arranging the coefficients:

$$\mathbf{Q} = \begin{bmatrix} A & D & E & G \\ D & B & F & H \\ E & F & C & I \\ G & H & I & J \end{bmatrix}, \quad \nabla\mathbf{Q} = 2 \begin{bmatrix} A & D & E & G \\ D & B & F & H \\ E & F & C & I \end{bmatrix}. \quad (3)$$

$d_q(\mathbf{x})$ can be viewed as an algebraic distance function. Similar to the quadric equation, the gradient at a given point can be written as $\nabla\mathbf{Q}(\mathbf{x}) := \nabla\mathbf{Q}[\mathbf{x}^T 1]^T$. Quadrics are general implicit surfaces capable of representing cylinders, ellipsoids, cones, planes, hyperboloids, paraboloids and potentially the shapes interpolating any two of those. All together there are 17 sub-types [55]. Once $\mathbf{Q}$ is given, this type can be determined from an eigenvalue analysis of $\mathbf{Q}$ and its subspaces. Note that quadrics have constant second order derivatives and are practically smooth.

Definition 2
*A **dual quadric** $\mathbf{Q}^* \sim \mathbf{Q}^{-1}$ is the locus of all planes $\{\mathbf{\Pi}_i\}$ satisfying $\mathbf{\Pi}_i^T \mathbf{Q}^{-1} \mathbf{\Pi}_i = 0$ and is formed by the Legendre transform, sending points to tangent planes as covectors.*

Knowing a point lies on the surface gives 1 constraint, and if, in addition, one knows the tangent plane at that point, then one gets 2 more constraints. Such view will lead to a mixed-space fitting, reducing the minimal point necessity.

Definition 3
*A **basis** $\mathbf{b}$ is a subset composed of a fixed number (b) of scene points and hypothesized to lie on the sought surface.*

4. Quadric Detection in Point Clouds

Our algorithm operates by repeatedly selecting bases. Once a basis $\mathbf{b}$ is fixed, an under-determined quadric fit parameterizes the solution and a local accumulator space, tied to $\mathbf{b}$, is formed. All other points in the scene are then combined with this basis to vote for the potential generic primitive. To discover the optimal basis, we perform RANSAC,

iteratively hypothesizing solution candidates. Subsequent to such joint RANSAC and voting, a hypotheses custering and ranking takes place, allowing for multiple quadric detection without re-executions of the whole algorithm.

A new perspective to quadric fitting Direct solutions for quadric fitting rely either solely on point sets [51], or use normals as regularizers [49]. This is due to vector-vector alignment being a non-linear constraint caused by the normalization $\nabla\mathbf{Q}(\mathbf{x}) / \|\nabla\mathbf{Q}(\mathbf{x})\|$, i.e. it is hard to know the norm in advance. Similar to $\nabla 1$-fitting [50, 49], our idea is to align the gradient vector of the quadric $\nabla\mathbf{Q}(\mathbf{x}_i)$ with surface normal $\mathbf{n}_i \in \mathbb{R}^3$, this time using a linear gradient constraint to increase the DoFs rather than regularizing the solution. To do so, we introduce per normal homogeneous scale α_i among the unknowns and write: $\nabla\mathbf{v}_i^T \mathbf{q} = \alpha_i \mathbf{n}_i$. Stacking it up for all N points $\mathbf{x}_i$ and normals $\mathbf{n}_i$ leads to:

$$\mathbf{A}' \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \cdots \mathbf{v}_i \cdots \end{bmatrix}^T & \begin{bmatrix} \mathbf{0} \end{bmatrix} \\ \begin{bmatrix} \cdots \nabla\mathbf{v}_i \cdots \end{bmatrix}^T & \begin{bmatrix} diag(-\mathbf{n}') \end{bmatrix} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\alpha} \end{bmatrix} = \mathbf{0}$$
$$(4)$$

where $\mathbf{A}'$ is $4N \times (N + 10)$, $\boldsymbol{\alpha}$ are the unknown homogeneous scales and $\mathbf{n}' = [\cdots \mathbf{n}_i^T \cdots]^T$. The solution defines a unique fit in the null space of $\mathbf{A}'$, which gets to full rank for ≥ 4 oriented points, the minimalist case known up to now.

Theorem 1
Although 3 oriented points provide 9 constraints, quadric fitting, as formulated, possesses a trivial solution in addition to the true one.

Proof. Let us call **data-plane**, the plane spanned by the coordinates of 3 data points. Any rank-1 quadric consists of a single plane $\mathbf{\Pi}$ and can be written as $\mathbf{Q} = \mathbf{\Pi} \mathbf{\Pi}^T$. Hence, for any point $\mathbf{u}$ on the data plane and thus on the quadric, we have $\mathbf{Q}\mathbf{u} = \mathbf{0}$ and $\nabla\mathbf{v}_i^T \mathbf{q} = \mathbf{0}$. This amounts to the choice $\boldsymbol{\alpha} = \mathbf{0}$. Thus the trivial solution is identified as the rank-1 quadric consisting of the data-plane and zero scale factors. Hence, the estimation problem admits at least a one-dimensional linear family of solutions, spanned by the true quadric and the rank-1 quadric of the data-plane. □

Indeed, for a non-degenerate surface, following relations hold: $N = 1 \Rightarrow \text{rk}(\mathbf{A}') = 4$, $N = 2 \Rightarrow \text{rk}(\mathbf{A}') = 7$, $N = 3 \Rightarrow \text{rk}(\mathbf{A}') = 9$ and $N > 3 \Rightarrow \text{rk}(\mathbf{A}') = 10$. Showing this and that adding further point constraints give diminishing returns is also relatively easy to see by Gaussian Elimination on the matrix, as equations are inter-dependent.

Despite sparsity, the unknowns in this system scale linearly with N, leading to a large system to solve. In practice, we approximate them with a single common scale $\bar{\alpha}$, similar to adding a **soft-regularizer** that tries to force $\alpha_i \leftarrow \bar{\alpha}$ [49]. Doing so rescues us from solving the sensitive homogeneous system [58], and lets us re-write the system in a com-

pact form $\mathbf{A}\mathbf{q} = \mathbf{n}$:

$$\mathbf{A} = \begin{bmatrix} x_1^2 & y_1^2 & z_1^2 & 2x_1y_1 & 2x_1z_1 & 2y_1z_1 & 2x_1 & 2y_1 & 2z_1 & 1 \\ x_2^2 & y_2^2 & z_2^2 & 2x_2y_2 & 2x_2z_2 & 2y_2z_2 & 2x_2 & 2y_2 & 2z_2 & 1 \\ & & & & \vdots & & & & & \\ 2x_1 & 0 & 0 & 2y_1 & 2z_1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2y_1 & 0 & 2x_1 & 0 & 2z_1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2z_1 & 0 & 2x_1 & 2y_1 & 0 & 0 & 2 & 0 \\ 2x_2 & 0 & 0 & 2y_2 & 2z_2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2y_2 & 0 & 2x_2 & 0 & 2z_2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2z_2 & 0 & 2x_2 & 2y_2 & 0 & 0 & 2 & 0 \\ & & & & \vdots & & & & & \end{bmatrix}$$

$$\mathbf{n} = \begin{bmatrix} 0 & 0 & \dots & n_x^1 & n_y^1 & n_z^1 & n_x^2 & n_y^2 & n_z^2 & \dots \end{bmatrix}^T$$

$$\mathbf{q} = \begin{bmatrix} A & B & C & D & E & F & G & H & I & J \end{bmatrix}^T \quad (5)$$

The matrix $\mathbf{A}$ is only $4N \times 10$ having identical rank properties as Eq. 4. To balance the contribution of normal induced constraints we, introduce a scalar weight w. Note that *true* quadric fitting is intrinsically of non-linear nature and hence formulating such a linear form is only possible with algebraic metrics, always being subject to certain bias [20]. We pose it a future work to study the bias introduced by our fit.

Fig. 2(a) shows the minimum number of constraints required for a full-rank fit in primal and dual spaces. Besides, to obtain a type-specific fit, a minor redesign of $\mathbf{A}$ is sufficient. Fig. 2(b) depicts the minimal cases for exemplary shapes and the voting space dimension as we describe hereafter. In the following, we elaborate on a more efficient way to calculate a solution to Eq. 5 rather than using a naive RANSAC on 4 tuples by analyzing its solution space. The rest of this section can be thought of as a generic method applicable to any problem formulated as a linear system.

System 5 describes an outlier-free closed form fit. To factor in clutter, a direct RANSAC on 9DoF quadric appears to be trivial. Yet, it has two drawbacks: 1) Evaluating the residuals many times is challenging, as it involves a scene-to-quadric overlap calculation in a geometric meaningful way. 2) Even with the proposed fitting, selecting random 4-tuples from the scene might be slow in practice. An alternative to that is Hough voting. However, $\mathbf{q}$ has 9 DoFs and is not discretization friendly. The complexity and size of this parameter space makes it hard to construct a voting space directly on $\mathbf{q}$. Instead, we now devise a scheme entailing a smaller basis cardinality, with local search.

Parameterizing the solution space Let $\mathbf{q}$ be a solution to Eq. 5. Then, $\mathbf{q}$ can be expressed by a linear combination of particular solution $\mathbf{p}$ and homogeneous solutions $\boldsymbol{\mu}_i$ as:

$$\mathbf{q} = \mathbf{p} + \sum_i^D \lambda_i \boldsymbol{\mu}_i \quad (6)$$

$$= \mathbf{p} + \begin{bmatrix} \boldsymbol{\mu}_1 & \boldsymbol{\mu}_2 & \cdots \end{bmatrix} \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots \end{bmatrix}^T = \mathbf{p} + \mathbf{N}_A \boldsymbol{\lambda}.$$

	# Pri.	# Dual		# Pri.	# Dual	|b|	VS
PD-0	9	0	Plane	1	1	1	0
PD-1	7	1	2-Planes	2	2	2	0
PD-2	5	2	Sphere	2	1	1	1
PD-3	4	3	Spheroid	2	2	2	3
	(a)				(b)		

Figure 2. **(a)** Number of constraints for a minimal fit in Primal(P) or Dual(D) spaces. PD-i refers to i[th] combination. **(b)** Number of minimal constraints and voting space size for various quadrics.

The dimensionality of the null space $\mathbf{N}_A$ (D) depends on the rank of $\mathbf{A}$, which is directly influenced by the number of points used: $D = 10 - rk(\mathbf{A})$. The exact solution could always be computed by including more points from the scene and validating them, i.e., by a local search. For that reason, the fitting can be split up into distinct parts: First a parametric solution as in Eq. 6 is computed using a subset of points (basis) $\mathbf{b} = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$ lying on a quadric. Next, the coefficients $\boldsymbol{\lambda}$, and thus the solution, can be obtained by searching for other point(s) $(\mathbf{x}_{m+1}, ..., \mathbf{x}_{m+k})$ which lie on the same surface as $\mathbf{b}$.

Theorem 2
If two point sets $\mathbf{b} = (\mathbf{x}_1, ..., \mathbf{x}_m)$ and $\mathbf{X} = (\mathbf{x}_{m+1}, ..., \mathbf{x}_{m+k})$ lie on the same quadric with parameters $\mathbf{q}$, then the coefficients $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots \end{bmatrix}^T$ of the solution space (6) are given by the solution of the system:

$$(\mathbf{A}_k \mathbf{N}_A)\boldsymbol{\lambda} = \mathbf{n}_k - \mathbf{A}_k \mathbf{p} \quad (7)$$

where $\mathbf{A}_k$, $\mathbf{n}_k$ are the linear constraints of the latter set $\mathbf{X}$ in form of (5), $\mathbf{p}$ is a particular solution and $\mathbf{N}_A$ is a $10 \times D$ stacked null-space basis as in (6), obtained from $\mathbf{b}$.

Proof. Let $\mathbf{q}$ be a quadric solution for the point set $(\mathbf{x}_1, ..., \mathbf{x}_m)$ and let $(\mathbf{A}_k, \mathbf{n}_k)$ represent the $4k$ quadric constraints for the k points $\mathbf{X} = (\mathbf{x}_{m+1}, ..., \mathbf{x}_{m+k})$ in form of (5) with the same parameters $\mathbf{q}$. As $\mathbf{x}_i \in \mathbf{X}$ by definition lies on the same quadric $\mathbf{q}$, it also satisfies $\mathbf{A}_k \mathbf{q} = \mathbf{n}_k$. Inserting Eq. (6) into this, we get:

$$\mathbf{A}_k(\mathbf{p} + \mathbf{N}_A \boldsymbol{\lambda}) = \mathbf{n}_k \quad (8)$$

$$(\mathbf{A}_k \mathbf{N}_A)\boldsymbol{\lambda} = \mathbf{n}_k - \mathbf{A}_k \mathbf{p}. \quad (9)$$

Solving Eq. (9) for $\boldsymbol{\lambda}$ requires a multiplication of a $4k \times 10$ matrix with a $10 \times m$ one and ultimately solving a system of $4k$ equations in m unknowns. This is much more efficient for $k < m$ than re-solving the system (5) and resembles updating the solution online for a stream of points.

Local voting for quadric detection Given a fixed basis composed of b points $(b > 0)$ as in Fig. 3, a parametric solution can be described. The actual solution can then be found quickly as explained above by incorporating new
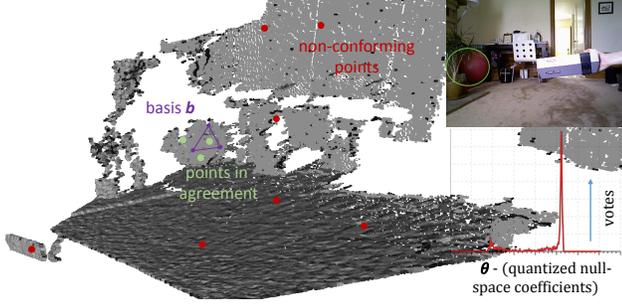
Figure 3. Once a basis is randomly hypothesized, we look for the points on the same surface by casting votes on the null-space. The sought pilates ball (likely quadric) is marked on the image and below that lies the corresponding filled accumulator by KDE [46].

points lying on the same quadric. Thus, the problem of quadric detection is de-coupled into 1) finding a proper basis and 2) searching for compatible scene points. In this part, we assume the basis is correctly found and explain the search by voting. For a fixed basis $\mathbf{b}_i$ on a quadric, we form the null-space decomposition of the under-determined system $\mathbf{A}_i\mathbf{q} = \mathbf{n}_i$. We then sample further points from the scene and compute the required coefficients $\boldsymbol{\lambda}$. Thanks to Thm. 2, this can be done efficiently. Sample points lying on the same quadric as the basis (inliers) generate the same $\boldsymbol{\lambda}$ whereas outliers will produce different values. Therefore we propose to construct a voting space on $\boldsymbol{\lambda}$ and cast votes to maximize the local consensus (Fig. 3). The size of the voting space is a design choice and depends on the size of $\mathbf{b}_i$ vs. the DoFs desired to be recovered (see Fig. 2(b)). We find from Fig. 2(a) that using a 3-point basis is advantageous for a generic quadric fit due to 1D search space.

Efficient computation of voting parameters For a 3-point basis, adding a fourth sample point $\mathbf{x}_4$ completes $rk(\mathbf{A}) = 10$ and a unique solution can be computed. Yet, as we will select multiple $\mathbf{x}_4$ candidates per basis, hypothesized in a RANSAC loop, an efficient scheme is required, i.e. it is undesirable to re-solve Eq. 5 for each incoming $\mathbf{x}_4$ tied to the basis. It turns out that once again, the solution can be obtained directly from Eq. (6):

Theorem 3

If the null-space is one dimensional (with only 1 unknown) it holds $\mathbf{N}_A\boldsymbol{\lambda} = \lambda_1\boldsymbol{\mu}_1$ and the computation in Thm. 2 reduces to the explicit form:

$$\lambda_1 = (\mathbf{A}_1\mathbf{N}_A)^T/\|\mathbf{A}_1\mathbf{N}_A\|^2 \cdot (\mathbf{n}_1 - \mathbf{A}_1\mathbf{p}) \qquad (10)$$

Proof. Let us re-write Eq. 9 in terms of the null space vectors: $\lambda_1(\mathbf{A}_1\boldsymbol{\mu}_1) = \mathbf{n}_1 - \mathbf{A}_1\mathbf{p}$. A solution λ_1 can be obtained via Moore-Penrose pseudoinverse [38] as $\lambda_1 = (\mathbf{A}_1\boldsymbol{\mu}_1)^+(\mathbf{n}_1 - \mathbf{A}_1\mathbf{p})$. Because for 1-dimensional null spaces $\mathbf{A}_1\boldsymbol{\mu}_1$ is a vector, $^+$ operator is defined as: $\boldsymbol{v}^+ = \boldsymbol{v}^T/(\boldsymbol{v}^T\boldsymbol{v})$. Substituting this in Eq. 7 gives Eq. 10. □

Thm. 3 enables a very quick computation of the parameter hypothesis in case of 1 point. A MATLAB implementation takes ca. $30\mu s$ per λ. Besides, for a 3-point method, inclusion of only a single equation is sufficient, letting the normal of the 4th point remain unused and amenable for verification of fit. We only accept to vote a candidate if the gradient of the fit quadric agrees with the normal at 4th point: $(\nabla\mathbf{Q}(\mathbf{x}_4)/\|\nabla\mathbf{Q}(\mathbf{x}_4)\|) \cdot \mathbf{n}(\mathbf{x}_4) > \tau_n$.

Quantizing $\boldsymbol{\lambda}$ To vote on $\boldsymbol{\lambda}$, we need a linear, bounded and discrete space. Yet, $\boldsymbol{\lambda}$ is not quantization-friendly, as it is unbounded and has a non-linear effect on the quadric shape (see Fig. 4(a)). Thus, we seek to find a geometrically meaningful transformation to a bounded and well behaving space so that quantization would lead to little bias and artifacts. From a geometric perspective, each column of $\mathbf{N}_A$ in Eq. 6 is subject to the same coefficient λ, representing the slope of a high dimensional line in the solution space. Thus, it could as well be viewed as a rotation. For 1D null-space, we set: $\theta = \text{atan2}\big((y_2 - y_1), (x_2 - x_1)\big)$, where $[x_1, y_1, \cdots]^T = \mathbf{p}$ and $[x_2, y_2 \cdots]^T$ is obtained by moving in the direction $\mathbf{N}_A$ from the particular solution $\mathbf{p}$ by an offset λ.[1] This new angle θ is bounded and thus easy to vote for. As the null-space dimension grows, λ starts to represent hyperplanes, still preserving the geometric meaning, i.e. for $d > 1$, different $\boldsymbol{\theta} = \{\theta_i\}$ can be found.

Hypotheses Aggregation Up to this point, we have described how to find plausible quadrics given local triplet bases. As mentioned, to discover the basis lying on the surface, we employ RANSAC [18], where each triplet might generate a hypothesis to be verified. Many of those will be similar as well as dissimilar. Thus, the final stage of the algorithm aggregates the potential hypotheses to reduce the number of candidate surfaces and to increase the per quadric confidence. Not to sacrifice further speed, we run an agglomerative clustering similar to [6] in a coarse to fine manner: First an algebraic but fast distance measure helps to cluster the obvious hypotheses:

$$d_{close}(\mathbf{Q}_1, \mathbf{Q}_2) := \mathbb{1}(\|\mathbf{q}_1 - \mathbf{q}_2\|_1 < \tau) \cdot \|\mathbf{Q}_1\mathbf{Q}_2^+ - \mathbf{I}\|_F \tag{11}$$
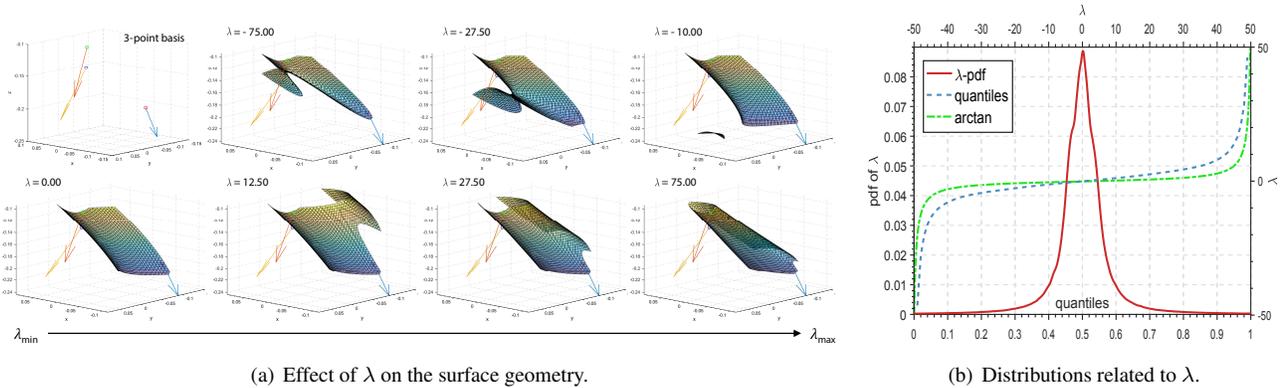
where $\mathbf{I} \in \mathbb{R}^{4\times4}$ is the identity matrix and $\mathbb{1} : \mathbb{R} \to \{0, 1\}$ the indicator function. Second, a pseudo-geometric one is executed only on a subset of the surviving quadrics:

$$d_{far}(\mathbf{Q}_1, \mathbf{Q}_2) := \tag{12}$$

$$1 - \frac{1}{K}\sum_{i=1}^{K} \mathbb{1}\left(|\mathbf{x}_i^T\mathbf{Q}_1\mathbf{x}_i| < \tau\right) \cdot \mathbb{1}\left(|\mathbf{x}_i^T\mathbf{Q}_2\mathbf{x}_i| < \tau\right) \cdot$$

$$\mathbb{1}\left(1 - \mathbf{n}_i \cdot \nabla\mathbf{Q}_1(\mathbf{x}_i) < \tau_n\right) \cdot \mathbb{1}\left(1 - \mathbf{n}_i \cdot \nabla\mathbf{Q}_2(\mathbf{x}_i) < \tau_n\right)$$

[1]Simple $tan^{-1}(\lambda)$ could work but would be more limited in the range.

(a) Effect of λ on the surface geometry.

(b) Distributions related to λ.

Figure 4. Characteristics of $\boldsymbol{\lambda}$. **(a)** We compute null-space decomposition for a fixed basis and vary λ from -75 to 75 to generate different solutions $\mathbf{q}$ along the line in the solution space. The plot presents the transition of the surface controlled by λ. **(b)** Statistical distribution of the solution-space coefficient and our quantization function: PDF (red curve) and inverse CDF (dashed blue-curve) of λ over collected data, and tan^{-1} function (green-line). Note that our quantization function also well aligns with practice. Smaller $|\lambda|$ results in finer binning.

where the scene is composed of K points $\{\mathbf{x}_i\}$. Each quadric is associated a score proportional to the number of points with compatible normals, found to lie on it. This score is used to re-rank the detections.

5. Experimental Evaluation and Discussions

Implementation details Planes are singular cases and occupy large spaces of 3D scenes. A preprocessing removes them using our type specific plane fit. Normalizing the points to a unit ball is also necessary to increase the numerical stability [28]. Next, we downsample the scene using a spatial voxel-grid [8] enforcing a minimum distance of $\tau_s \cdot diam(\mathbf{X})$ between the samples ($0.025 \leq \tau_s \leq 0.05$) and compute the normals [29] on this subset. What follows is an iterative selection of triplets from this reduced set. At each iteration, once $\mathbf{x}_1$ is fixed, we query the points which are likely to be combined with $\mathbf{x}_1$ to form the 3-point basis $\mathbf{b}$. The rest of the points are then randomly selected from this subset. Triplets which are found to be collinear or which do not result in a rank-9 system are directly removed. We also hash the triplets such that any basis is seen only once.

Synthetic tests of fitting To asses the accuracy of the proposed fitting, we generate a synthetic test set of multitudes of random quadrics and compare our method with the fitting procedures of Taubin [51], Tasdizen [49], Andrews [3], and Beale [4] as well as our full, least squares fit. Gaussian noise with $\sigma = [0\% - 5\%]$ relative to the size s of the quadric is added to the vertices before the methods are run. At each noise level, 10 random quadrics are tested. We report the average point-to-mesh distance and the angle deviation as well as the runtime performances in Fig. 5. For the constrained fitting method [3] we pre-specified the type, which might not be possible in a real application. We perform not

single but 20 fits per set. Although, our fit is designed to use a minimal number of points, it also proves to be robust when more points are added and is among the top fitters for the distance & angle errors. This also demonstrates that gradient alignment cannot be surpassed by simply inserting more primal points. Fig. 5D shows that norms of the gradient of our quadrics also align well with the ground truth, favoring the validity of our approximation. Our full method performs the best on low noise levels but quickly destabilizes. This is because the system is biased to compute correct norms and it has increased parameters. We believe the reason for our compact fit to work well is the soft constraint of common scale factor acting as a weighted regularizer towards special quadrics. When this constraint cannot be satisfied, the solution settles for an acceptable shape. We also found that for a visually appealing fit, the normal alignment is crucial which is why the closest shapes arise from our method and [4]. The latter suffers from a higher point error due to the use of the strong ellipsoid prior.

Is atan2 **a valid transformation for** $\boldsymbol{\lambda}$**?** To assess the practical validity of the quantization, we collect a set of 2.5 million oriented point triplets from several scenes and use them as bases to form the underdetermined system $\mathbf{A}$. We then sample the 4th point from those scenes and compute λ. Next, we establish the probability distribution $p(\lambda)$ for the whole collection to calculate the quantiles, mapping λ to bins as the inverse CDF. A similar procedure has been applied to cross ratios in [5]. We plot the findings together with the atan2 function in Fig. 4(b) and show that empirical distribution and atan2 follow a similar trend, justifying that our quantizer is well suited to the data behavior.

Quadric detection on a real dataset There is no well-established dataset of real scenes for quadric detection.
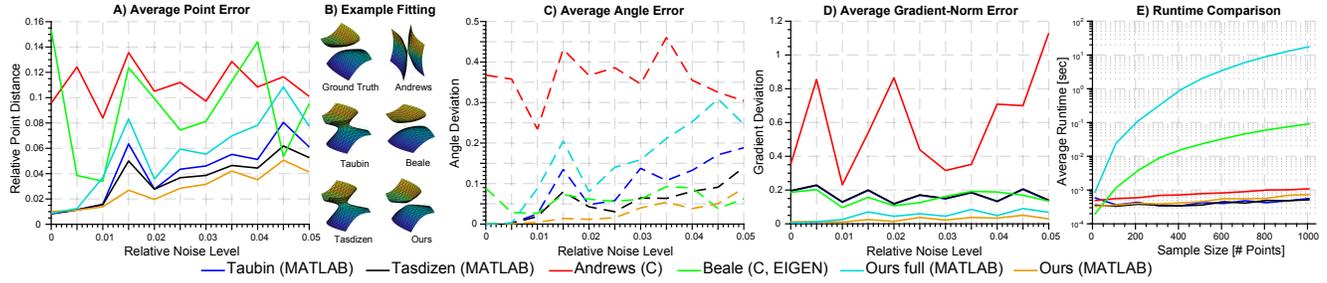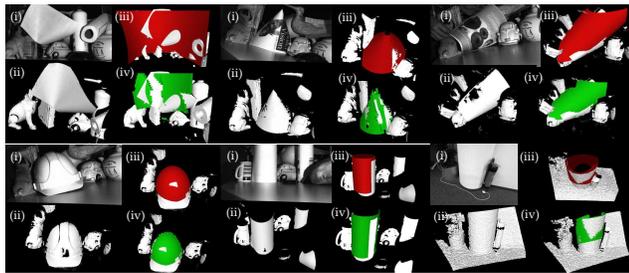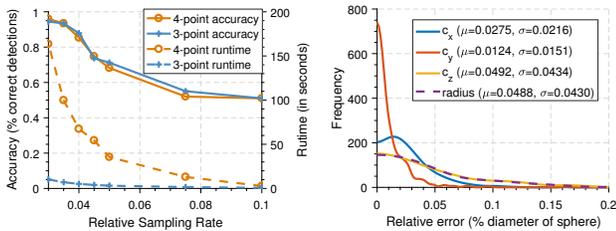
Figure 5. Synthetic evaluations. **(A,C)** Mean geometric point(A) and angle(C) errors for different quadric fitting methods. The per point error is measured as the average point-to-mesh distance from every gnd. truth vertex to the fitted quadric. The angular error (dashed) is computed as the negated dot product between quadric gradient and the gnd. truth normal. **(B)** gives exemplary fitting results while **(D)** shows the average error of the gradient norm compared to the ground truth. **(E)** plots the speed and detection rate on synthetic data.

(a) **(i)** Images captured by an industrial structured light sensor (1-5) and Kinect (last image). **(ii)** Corresponding 3D scene. **(iii)** Detected quadric, shown without clipping. **(iv)** Quadric in (iii) clipped to the points it lies on.

Table 1. Detection accuracy on real datasets.

	Dataset	# Objects	Type	Occlusion	Accuracy
Pilates Ball 1	Ours	580	Generic	Yes	94.40%
Rugby Ball	[11]	1337	Generic	No	100.00%
Pilates Ball 2	[11]	1412	Sphere	Yes	100.00%
Big Globe	[11]	2612	Sphere	Yes	90.70%
Small Globe	[11]	379	Sphere	Yes	56.90%
Apple	[11]	577	Sphere	Yes	99.60%
Football	[11]	1145	Sphere	Yes	100.00%
Orange Ball	[11]	270	Sphere	Yes	93.30%

(b) Speed and detection results.

(c) Distribution of errors of the fit.

Figure 6. Experiments on real datasets.

Therefore we collect our own set. We use a phase-shift stereo structured light scanner and capture 35 3D scenes of 5 different objects within clutter and occlusions. Our objects are bending papers, helmet, paper towel and cylindrical spray bottle. Other objects are included to create clutter. To obtain the ground truth, for each scene, we generated a visually acceptable set of quadrics using 1) [47] when shapes represent known primitives 2) by segmenting the cloud manually and performing a fit when shapes are more general. Each scene then contains 1-3 ground truth quadrics. To assess the detection accuracy, we manually count the number of detected quadrics aligning with the ground truth. We compare the 4-point and 3-point algorithms, both of which we propose. We also tried the naive 9-point RANSAC algorithm (with [51]), but found it to be infeasible when the initial hypotheses of the inlier set is

not available. Fig. 6(a) visualizes the detected quadrics both on our dataset and on a 3D data captured by Kinect v1. Fig. 6(b) presents our accuracy over different sampling rates along with the runtime performance. It is clear that our 3-point method is on par with the 4-point variant in terms of detection accuracy, while being significantly faster. We also evaluate our detector on the large objects dataset of [11] without further tuning. Tab. 1 shows 100% accuracy in locating a frontally appearing ellipsoidal rugby ball over a 1337 frame sequence without type prior. While such scenes are not particularly difficult, it is noteworthy that we manage to generate the similar quadric repeatedly at each frame within 5% of the quadric diameter.

How fast is it? As our speed is influenced by the factors of closed form fitting, RANSAC and local voting, we evaluate the fit and detection separately. Fig. 5E shows the runtime of fitting part. Our method scales linearly due to the solution of an $4N \times 10$ system, but it is the fastest approach when < 300 points are used. Thus, it is more preferred for a minimal fit. Fig. 6(b) then presents the order of magnitude speed gain, when our 4-point C++ version is replaced by the 3-point without accuracy loss. Although the final runtime is in the range of 1-2 seconds, our 3-point algorithm is still the fastest known method in segmentation free detection.

How accurate is the fit? To evaluate the pose accuracy on real objects, we use closed geometric objects of known

Figure 7. Qualitative visualizations of sphere detection in the wild: Our algorithm is very successful in difficult scenarios including clutter, occlusions, large distances. Note that the sphere is detected in 3D only using the point clouds of depth images and we draw the apparent contour of the quadric. The RGB pictures are also included in the top row to ease the visual perception.

(a) An ITODD scene composed of many cylinders

(b) Schnabel et. al. can mis-classify or -segment certain types, thereby fitting the wrong primitives.

(c) 6DoF PPF pose estimation using Cylinder as model [6,17]

(d) Our model-free, unsupervised result
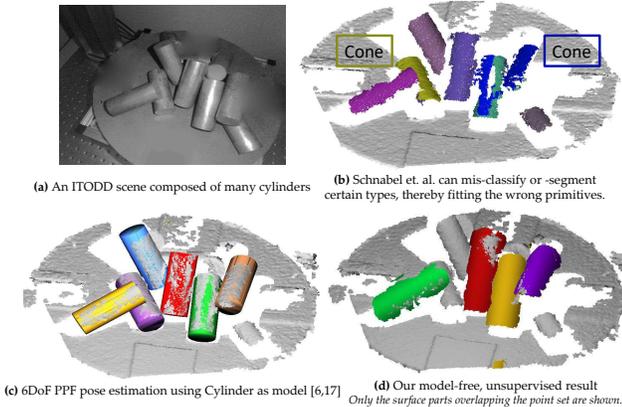*Only the surface parts overlapping the point set are shown.*

Figure 8. Multiple cylinder detection in clutter and occlusions: Our approach is type agnostic and uninformed about cylinders.

size from the datasets and report the distribution of the errors, and its statistics. We choose *football* and *pilates ball 1* as it is easy to know their geometric properties (center and radius). We compare the radius to the true value while the center is compared to the one estimated from a non-linear refinement[2] of the sphere. Our results are depicted in Fig. 6(c). Note that the errors successfully remain about the used sampling rates, which is as best as we could get.

Type-specific detection   It is remarkably easy to convert our algorithm to a type specific detection by re-designing matrix $\mathbf{A}$. We evaluate a sphere-specific detection on scenes from [11] which contains spherical everyday objects. Tab. 1 summarizes the dataset and reports our accuracy while Fig. 7 qualitatively shows that our sphere-specific detector can indeed operate in challenging real scenarios. Due to reduced basis size ($b = 1$) this type specific fit can meet real-time criteria, operating in ~ 27ms on an Intel i5 CPU.

Comparison to model based detectors   The literature is overwhelmed by the number of 3d model based pose estimation methods. Hence, we decide to compare our model-free approach to the model based ones. For that, we take the cylinders subset of the recent ITODD dataset [17] and run

---

[2]We only use Gauss-Newton non-linear quadric refinement for evaluations of accuracy and never for the actual surface estimation.

Table 2. Results on ITODD [17] cylinders: Even without looking for a cylinder, we can do better than the model based [53].

PPF3D	PPF3D-E	PPF3D-E-2D	S2D [53]	RANSAC [40]	Ours
72%	73%	74%	24%	86%	41.9%

our generic quadric detector without training or specifying the type. Visual results of different methods are presented in Fig. 8 whereas detection performance are reported in Tab. 2. Our task is not to explicitly estimate the pose. Thus, we manually accept a hypothesis if ICP [19] converges to a visually pleasing outcome. Note, multiple models are an important source of confusion for us, as we vote on generic quadrics. However, our algorithm outperforms certain detectors, even when we are solving a more generic problem as our shapes are allowed to deform into geometries other than cylinders.

Cons   Unless made specific, our method is surpassed by type-specific fits in detection rate. However, we detect a larger variety of shapes. Yet, if $\mathbf{A}$ targets a specific type, we perform even better. Degenerate cases are also difficult for us, but we find a close approximation - see Fig. 9.
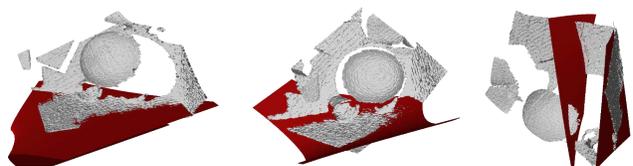

Figure 9. When planes gather a majority of the votes, our method approximates them by a close non-degenerate quadric surface.

6. Conclusion

We presented a fast and robust approach for generic primitive detection in noisy / cluttered 3D scenes. In a nutshell, a novel, linear fitting formulation for oriented point quadruplets is backed by an efficient null-space voting, potentially paving the way towards real-time operation. We establish the minimalist case known up to now: 3 oriented points. Optionally, we can convert to a type-specific fit to boost speed and accuracy. Future work will address the enlisted limitations and further analysis of the bias in the fit.

References

[1] S. Allaire, J.-J. Jacq, V. Burdin, C. Roux, and C. Couture. Type-constrained robust fitting of quadrics with application to the 3d morphological characterization of saddle-shaped articular surfaces. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 1, 2

[2] J. Andrews. User-guided inverse 3d modeling. Technical Report UCB/EECS-2013-103, EECS Department. University of California, Berkeley, May 2013. 1

[3] J. Andrews and C. H. Séquin. Type-constrained direct fitting of quadric surfaces. *Computer-Aided Design and Applications*, 11(1):107–119, 2014. 2, 6

[4] D. Beale, Y.-L. Yang, N. Campbell, D. Cosker, and P. Hall. Fitting quadrics with a bayesian prior. *Computational Visual Media*, 2(2):107–117, 2016. 2, 6

[5] T. Birdal, I. Dobryden, and S. Ilic. X-tag: A fiducial tag for flexible and accurate bundle adjustment. In *IEEE International Conference on 3DVision*, October 2016. 6

[6] T. Birdal and S. Ilic. Point pair features based object detection and pose estimation revisited. In *3D Vision (3DV), 2015 International Conference on*, pages 527–535. IEEE, 2015. 2, 5

[7] T. Birdal and S. Ilic. Cad priors for accurate and flexible instance reconstruction. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 133–142, Oct 2017. 1

[8] T. Birdal and S. Ilic. A point sampling algorithm for 3d matching of irregular geometries. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6871–6878, Sept 2017. 6

[9] M. M. Blane, Z. Lei, H. Çivi, and D. B. Cooper. The 3l algorithm for fitting implicit polynomial curves and surfaces to data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):298–313, 2000. 1, 2

[10] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter. The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2):3, 2011. 2

[11] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 7, 8

[12] G. Cross and A. Zisserman. Quadric reconstruction from dual-space geometry. In *Computer Vision, 1998. Sixth International Conference on*, pages 25–31. IEEE, 1998. 1, 2

[13] H. Deng, T. Birdal, and S. Ilic. Ppfnet: Global context aware local features for robust 3d point matching.

In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 1

[14] L. Di Angelo and P. Di Stefano. Geometric segmentation of 3d scanned surfaces. *Computer-Aided Design*, 62:44–56, 2015. 2

[15] J. Digne, S. Valette, and R. Chaine. Sparse geometric representation through local shape probing. *arXiv preprint arXiv:1612.02261*, 2016. 2

[16] B. Drost and S. Ilic. Local hough transform for 3d primitive detection. In *3D Vision (3DV), 2015 International Conference on*, pages 398–406. IEEE, 2015. 2

[17] B. Drost, M. Ulrich, P. Bergmann, P. Hartinger, and C. Steger. Introducing mvtec itodd - a dataset for 3d object recognition in industry. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 8

[18] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 1, 5

[19] A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003. 8

[20] A. W. Fitzgibbon, R. B. Fisher, et al. A buyer's guide to conic fitting. *DAI Research paper*, 1996. 4

[21] J.-M. Frahm and M. Pollefeys. Ransac for (quasi-) degenerate data (qdegsac). In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 453–460. IEEE, 2006. 2

[22] J.-M. Frahm and M. Pollefeys. Ransac for (quasi-) degenerate data (qdegsac). In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 453–460. IEEE, 2006. 2

[23] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017. 1

[24] P. Gay, C. Rubino, V. Bansal, and A. Del Bue. Probabilistic structure from motion with objects (psfmo). In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[25] K. Georgiev, M. Al-Hami, and R. Lakaemper. Realtime 3d scene description using spheres, cones and cylinders. *arXiv preprint arXiv:1603.03856*, 2016. 1, 2

[26] P. F. Gotardo, K. L. Boyer, O. R. P. Bellon, and L. Silva. Robust extraction of planar and quadric

surfaces from range images. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 216–219. IEEE, 2004. 2

[27] G. Guennebaud and M. Gross. Algebraic point set surfaces. In *ACM Transactions on Graphics (TOG)*, volume 26.3, page 23. ACM, 2007. 2

[28] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997. 6

[29] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. *Surface reconstruction from unorganized points*, volume 26.2. ACM, 1992. 6

[30] K.-i. Kanatani. Further improving geometric fitting. In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 2–13. IEEE, 2005. 2

[31] Z. Kukelova, J. Heller, and A. Fitzgibbon. Efficient intersection of three quadrics and applications in computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1799–1808, 2016. 2

[32] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. In *ACM Transactions on Graphics (TOG)*, volume 30.4, page 52. ACM, 2011. 2

[33] Y. Liu. Robust segmentation of raw point clouds into consistent surfaces. *Science China Technological Sciences*, 59(8):1156–1166, 2016. 2

[34] E. López-Rubio, K. Thurnhofer-Hemsi, Ó. D. de Cózar-Macías, E. B. Blázquez-Parra, J. Muñoz-Pérez, and I. L. de Guevara-López. Robust fitting of ellipsoids by separating interior and exterior points during optimization. *Journal of Mathematical Imaging and Vision*, pages 1–22, 2016. 2

[35] A. Makhal, F. Thomas, and A. P. Gracia. Grasping unknown objects in clutter by superquadric representation. *arXiv preprint arXiv:1710.02121*, 2017. 2

[36] J. R. Miller. Analysis of quadric-surface-based solid models. *IEEE Computer Graphics and Applications*, 8(1):28–42, Jan 1988. 1

[37] L. Minto, G. Pagnutti, and P. Zanuttigh. Scene segmentation driven by deep learning and surface fitting. In *Computer Vision–ECCV 2016 Workshops*, pages 118–132. Springer, 2016. 1

[38] E. MOORE. On the reciprocal of the general algebraic matrix. *Bull. Am. Math. Soc.*, 26:394–395, 1920. 5

[39] T. Mörwald, A. Richtsfeld, J. Prankl, M. Zillich, and M. Vincze. Geometric data abstraction using b-splines

for range image segmentation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 148–153. IEEE, 2013. 1, 2

[40] C. Papazov and D. Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *Asian Conference on Computer Vision*, pages 135–148. Springer, 2010. 8

[41] A. t. Pas and R. Platt. Localizing grasp affordances in 3-d points clouds using taubin quadric fitting. *arXiv preprint arXiv:1311.3192*, 2013. 2

[42] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys (CSUR)*, 34(2):211–262, 2002. 2

[43] K. Qin. Representing quadric surfaces using nurbs surfaces. *Journal of Computer Science and Technology*, 12(3):210–216, 1997. 2

[44] K. Qin, W. Wang, and Z. Tang. Representing spheres and ellipsoids using periodic nurbs surfaces with fewer control vertices. In *Computer Graphics and Applications, 1998. Pacific Graphics' 98. Sixth Pacific Conference on*, pages 210–211. IEEE, 1998. 2

[45] R. Qiu, Q.-Y. Zhou, and U. Neumann. Pipe-run extraction and reconstruction from point clouds. In *European Conference on Computer Vision*, pages 17–30. Springer, 2014. 2

[46] M. Rosenblatt et al. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956. 5

[47] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26.2, pages 214–226. Wiley Online Library, 2007. 2, 7

[48] A. Sveier, A. L. Kleppe, L. Tingelstad, and O. Egeland. Object detection in point clouds using conformal geometric algebra. *Advances in Applied Clifford Algebras*, pages 1–16, 2017. 2

[49] T. Tasdizen. *Robust and repeatable fitting of implicit polynomial curves to point data sets and to intensity images*. PhD thesis, Brown University, 2001. 1, 2, 3, 6

[50] T. Tasdizen, J.-P. Tarel, and D. B. Cooper. Algebraic curves that work better. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 35–41. IEEE, 1999. 3

[51] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991. 1, 2, 3, 6, 7

[52] T.-T. Tran, V.-T. Cao, and D. Laurendeau. Extraction of reliable primitives from unorganized point clouds. *3D Research*, 6(4):44, 2015. 2

[53] M. Ulrich, C. Wiedemann, and C. Steger. Combining scale-space and similarity-based aspect graphs for fast 3d object recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(10):1902–1914, 2012. 8

[54] S. Uto, T. Tsuji, K. Harada, R. Kurazume, and T. Hasegawa. Grasp planning using quadric surface approximation for parallel grippers. In *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pages 1611–1616. IEEE, 2013. 2

[55] E. W. Weisstein. Quadratic surface. MathWorld - A Wolfram Web Resource, 2017. 3

[56] D.-M. Yan, Y. Liu, and W. Wang. Quadric surface extraction by variational shape approximation. In *International Conference on Geometric Modeling and Processing*, pages 73–86. Springer, 2006. 2

[57] D.-M. Yan, W. Wang, Y. Liu, and Z. Yang. Variational mesh segmentation via quadric surface fitting. *Computer-Aided Design*, 44(11):1072–1082, 2012. 2

[58] S. You and D. Zhang. Think locally, fit globally: Robust and fast 3d shape matching via adaptive algebraic fitting. *Neurocomputing*, 2017. 2, 3

[59] H. Zhao, D. Yuan, H. Zhu, and J. Yin. 3-d point cloud normal estimation based on fitting algebraic spheres. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2589–2592. IEEE, 2016. 2