Noname manuscript No.
(will be inserted by the editor)

Model-Based Multiple Rigid Object Detection and Registration in Unstructured Range Data

Dirk Breitenreicher · Christoph Schnörr

Received: date / Accepted: date

Abstract We present a two-stage approach to the simultaneous detection and registration of multiple instances of industrial 3D objects in unstructured noisy range data. The first non-local processing stage takes all data into account and computes in parallel multiple localizations of the object along with rough pose estimates. The second stage computes accurate registrations for all detected object instances individually by using local optimization.

Both stages are designed using advanced numerical techniques, large-scale sparse convex programming, and second-order geometric optimization on the Euclidean manifold, respectively. They complement each other in that conflicting interpretations are resolved through non-local convex processing, followed by accurate non-convex local optimization based on sufficiently good initializations.

As input data a sparse point sample of the object's surface is required exclusively. Our experiments focus on industrial applications where multiple 3D object instances are randomly assembled in a bin, occlude each other, and unstructured noisy range data is acquired by a laser scanning device.

1 Introduction

1.1 Overview and Motivation

We focus on computer vision techniques for industrial tasks as illustrated in Fig. 1. Multiple instances of an arbitrary, rigid 3D object are randomly assembled in a bin. A laser scanning device acquires unstructured and noisy point measurements. The objective is to detect reliably and to determine accurately the pose of the object instances in terms of rigid body transformations for subsequent tasks, such as picking individual objects by a robot.

In this context, we focus on the following requirements:

(a) The approach should not rely on properties of *specific* objects, such as the geometry of flat disks, for instance. Rather, we only require as input a sparse point sample of the object's surface, obtained from a CAD model if available or by direct measurements if not. This enables flexible adaption to novel scenarios by non-experts as user.
(b) Numerous ambiguities due to object symmetries and occlusion require a non-local contextual first processing stage in order to reliably detect multiple object instances and rough pose estimates. The latter should be sufficiently accurate to avoid problems with local minima of subsequent pose estimation which is an intrinsically non-convex problem.
(c) The subsequent numerical pose estimation should adequately take into account the geometry of the manifold of Euclidean transformations so as to minimize the number of iterations while having a large basin of attraction to the correct local minimum.

This paper elaborates our conference contribution [11] which contains an abridged version of (c) to optimize an objective functional proposed in [12], and additionally presents the non-local first processing stage (b). Our specific contributions are detailed in Sect. 1.3.

1.2 Related Work

There is a vast literature on the processing of range data and on object registration. We confine ourselves to recent related

University of Heidelberg
Image & Pattern Analysis Group (IPA)
Heidelberg Collaboratory for Image Processing (HCI)
Speyerer Strasse 6, 69115 Heidelberg, Germany
Tel.: +49-6221-54-8875
Fax: +49-6221-54-5276
E-mail: {breitenreicher, schnoerr}@math.uni-heidelberg.de

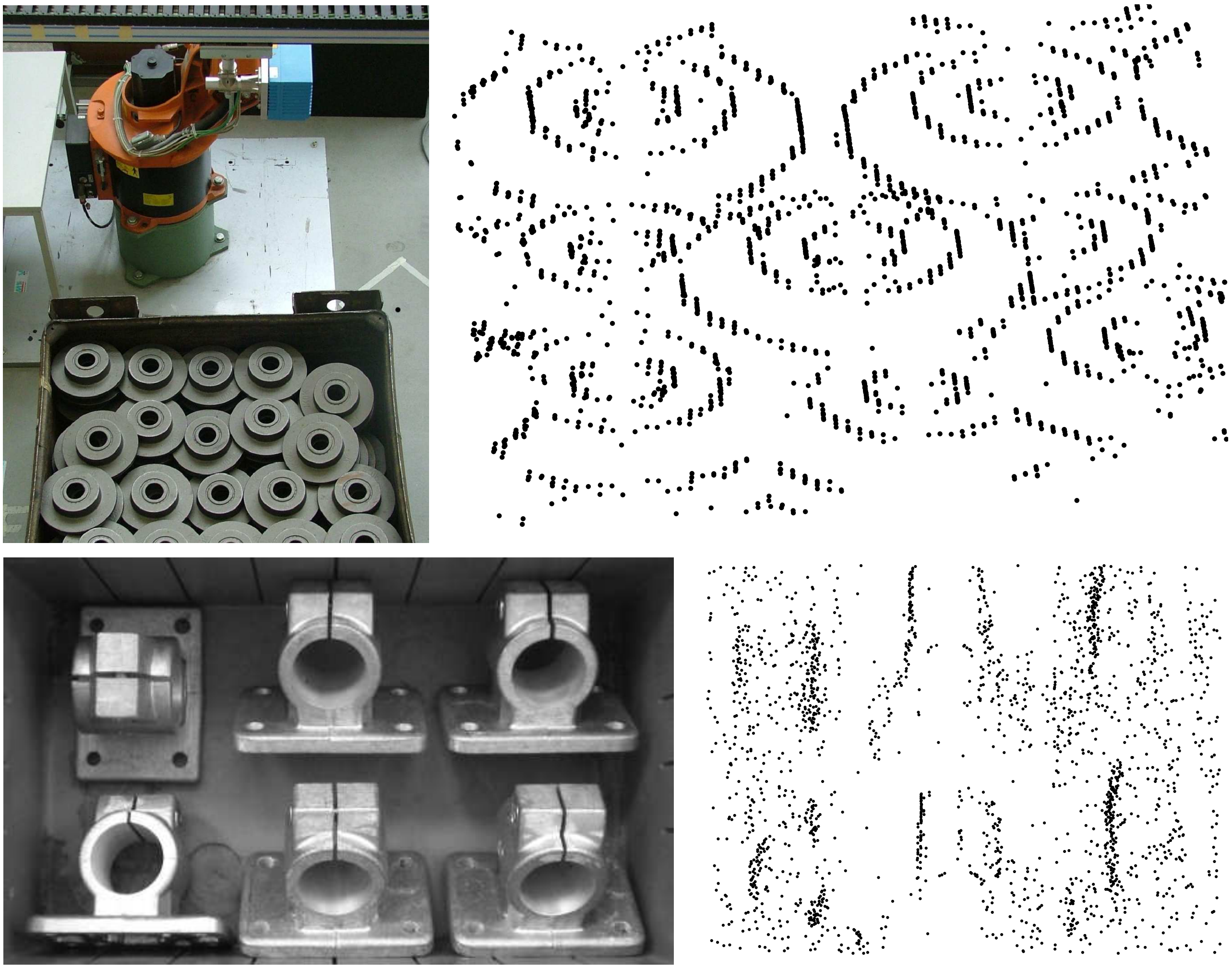

Fig. 1 Visualization of industrial scenarios motivating our work. A laser scanner is mounted on a linear axis and records the scene containing multiple objects randomly assembled in a bin. Substantial self-occlusion, noise, and unstructured sparse measurements render the task of multiple object detection and registration difficult.

work in order to elucidate the specific properties of our approach discussed under (b) and (c) above, and the way these processing stages complement each other.

Robust Iterative Registration The problem to register two point sets amounts to the chicken-and-egg problem of determining simultaneously point correspondences and a rigid transformation. Having solved either problem, the other one becomes trivial. Consequently, most approaches proceed in an alternating fashion: given an estimate of the transformation, correspondence can be determined followed by improving the estimated transformation, and so forth. The prototypical representant is the Iterative Closest Point (ICP) algorithm [8] that is due to its simplicity still a state-of-the-art algorithm [47,49,59].

It is well known that this two-step iteration is susceptible to noise and poor initialization, and numerous variants including [25,43,46] have been suggested in order to enlarge the region of attraction. A major drawback concerning the *representation* of the problem remains, however, in particular when dealing with unstructured point sets: explicit correspondences increase both the non-convexity and the non-smoothness of the objective function, and gaining insight into the optimization problem is hampered by the complicated structure of the domain of optimization comprising *both* Euclidean transformations and correspondence.

In order to obtain an optimization criteria that avoids computing corresponding points in each iteration, Mitra et al. [37] as well as Pottmann et al. [42] approximate the objective distance by local quadratic functions that represent the distance of certain points to the scene. Another way to avoid the explicit determination of correspondence has been suggested by Tsin and Kanade [54], Jian and Vemuri [31], and Wang et al. [56]. By representing point clouds of both the scene and the model by mixture distributions, registration can be achieved by minimizing the squared ℓ_2 distance

[54,31] or the Jensen-Shannon divergence [56] between two distributions. Compared to [37,42] this avoids exhaustive pre-computation of the local distance approximation at the cost of more expensive function evaluations.

As we prefer this class of approaches due to dealing with *unstructured* noisy point sets, we adopt mixture distributions to model scene and object measurements in this paper. The advantage of the resulting correspondence-independent objective function for registration is gained by loosing the possibility of closed-form local optimization, however. Moreover, the intrinsic non-convex nature of the overall problem still prevails, rendering sufficiently accurate initializations essential, similar to ICP.

To obtain initial estimates of the objects' pose, a natural approach is to identify parts of the model like cones, tubes, lines, etc., in the scene and to infer objects' pose accordingly [7,15,34]. Although such approaches can dramatically limit the amount of potential pose estimates, in view of self occlusions, noise, and the ability to uniformly deal with a large variety of objects, basing the approach on the accurate detection of a limited number of specific parts is less attractive, however. Instead, more recent work [3,16,26,27,32,48] focused on the extraction of local salient features from scene and model. Feature extraction and correspondence is quite difficult to establish, however, if objects exhibit symmetries as commonly occur in industrial settings, and if noisy and sparsely distributed samples are only available as measurements.

Another established line of research in this context concerns hypothesis generation and verification techniques [2,24,58] to obtain rough estimates of the pose [47]. Recent work [29,57] include accurate data structures to speed up the recognition process at the cost of exhaustive pre-computation, or randomized algorithms [40] along with clustering techniques to efficiently explore the corresponding voting space.

To this end, we also refer to closely related field of tensor voting, see [44] and the references therein. Due to the efficient propagation of local correspondence information, such approaches are typically superior to standard hypothesis generation approaches.

In general, however, these approaches are designed to generate hypotheses about *single* object instances matching the scene. Consequently, concerning applications with *multiple* object instances, iterative "search and pick" approaches or sequential object removal based on local strategies [34] have to be applied, where every incorrect detection, however, affects the entire subsequent process.

In contrast, we consider in this work an approach that *jointly* estimates the pose of *multiple* object instances and resolves conflicting hypotheses through *non-local* contextual processing. Furthermore, we *adaptively* prune the corresponding parameter space based on the given data in order to drastically reduce the otherwise huge problem size in an on-line manner. As detailed in Sect. 2, both objectives are accomplished by convex optimization.

Large-Scale Convex Programming Convex programming and models pervade most disciplines and current work on empirical data processing, including reasoning with dictionaries [14], compressed sensing [20], graphical models and inference [55], and machine learning [5]. Discrete and continuous graph cuts [10,13] and numerous applications provide prominent examples in the field of computer vision. The relevance of globally optimal inference for model evaluation and the guidance of convex modeling for the relaxation of more intricate models can hardly be overestimated. Accordingly, algorithms for efficiently coping with large problem sizes attract more interest in applied research.

In this paper, we aim at taming the optimization of a highly *non*-convex objective function for the registration of noisy unstructured point sets by detecting in parallel multiple objects together with rough pose estimates in a preprocessing step through large-scale convex programming. By inspecting and evaluating the optimality condition, a simple and efficiently computable criterion is obtained that can be applied to any problem instance in order to drastically reduce the problem size in an on-line fashion. For numerically solving the remaining and still large optimization problem, we competitively evaluate two different state-of-the-art approaches to sparse convex programming [9,39].

We demonstrate that in this way sufficiently accurate initializations are obtained that can be refined in a subsequent processing stage by more sophisticated local geometric optimization.

Geometric Optimization Although there are geometric optimization problems [8] that can be solved with respect to Euclidean transformations in closed form, assuming proper initializations are given, distance measures between mixture distributions representing unstructured point sets have to be minimized using methods of continuous optimization like gradient descent or Newton-like schemes. This task differs from standard applications because the underlying domain where an optimum has to be computed is a curved space (manifold).

Concerning manifolds related to the orthogonal group (Grassmann and Stiefel manifolds) continuous optimization methods are considered in [23]. Adler et al. [1], for instance, proposed a corresponding Newton-like algorithm for human spine alignment.

Pottmann et al. [42] suggested an iterative registration algorithm based on successive local first- and second-order approximations of the manifold of Euclidean transformations at the current iterate. Related problems of computer vision, including multiple point set alignment and tracking, were studied e.g. by Krishnan et al. [33], Taylor and Kriegman [51], Benhimane and Malis [4], and Drummond and Chipolla [22]. We consider the closely related geometric optimization approach [42] in more detail below and work out differences to our approach (Sect. 3).

Finally, we refer to very recent work [30, 35, 41] on global optimization approaches to the pose estimation problem, thus making the initialization problem obsolete in principle. While all of them apply Branch and Bound techniques in order to explore the pose parameter space, Hartley and Kahl [30] as well as Olsson et al. [41] require explicit correspondences of scene points to convex model parts, whereas the approach of Li and Hartley [35] works without point correspondences.

Concerning our own work, the major problem with these approaches is that run-time scales badly with the problem size, e.g. about 20min. for 200 points. Unfortunately, therefore, these sophisticated approaches are not currently applicable to realistic industrial settings with hundreds of points.

1.3 Contribution

We introduce a novel *initialization and refinement* approach for the model-based detection and determination of the rigid transformations of multiple objects in industrial bin-picking scenarios where the scene is represented by noisy, unstructured, and sparse point measurements.

The *initialization stage* in terms of a global convex objective function

- describes the *geometrical constraints* of the pose estimation problem accurately,
- allows *efficient preprocessing* techniques derived from the optimality conditions as well as application of dedicated algorithms of convex optimization, and
- yields *promising performance* making the approach attractive for solving real world applications with tight run-time constraints.

At the subsequent *refinement stage* a Newton algorithm is individually applied to each detected object that

- fully *exploits* the intrinsic *geometry* of the underlying space of Euclidean transformations,
- *convergences fast* to the local optimum, and
- exhibits a sufficiently *large region of attraction* matching the output of the preceding initialization stage.

A thorough numerical evaluation demonstrates the potential of our approach to meet the accuracy and run-time constraints of the industrial scenario. Additionally, we believe that adopting our approach might be attractive in other related scenarios of computer vision as well.

1.4 Organization

In Sect. 2, we formulate the problem of multiple rough pose estimation as a global convex optimization problem. This includes the derivation of a criterion for efficient preprocessing by inspecting the corresponding optimality condition.

To refine the initial hypotheses, we devise in Sect. 3 two different Newton procedures for geometric optimization that yield accurate results after short processing times.

In Sect. 4, we validate each steps of the overall approach by numerical experiments on synthetic data examples with ground truth. We compare two different state-of-the-art algorithms for solving the corresponding large-scale convex initialization problem and assess important properties, like the basin of attraction of geometric pose estimation.

The applicability of the complete approach to real world scenarios is demonstrated in Sect. 5. Numerous experiments show that our two-step scheme accurately detects multiple object instances along with their pose. We finally discuss pros and cons of our approach in Sect. 6 and point out further directions of research.

1.5 Notation

For readers' convenience, we briefly summarize the notation used within this work. The space of Euclidean transformations is denoted by $\mathsf{SE}(3)$, where the associated Lie algebra (cf. Sect. 3) reads as $\mathcal{T} = \mathfrak{se}\,(3)$. A Euclidean transformation in terms of a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a translation vector $t \in \mathbb{R}^3$ is written as $Y = \{R, t\} \in \mathsf{SE}(3)$. A sample of the Euclidean manifold is given by $\mathcal{S} = \{Y_j,\, j = 1, \ldots, n\} \subset \mathsf{SE}\,(3)$.

Data in terms of scene samples (point measurements) obtained by a scanning device is denoted by $\{u_i\} \subset \mathbb{R}^3$, where $i = 1, \ldots, m$. The object (model) is given by point measurements $\mathcal{O} = \{v_1, v_2, \ldots\} \subset \mathbb{R}^3$. An object $\mathcal{O}$ in pose Y is denoted by $\mathcal{O}_Y$.

Finally, a matrix $A = \{A_{ij}\} = (a_1, \ldots, a_n) \in \mathbb{R}^{m\times n}$ is given in terms of its entries A_{ij}, or column vectors $\{a_i\}$, respectively.

2 Multiple Object Detection and Pose Initialization by Sparse Convex Programming

In this section, we describe the first stage of our approach. Given point measurements of the scene, we wish to detect in parallel object instances $\mathcal{O}_{Y_l}\,,\ l = 1, 2, \ldots$, and determine rough estimates of their poses $Y_l\,,\ l = 1, 2, \ldots$, as input for the subsequent registration stage refining these estimates (Sect. 3).

To this end, we adopt the basis pursuit approach [14] based on convex programming, as illustrated in Fig. 2 for

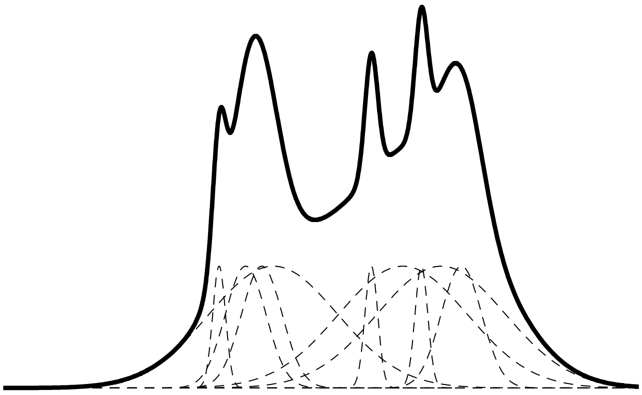

Fig. 2 Sketch of the sparse signal recovery problem. A given input signal (thick line) is approximated by a linear combination of only few basis functions (dashed lines). The selection of these basis functions is accomplished by solving for a sparse coefficient vector by convex programming. In this paper, we model the problem of multiple object detection as a sparse signal recovery problem – see Fig. 3

the original setting, and for our setting in Fig. 3. The "dictionary" in our case corresponds to a sample $\mathcal{S}$ of the Euclidean manifold and the corresponding object instances $\mathcal{O}_{Y_l}$, $Y_l \in \mathcal{S}$. Formally, this dictionary becomes quite large. Yet, we will show that by inspecting the optimality condition beforehand, the convex optimization problem can be considerably reduced such that applying a state-of-the-art solver computes the solution in few seconds only.

The approach delivers a sparse solution that effectively resolves conflicting object hypotheses due to mutually overlapping supports. A numerical evaluation of all relevant aspects will be provided in Sect. 4.

2.1 Objective Function

The distance between a scene point u_i and an object $\mathcal{O}_{Y_j}$ in terms of transformed model points v_k (see Sect. 1.5) is given by

$$d(u_i, \mathcal{O}_{Y_j}) = \min_k \|u_i - Y_j(v_k)\| \ . \tag{1}$$

Evaluating this distance function requires a careful implementation to be computationally efficient, like pre-computed look-up tables [37] or search trees [46]. An often feasible option is to separate the object $\mathcal{O}$ into simple geometric parts $\mathcal{O}^l$, $l = 1, 2, \dots$, such that the distance can be evaluated in closed form. See Sect. 5.1 discussing further implementation aspects.

Based on the distance (1), we require that a scene point u_i votes for an object instance $\mathcal{O}_{Y_j}$ only if its distance is small within a local *neighborhood*. Using indicator variables

$$\eta_{ij} = \begin{cases} 1 \text{ , if } d(u_k,\, \mathcal{O}_{Y_j}) \leq \delta \ , \ \forall u_k \in \mathcal{N}(u_i) \ , \\ 0 \text{ , otherwise ,} \end{cases} \tag{2}$$

where $\delta > 0$ is a user parameter and $\mathcal{N}(u_i)$ denotes a local neighborhood of u_i computed in a preprocessing step, we define the similarity measure $A_{ij} \in [0, 1]$ between u_i and $\mathcal{O}_{Y_j}$ by

$$A_{ij} = \exp\left(-\frac{1}{\sigma} d(u_i,\, \mathcal{O}_{Y_j})\right) \eta_{ij} \ , \tag{3}$$

where $\sigma > 0$ controls the sensitivity to noise.

Let $x \in \{0, 1\}^n$ collect indicator variables x_j representing the presence of object instance $\mathcal{O}_{Y_j}$ in the scene. The term $A_{ij}x_j$ then indicates how likely observation u_i belongs to $\mathcal{O}_{Y_j}$. Unique "explanation" for each observation in terms of an object instance, as geometry suggests, leads to the constraint

$$\sum_j A_{ij} x_j = 1 \ , \quad \forall i = 1, \dots, m \ . \tag{4}$$

As a small fraction of the measurements is caused by background, we sum up the squared residual of (4) for each scene sample to obtain the objective function

$$\|Ax - e\|^2 \ , \tag{5}$$

where $A \in \mathbb{R}^{m \times n}$, $m \ll n$, defines in (4) a large underdetermined system and $e^\top = (1, 1, \dots)^\top$ denotes the vector of ones.

2.2 Sparseness Prior

Ruling out conflicting object instances that may have caused the same observation amounts to penalize the support of solution x to (5) in terms of the (pseudo) ℓ_0-norm [21]

$$\|x\|_0 = |\{x_j,\, x_j \neq 0\}| \ , \tag{6}$$

where $|\cdot|$ denotes the cardinality of a finite set. Supplementing (5) accordingly, we obtain the objective function

$$\min_{x \in \{0,1\}^n} h(x) \ , \qquad h(x) = \mu\|x\|_0 + \|Ax - e\|^2 \ , \tag{7}$$

where $\mu > 0$ denotes the regularization parameter.

2.3 Problem Reduction and Relaxation

Finding the global optimizer of problem (7) is combinatorially complex [38] and elusive as in our applications n is very large in general. We therefore consider in this section two simplifications: Firstly, by checking the optimality conditions corresponding to (7), we can safely remove a substantial part of the variables $\{x_i\}_{i=1,\dots,n}$. Secondly, we solve the resulting much smaller problem by replacing in (7) the intricate penalty term $\|x\|_0$ by the ℓ_1-norm $\|x\|_1$, which is the "closest" convex function. We detail these two steps next.

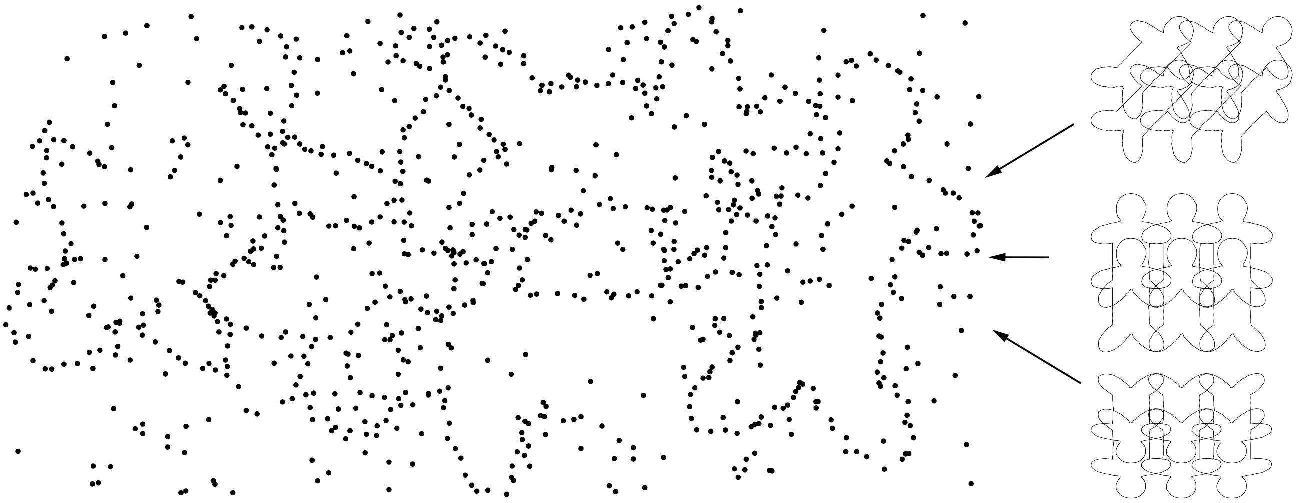

Fig. 3 Extending the principle of sparse signal recovery (see Fig. 2) to the problem of 3D template matching – here in 2D for illustration – amounts to approximate the scene (left) by a small subset selected from a large collection of candidates (right panel). Again this can be done by convex programming – cf. figures 2 and 7.

2.3.1 Evaluating the Optimality Condition

Elimination of variables x_j in a preprocessing step is based on the following

Proposition 1 *Let $x^* \in \{0,1\}^n$ be a global minimizer of the objective function $h(x)$ stated in* (7)*. Then, for all $k \in \{1,\ldots,n\}$, $x^*_k = 0$ if*

$$-\mu + 2(1^\top a_k) - {a_k}^\top a_k < 0\ . \tag{8}$$

Proof Assume $x^*_k \neq 0$. Due to global optimality

$$h(x) \geq h(x^*)\ , \tag{9}$$

holds true for all $x \in \{0,1\}^n$, in particular for $\widetilde{x}$ given by $\widetilde{x}_j = x^*_j$, $\forall j \neq k$, and $\widetilde{x}_k = 0$. By inserting $\widetilde{x}$ into (9) and inserting h from (7), we obtain

$$\mu\|\widetilde{x}\|_0 + \|A\widetilde{x} - e\|_2^2 \geq \mu\|x^*\|_0 + \|Ax^* - e\|_2^2\ . \tag{10}$$

By construction, x^* and $\widetilde{x}$ are equal except for a single entry. Thus, (10) simplifies to

$$-\mu x^*_k + 2(e^\top a_k)x^*_k - 2x^*_k {a_k}^\top Ax^* + {a_k}^\top a_k x^*_k x^*_k \geq 0\ , \tag{11}$$

and dividing by x^*_k gives

$$-\mu + 2(e^\top a_k) - 2{a_k}^\top Ax^* + {a_k}^\top a_k x^*_k \geq 0\ . \tag{12}$$

Since $A_{ij} \geq 0$, the left hand side of (12) is upper bounded by

$$-\mu + 2(e^\top a_k) - {a_k}^\top a_k x^*_k\ . \tag{13}$$

Then, due to the hypothesis $x^*_k \neq 0$ and $x^*_k \in \{0,1\}$, we finally obtain

$$-\mu + 2(e^\top a_k) - {a_k}^\top a_k \geq 0\ , \tag{14}$$

contradicting (8).

Condition (8) roughly reflects that an object in a specific pose may be present in the scene only if it “explains” a certain number of points encoded by μ. Although unlikely candidates can be removed safely according to Prop. 1, the set of candidates still contains outliers and large pose variations, see Fig. 4. Thus, the subsequent convex optimization step is essential.

Nonetheless, as we will see in Sect. 5, Prop. 1 provides the basis for drastically reducing the number of unknown variables efficiently, because the evaluation of (8) only requires simple vector operations.

2.3.2 Relaxation and Convex Optimization

A straightforward approach to optimizing problem (7) is to devise a greedy strategy. However, this would require about $O(mkn^2)$ function evaluations, where k is the number of model instances. Even after a substantial reduction of the number of free variables according to Prop. 1, such a procedure would be too inefficient to meet industrial time restrictions, in particular as k is unknown.

A more reasonable strategy to tackle (7) is to use the *convex* sparse regularizer $\|x\|_1$ instead of the non-convex penalty $\|x\|_0$ [21,53], and to relax the integer constraint $x \in \{0,1\}^n$ to $x \in [0,1]^n$:

$$\min_{x\in[0,1]^n} f(x)\ , \qquad f(x) := \mu\|x\|_1 + \|Ax - e\|_2^2\ . \tag{15}$$

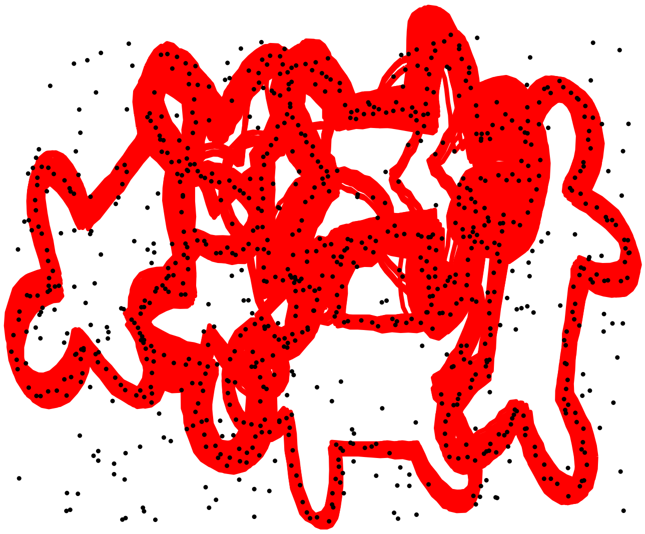

Fig. 4 Visualization of the set of candidate poses remaining after application of Prop. 1 using the set-up of Fig. 3. Although, the amount of possible candidates is reduced dramatically, there are still outliers and large pose variations that have to be removed in a subsequent convex optimization step.

The evaluation of two state-of-the-art solvers for solving (15) will be reported in Sect. 4.1.1, and the issue to convert the corresponding solution x into a binary solution is addressed in Sect. 4.1.2.

3 Pose Refinement by Geometric Optimization

The solution x to problem (15) yields both the number of detected objects and an estimate of their pose. As these estimates are related to the finite set $\mathcal{S}$ of samples of the Euclidean manifold, their accuracy is necessarily limited.

Consequently, we refine these estimates in a subsequent second processing step described in this section. Specifically, based on the initializations delivered by x, we optimize each pose individually by continuous geometric optimization on the Euclidean manifold SE(3), using an objective function that does not rely on explicit point correspondences, in view of the discussion following below.

We employ second-order approximations for fast convergence while providing a sufficiently broad basin of attraction that enables to converge to the correct local minimum. These properties will be demonstrated by numerical experiments and compared to related work in Sect. 4.

3.1 Alignment of Point Sets without Correspondence

The common objective criterion for the registration of two point sets is

$$\min_{Y=\{R,t\}\in \mathsf{SE}(3)} \sum_{i=1} \|u_i - Rv_{\mu(i)} - t\|_2^2 \,, \tag{16}$$

where $\mu(i)$ denotes the *unknown* correspondence function assigning model points to measured scene points. Rather than solving alternatingly for the transformation parameters R, t and correspondences $\{\mu(i)\}$ [8,46], which suffers from the pronounced non-convexity of the objective function (16), a smoothing procedure is advisable.

To this end, we apply a standard device well known from clustering (cf. e.g. [52] and references therein), and registration [31,54]. We represent model points by a smooth function in terms of the kernel density estimate

$$m(x;Y) := \frac{1}{m}\sum_{j=1}^{m} K\big(\frac{1}{2\sigma_m^2}\|x - Rv_j - t\|_2^2\big), \tag{17}$$

where $K(\cdot)$ denotes a smoothing kernel integrating to 1, and σ_m is a scale parameter.

A natural replacement for (16) in order to measure the distance between a model instance and the scene, is the distance between the distribution (17) and the empirical distribution of the observations

$$s(x) := \frac{1}{n}\sum_{j=1}^{n} \delta(x - u_j) \tag{18}$$

in terms of the relative entropy [18]

$$\begin{aligned} D\big(s\|m(Y)\big) &= \int s(x) \log \frac{s(x)}{m(x;Y)} \\ &= \int s(x)\log s(x) - \int s(x) \log m(x;Y) \;. \end{aligned} \tag{19}$$

Ignoring the first term as it does not depend on the pose Y, inserting (18), and using Gaussian kernel functions in (17), we obtain[1]

$$f(Y) = -\sum_{i=1}^{n} \log \frac{1}{m}\sum_{j=1}^{m} \exp\big(-\frac{1}{2\sigma_m^2}\|u_i - Rv_j - t\|_2^2\big), \tag{20}$$

where we dropped $1/n$ and the constant normalizing K.

In order to see the connection to (16), note that (20) corresponds up to the constant $1/m$ to the log-exponential function having well-known properties [45]. Correspondingly, $\forall \sigma_m > 0$, we immediately obtain the estimate

$$\begin{aligned} &\sigma_m^2 \log \sum_{j=1}^{m} \exp\left(-\frac{1}{2\sigma_m^2}\|u_i - Rv_j - t\|^2\right) - \sigma_m^2 \log m \\ &\quad\le \max_{j=1,\dots,m}\left\{-\frac{1}{2}\|u_i - Rv_j - t\|^2\right\} \qquad (21) \\ &\quad\le \sigma_m^2 \log \sum_{j=1}^{m} \exp\left(-\frac{1}{2\sigma_m^2}\|u_i - Rv_j - t\|^2\right) \;, \end{aligned}$$

that depicts

[1] We deliberately denote this objective function again with f, as in (15). By inspecting the argument and from the context, the meaning of f will be clear.

(i) that the maximum in (21) is uniformly approximated as the scaling parameter σ_m goes to zero, and consequently,

(ii) that (20) implicitly encodes the unknown correspondence function $\mu(i)$ in (16) in terms of the closest point of the smoothed model representation to the observation u_i.

The effectiveness of this smoothing procedure for dealing with unstructured point sets is further illustrated in Fig. 5.

We next focus on a numerical optimization procedure for evaluating the objective function (20).

3.2 Geometric Optimization

Newton's method is the method of choice for minimizing a smooth function $f\colon \mathbb{R}^n \to \mathbb{R}$ because it converges quadratically provided the initial point x_0 is sufficiently close to a local minimum. Based on a second order approximation of f around $x_0 \in \mathbb{R}^n$, $f(x)$ can be written as

$$f(x_0)+(\nabla f_{x_0})^\top(x-x_0)+\frac{1}{2}(x-x_0)^\top H_{x_0}(x-x_0)\ , \quad (22)$$

where $\nabla f_{x_0}, H_{x_0}$ denote the gradient and Hessian of f evaluated at x_0, respectively. Hence, the equation for determining the solution of the sufficient optimality condition is given by

$$H_{x_0}x = \nabla f_{x_0} \quad (23)$$

and can be solved numerically. In order to apply this scheme to the minimization of (20), we have to take into account that $Y \in \mathsf{SE}(3)$ is a curved space, however.

In this section, we work out two algorithms for geometric optimization that utilize second-order information, based on [42] and another variant suggested by ourselves. For the mathematical background, we refer to e.g. [19,23,36].

3.2.1 The Manifold of Euclidean Transformations

The Lie Group $\mathsf{SE}(3)$ Euclidean transformations in terms of $Y = \{R,t\} \in \mathsf{SE}(3)$ map a point x to $Yx = Rx + t$ and form a group via concatenation: $Y_1Y_2 = \{R_1,t_1\}\{R_2,t_2\} = \{R_1R_2, t_1 + R_1t_2\}$. The inverse element Y^{-1} is given by $\{R^{-1}, -R^{-1}t\}$.

For the purpose of optimization and numerical analysis, it is common to identify $\mathsf{SE}(3) \subset \mathsf{GL}(4)$ with a subgroup of all 4×4 regular matrices with respect to matrix multiplication. Keeping the symbol Y for simplicity, this representation reads

$$Y = \begin{pmatrix} R & t \\ 0^\top & 1\end{pmatrix}\ , \quad Y^{-1} = \begin{pmatrix} R^\top & -R^\top t \\ 0^\top & 1\end{pmatrix}\ . \quad (24)$$

In this way $\mathsf{SE}(3)$ becomes a differentiable manifold embedded into $\mathsf{GL}(4)$, hence a Lie group.

Tangents With each Lie group is associated its Lie algebra, the vector space tangent to the manifold at I. In case of $\mathsf{SE}(3)$, the tangent space $\mathcal{T}$ reads

$$\mathfrak{se}(3) = \left\{ \begin{pmatrix} \Phi_R & \Phi_t \\ 0^\top & 0\end{pmatrix} \middle| {\Phi_R}^\top = -\Phi_R\ ,\ \Phi_t \in \mathbb{R}^3 \right\}\ , \quad (25)$$

which is easily deduced from the fact that $\mathfrak{se}(3)$ contains all matrices Φ such that for all $\tau \in \mathbb{R}$, the matrix exponential $\exp(\tau\Phi) \in \mathsf{SE}(3)$ is a Euclidean transformation, and $R = \exp(\Phi_R)$ for some skew-symmetric Φ_R. The latter is just Rodrigues' formula for 3D rotations.

Vector space (25) is equipped with the Riemannian metric inherited from the canonical inner product $\langle\Phi_1,\Phi_2\rangle = \mathrm{tr}({\Phi_1}^\top\Phi_2)$ of the ambient Euclidean matrix space $\mathbb{R}^{4\times4}$. Furthermore, functions and the corresponding derivatives defined on $\mathsf{SE}(3)$ are evaluated at $Y = I$ without loss of generality, because during iterative optimization the current iterate Y can be regarded as offset redefining the model's original pose.

Gradients The gradient $\nabla f \in \mathcal{T}$ of a function $f\colon \mathsf{SE}(3) \to \mathbb{R}$ is uniquely defined by the relation

$$\langle\nabla f, \Phi\rangle = \langle\partial f, \Phi\rangle\ ,\ \forall\Phi \in \mathcal{T}\ , \quad (26)$$

where ∂f is the usual matrix derivative given by $(\partial f)_{ij} = \frac{\partial}{\partial Y_{ij}}f$. Eqn. (26) shows that $\nabla f - \partial f$ is orthogonal to all $\Phi \in \mathcal{T}$. Hence ∇f is the orthogonal projection $\Pi_{\mathcal{T}}(\partial f)$ of ∂f onto $\mathcal{T}$. Using the same block-factorization as in (25),

$$\partial f = \begin{pmatrix} \partial f_{11} & \partial f_{12} \\ \partial f_{21} & \partial f_{22}\end{pmatrix}\ , \quad (27)$$

this projection can be computed in closed form

$$\nabla f = \Pi_{\mathcal{T}}(\partial f) = \begin{pmatrix} \frac{1}{2}\left(\partial f_{11} - {\partial f_{11}}^\top\right) & \partial f_{1,2} \\ 0^\top & 0\end{pmatrix}\ . \quad (28)$$

Hessian The Hessian of a function $f\colon \mathsf{SE}(3) \to \mathbb{R}$, evaluated at $Y = I$, is a linear mapping from $\mathcal{T}$ onto itself given by $\overline{\nabla}_\Phi(\nabla f)\ ,\ \forall\Phi \in \mathcal{T}$, where the gradient ∇f is given by (28) and $\overline{\nabla}$ is the Levi-Civita connection defining the covariant derivative $\overline{\nabla}_\Phi$ of the vector field ∇f with respect to $\Phi \in \mathcal{T}$.

To obtain a more explicit expression in terms of the ordinary first- and second-order derivatives, we denote by $\{\mathcal{L}_k\}$ with $k = 1,\ldots,6$ the canonical basis spanning the translational and skew-symmetric components of tangents $\Phi = \sum_k \phi_k\mathcal{L}_k \in \mathcal{T}$ defined by eqn. (25). Then, the quadratic form of the Hessian with respect to any Φ is given by

$$\langle\overline{\nabla}_\Phi(\nabla f), \Phi\rangle = \partial^2 f\,(\Phi,\Phi) - \langle\partial f, \Gamma(\Phi,\Phi)\rangle\ , \quad (29)$$

with $\partial^2 f\,(\Phi,\Psi) = \sum_{ij,kl}\frac{\partial^2 f}{\partial Y_{ij}\partial Y_{kl}}\Phi_{ij}\Psi_{kl}$ and

$$\Gamma(\Psi,\Phi) = \sum_{i,j,k}\psi_i\phi_j\Gamma^k_{ij}\mathcal{L}_k\ . \quad (30)$$

The Christoffel symbols Γ^k_{ij} are listed in appendix A.

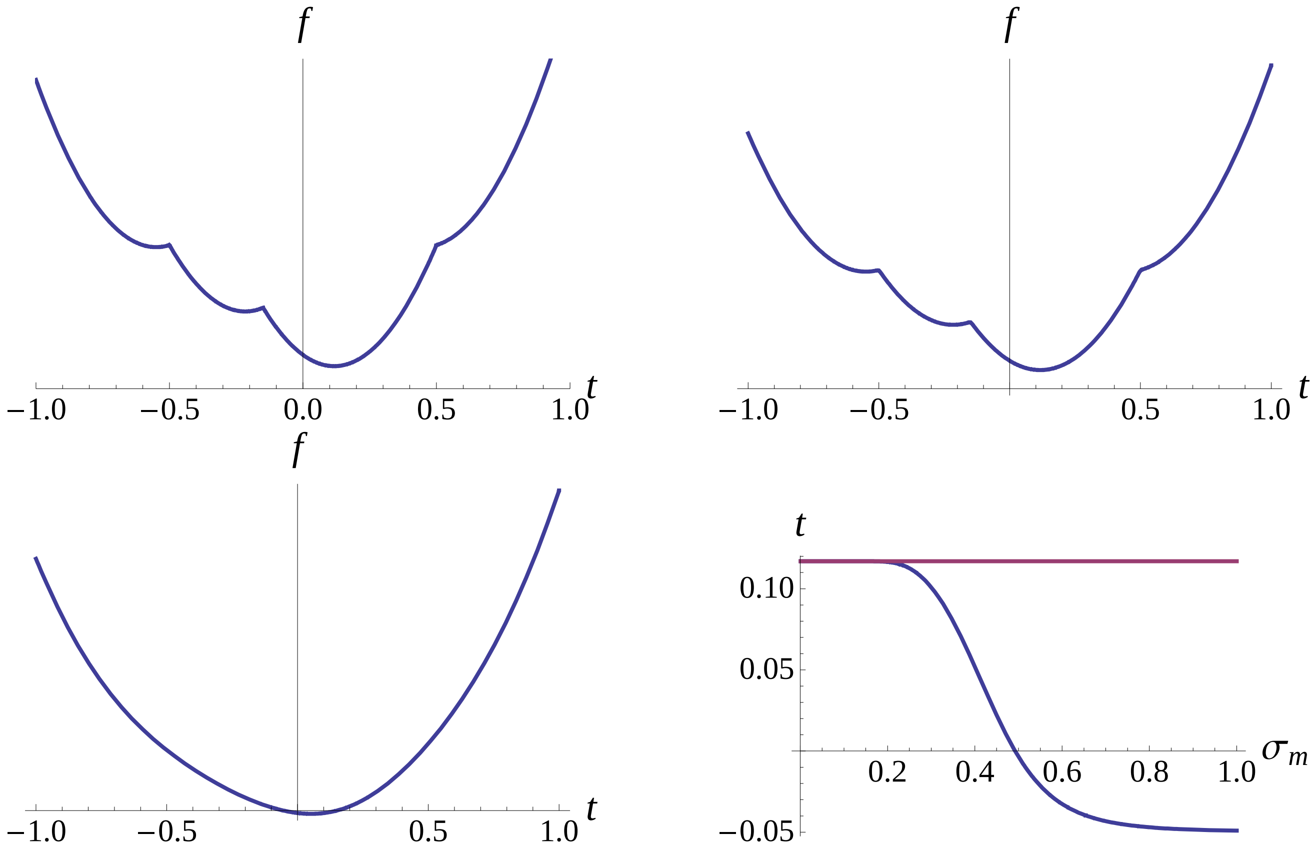

Fig. 5 Comparison of the smooth objective functional (20) with the criterion (16) where the correspondence function $\eta(i)$ assigns the observation u_i to the *closest* model point $v_{\eta(i)}$. The "model" consists of two scalar values $v_1 = 0\,,\ v_2 = 1$, and we assume to have observed the same two values as u_1, u_2, and a single additional value $u_3 \in (0,1)$ at some arbitrary position in between. We inspect both objective functions depending on the unknown translational pose parameter t, with $t = 0$ being the true unknown parameter value. Top left: Objective function (16) not only is non-convex but also shifts the global minimum. Top right, bottom left: For increasing values of σ_m objective function (20) is not only "convexified" but also exhibits a less biased global optimum. Bottom right: Position of the global optimum as a minimum, depending on σ_m. For a significant range of this parameter value, minimizing (20) gives a more accurate result. The constant value on the top corresponds to the global minimum of (16) depicted on the upper left panel.

3.2.2 Newton Optimization by Motion Approximation

Transforming a point $x \in \mathbb{R}^3$ according to the Euclidean transformation specified by (24) amounts to compute $Yx = Rx + t$, where $Y \in \mathsf{SE}(3)$ can be uniquely specified by a corresponding tangent element $\varPhi \in \mathfrak{se}(3)$ such that

$$Y = \exp(\varPhi) = \sum_{k=0}^{\infty} \frac{\varPhi^k}{k!}\ . \tag{31}$$

Accordingly, it makes sense to consider local approximations

$$Y_{lin} \approx I + \varPhi \tag{32a}$$

$$Y_{quad} \approx I + \varPhi + \frac{1}{2}\varPhi^2, \tag{32b}$$

respectively, as suggested by Pottmann et al. [42], and to determine the optimal tangent vector $\varPhi$. By inserting the approximations (32a) and (32b) into $f(Y)$, and by expanding $\varPhi$ with respect to the basis $\{\mathcal{L}_k\}_{k=1,\dots,6}$ introduced above, the objective function $f(Y)$ is restricted to the 6-dimensional vector space $\mathcal{T}$ in terms of the coefficients $(\phi_1, \dots, \phi_6)^\top$ as variables.

As a result, the linear system (23) defining the Newton iteration is replaced by (we keep the symbols H and ∂f for simplicity)

$$H(\phi) = -\partial f, \tag{33}$$

where $(\partial f)_i = \frac{\partial}{\partial \phi_i} f$ and $H_{ij} = \frac{\partial^2}{\partial \phi_i \partial \phi_j} f$ are evaluated at $\phi = 0$.

As (32a) and (32b) are local approximations of the Euclidean group, the solution $\varPhi = \sum_k \phi_k \mathcal{L}_k$ of the linear system (33) will not be an element of $\mathsf{SE}(3)$ in general. Rather, the Newton update $Y \in \mathsf{SE}(3)$ is determined by inserting $\varPhi$ into the exponential map (31).

3.2.3 Intrinsic Newton Updates

Instead of restricting first the objective function f to the tangent space $\mathcal{T}$ through the local manifold approximations

(32), and then computing Newton updates by solving (33), we may base the Newton iteration directly on the intrinsic gradient and Hessian of the manifold $\mathsf{SE}(3)$.

This means that the linear system (23) in the Euclidean case is replaced by the linear system defined by the variational equation

$$\langle \overline{\nabla}_{\varPhi}(\nabla f), \varPsi \rangle = -\langle \nabla f, \varPsi \rangle \ , \quad \forall \varPsi \in \mathcal{T} \ , \tag{34}$$

with the gradient ∇f given by (28) and the Hessian defined in (29). While system (34) is slightly more expensive to solve than (33), it better reflects the geometry of the underlying manifold. We will consider this aspect in more detail in the following subsection and demonstrate favorable properties of (34) also below in the evaluation part of this paper.

As in the case of (33), the tangent vector $\varPhi$ solving (34) does not directly results in a Euclidean transformation Y as Newton update. Rather, we have to apply the exponential mapping $Y = \exp(\varPhi)$ defined by (31), too.

3.2.4 Local vs. Intrinsic Approximation

While both schemes, (33) and (34), require to solve linear systems in each iteration, respectively, as well as retracting the obtained solution back to the manifold, there are major differences in terms of convergence properties. We address this issue in this section and take it up again in connection with discussing experimental results in Sect. 4, see in particular Sect. 4.2.2.

Recall that the objective function to be studied in this paper reads

$$f(Y) = -\sum_{i=1}^{n} \log\Big(\frac{1}{m}\sum_{j=1}^{m} \exp\big(-h_{ij}(Y)\big)\Big), \tag{35}$$

where $h_{ij}(Y) = \frac{1}{\sigma^2}\|u_i - Rv_j - t\|_2^2$ and $Y \in \mathsf{SE}(3)$.

Approximating the rigid body transformation by truncating (31) after the linear term (32a) yields a redefinition of h_{ij} such that optimization of f is restricted to the tangent space $\mathcal{T}$. As this approach provides an accurate approximation only within a small neighborhood around the current iterate, however, convergence to the correct local optimum is unlikely if it lies outside this neighborhood [42].

In contrast, second order truncation (32b) provides a more accurate local approximation of the manifold $\mathsf{SE}(3)$. On the other hand, inserting the quadratic approximation into h_{ij} maps $Rv_j + t$ to

$$v_j + \varPhi_t + \varPhi_R v_j + \frac{1}{2}\varPhi_R\big(\varPhi_t + \varPhi_R v_j\big). \tag{36}$$

Using the fact that $\varPhi_R$ is skew symmetric, the latter part rewrites as

$$\frac{1}{2}\big(\varPhi_R\varPhi_t + (\phi^\top v_j)\phi - (\phi^\top\phi)v_j\big), \tag{37}$$

where ϕ are the coefficients of the expansion $\varPhi_R = \sum_k \phi_k \mathcal{L}_k$.

As a consequence, when the rotation components of Newton updates happen to become large in magnitude, the nonconvexity of the objective function due to the quadratic terms in (37) may cause Newton updates to step into wrong directions. This will be confirmed by numerical experiments in the following section.

This argument can be underlined by considering the Rodrigues' formula, the closed form expression of the exponential map

$$R = I + \varPhi_R\frac{\sin(\|\varPhi_R\|)}{\|\varPhi_R\|} + \varPhi_R^2\frac{1-\cos(\|\varPhi_R\|)}{\|\varPhi_R\|^2} \ . \tag{38}$$

Approximating the trigonometric function by its first and second order Taylor expansion in $\|\varPhi_R\|$, given by

$$\sin(\|\varPhi_R\|) \approx \|\varPhi_R\| \ , \ \cos(\|\varPhi_R\|) \approx 1 \ , \tag{39a}$$

$$\sin(\|\varPhi_R\|) \approx \|\varPhi_R\| \ , \ \cos(\|\varPhi_R\|) \approx 1 - \frac{1}{2}\|\varPhi_R\|^2 \ , \tag{39b}$$

respectively, insertion into (38) directly results in (32a) and (32b). Thus, with increasing $\|\varPhi_R\|$ the approximation fails to be accurate. Moreover, as this approximation affects the translation part too, large magnitudes in rotation affects the accuracy in t.

Another issue concerns the choice of the metric. While we suggest the canonical metric in the ambient space [23], embeddings of the Euclidean transformations into $\mathbb{R}^6$ and using the corresponding metric, i.e. the standard inner product in $\mathbb{R}^6$, results in a different scaling of the rotational part.

Moreover, representing $\varPhi$ in terms of its basis expansion, first and second order approximation yield the restriction of $f : \mathsf{SE}(3) \mapsto \mathbb{R}$ to $f : \mathbb{R}^6 \mapsto \mathbb{R}$. Consequently, second-order derivatives are symmetric in the latter Euclidean space, i.e. $\frac{\partial^2}{\partial\phi_i\partial\phi_j}f = \frac{\partial^2}{\partial\phi_j\partial\phi_i}f$. As in general the Lie bracket of two elements $\mathcal{L}_i, \mathcal{L}_j \in \mathfrak{se}(3)$ does not vanish, however, using standard second-order derivatives only yield approximations to the correct Hessian. Thus, if the components of the transformation become large in magnitude, the resulting approximation of the Hessian in (33) becomes worse, whereas (34) is based on (29) that includes corrective terms and thus better reflects the geometry of the underlying space. Our numerical evaluation discussed in the subsequent section demonstrates that this difference is relevant to applications.

4 Numerical Evaluation

Accuracy, robustness and speed are of primary importance for industrial applications. In this section, we therefore analyze our proposed two-step approach accordingly using synthetic data samples with ground truth. Real-world applications will be discussed in Sect. 5.

4.1 Initialization Estimation by Convex Optimization

Finding proper initializations amounts to solve the convex optimization problem (15). In the following two subsections, we separately discuss the two major issues involved in this connection: the large problem size of the convex relaxation, and the conversion of the global optimum to a binary solution in a post-processing step.

4.1.1 Convex Optimization

We study two different state-of-the-art approaches to solve (15): the *Spectral Projected Gradient (SPG)* method [9] and *Nesterov's algorithm* [39]. Both algorithms only require evaluations of the objective function and its gradient, hence are suited for large-scale sparse convex programming.

Spectral Projected Gradients (SPG) The general idea underlying SPG [9] is to successively approximate the objective $f(x)$ in terms of the current iterate x_k by the simplified Taylor series

$$f(x_k) + (\nabla f(x_k))^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top \lambda I (x - x_k) \;, \tag{40}$$

where λI with $\lambda \in \mathbb{R}$ corresponds to a simplified approximation of the Hessian.

A non-monotone line search allows to temporarily increase the objective such that variables of the optimal configuration can be fixed in early iterations. Due to the second-order approximation of the objective in terms of the Hessian λI, SPG belongs to the class of quasi-Newton methods that typically exhibit fast convergence to the global optimum.

A drawback of the method is that no accuracy bound can be guaranteed depending on the number of iterations.

Nesterov's Algorithm Accuracy bounds are provided by Nesterov's optimization procedure [39]. This approach is based on the Lipschitz continuity of the gradient of f (constant L) and computes the optimal configuration by subsequently solving simple minimization problems of the form

$$y_k = \min_{y \in [0,1]^n} \left(\langle \nabla f(x_k), y - x_k \rangle + \frac{1}{2} L \|y - x_k\|^2 \right) \tag{41a}$$

$$z_k = \min_{x \in [0,1]^n} \left(\frac{1}{\sigma} L d(x) + \sum_{i=0}^{k} \alpha_i g_i(x) \right) \tag{41b}$$

where $g_i(x) = f(x_i) + \langle \nabla f(x_i), x - x_i \rangle$ corresponds to an approximation of f at x_i, $d(\cdot)$ being a proper prox-function, σ the corresponding convexity parameter and the next iterate x_{k+1} is given by $\frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k$. While (41a) bounds the deviation from the current iterate, (41b) takes into account previous iterates in order to model the objective function locally. For further details, we refer to [39] and the references therein.

Let $x^* \in [0,1]^n$ denotes the global optimum of the convex function f. Then the error bound

$$f(y_k) - f(x^*) \leq \frac{4 L d(x^*)}{\sigma (k+1)(k+2)} \tag{42}$$

holds depending on the number of iterations k. We point out that the Lipschitz constant L of the objective function's gradient appears in both problems (41a) and (41b). As a consequence, having a tight estimate of L is essential for the performance of this method.

Comparison In order to competitively evaluate the performance of SPG and Nesterov's approach, we consider the 2D setup depicted in Fig. 3. Using a total of $1\,335\,840$ candidate transformations, application of Prop. 1 fixes $\approx\ 99.7\%(!)$ of the variables beforehand. The remaining 3497 variables were determined using SPG and Nesterov's algorithm, respectively.

Concerning Nesterov's approach, we used three methods to numerically determine or estimate the Lipschitz constant $L = \|A^\top A\|_2$ of the gradient of f: the power iteration [28] to compute L, application of Gerschgorin's disk theorem to obtain an upper bound, and evaluating the trace of $A^\top A$ returning the sum of all eigenvalues as upper bound.

While the power iteration converges within few iterations it has to perform multiple matrix-vector multiplications and therefore took about 0.75 seconds. In contrast Gerschgorin's disk theorem only requires inspection of the data matrix and computed an upper bound in 0.11 seconds. Finally, the trace operator returned an upper bound within 0.02 seconds whose quality highly depends on the number of dominant eigenvalues that increase with the number of objects in the scene.

Our numerical experiments confirmed that the value chosen for L highly influences the performance of Nesterov's algorithm, see Fig. 6. SPG on the other hand outperforms Nesterov's approach in terms of the number of iterations. This however is primarily due to the line search involved and at the cost of additional function evaluations such that the time per iteration is significantly smaller for Nesterov's approach, resulting in an overall faster convergence.

As a consequence, for our real world experiments summarized in Sect. 5, we throughout used Nesterov's algorithm to determine the solution of (15).

4.1.2 Binarization of the Solution

Due to the relaxation of the integer constraint, the global optimum x^* of (15) in not an element of $\{0,1\}^n$ in general but has real-valued components $0 < x_i < 1$.

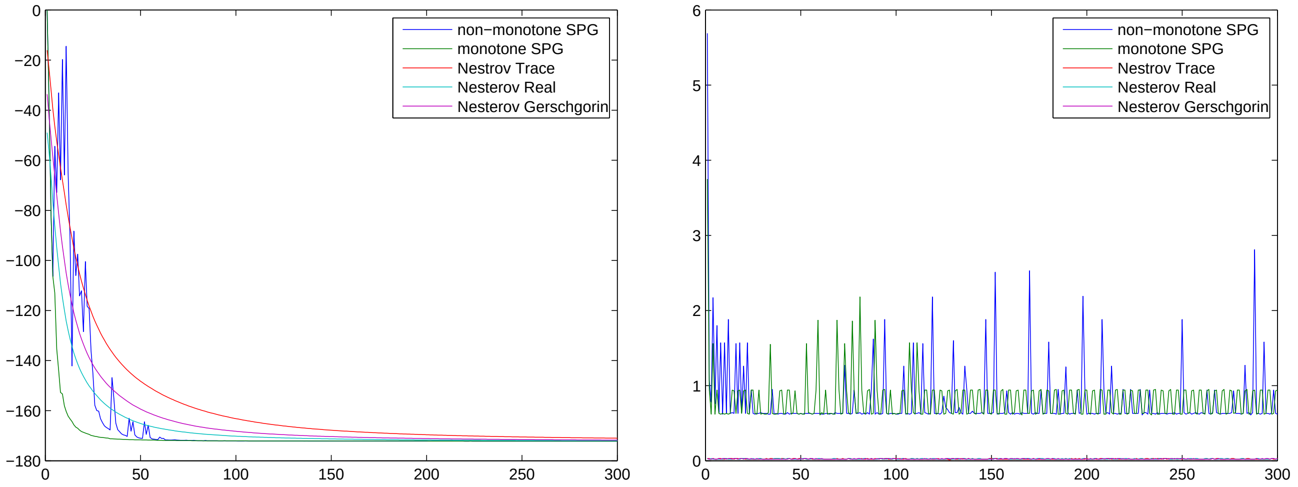

Fig. 6 Comparison of optimization algorithms using the experimental setup of Fig. 3: While the SPG algorithm typically requires less iterations to converge to the global optimum (left: energy vs. iterations), due to the line search involved, more time per iteration (in seconds) is spent in comparison to other approaches (right: time [sec.] vs. iterations).

To infer a corresponding high-quality discrete configuration, we studied two post-processing steps described next. Neither of them guarantees to return the *discrete* optimal solution to (7), of course.

Clustering Regarding the results of the convex optimization procedure as probabilities x_i^* indicating the presence of an object with pose Y_i, the components of x^* typically form compact clusters in the model-pose space and are well-localized in the image domain – see Fig. 7.

Consequently, a simple clustering post-processing step, where nearby poses are assigned to the same cluster, followed by averaging the elements within each cluster provides a high-quality solution. Strictly speaking, this clustering step should take into account the underlying manifold geometry (cf., e.g. [50]). Due to the clusters' compactness, however, simple Euclidean clustering turned out to work very well for computing a reasonable initialization of the subsequent geometric optimization procedure (Sect. 3.2 and 4.2), that *does* take into account the underlying geometry.

Randomized Rounding This method proceeds by excluding in turn each variable x_i for a candidate pose i and solving the convex relaxation for the remaining variables. Again, interpreting the globally optimal values x_j^* as probabilities, we set x_i to 0 if

$$\begin{aligned} &E\left[f(x)|x_i = 0, x_{i-1}, \ldots, x_1\right] \\ \leq\, &E\left[f(x)|x_i = 1, x_{i-1}, \ldots, x_1\right] \end{aligned} \tag{43}$$

and to 1 otherwise, where E denotes the expected value of $f(x)$ with respect to the probability distribution, estimated by averaging samples.

The binary solution obtained by this procedure is guaranteed to differ from the global continuous configuration only by a fixed constant [6]. Yet, due to the need to solve the large-scale optimization problem multiple times, the clustering procedure sketched above turned out to be a better compromise between accuracy of initialization and computational speed.

4.2 Geometric Fine Alignment

Next, we evaluated the geometric optimization algorithms presented in Sect. 3 by applying it to computer-generated point sets, and analyzed the performance with respect to runtime and robustness to inaccurate initializations.

4.2.1 Speed of Convergence

Algorithms like ICP [8] or Softassign [43] return less accurate registrations in cases where the underlying point set has no or only few salient regions. This often occurs in industrial applications where smooth surfaces have to be registered accurately. To compare the ability of the approaches to cope with such scenarios, we generated 2500 data points by randomly sampling from the smooth function $3(x-1)^2 + 3\sin(2y)$ on the unit interval $[0,1]^2$.

We transformed a copy of the model only slightly (about 4 degree in each rotation and by a total of 0.12 in translation), such that all approaches including ICP [8], Softassign [43], the Newton schemes based on local approximation [42] and the approach proposed in this paper converged to the true solution. Figure 8 reveals that the convergence rates differ significantly.

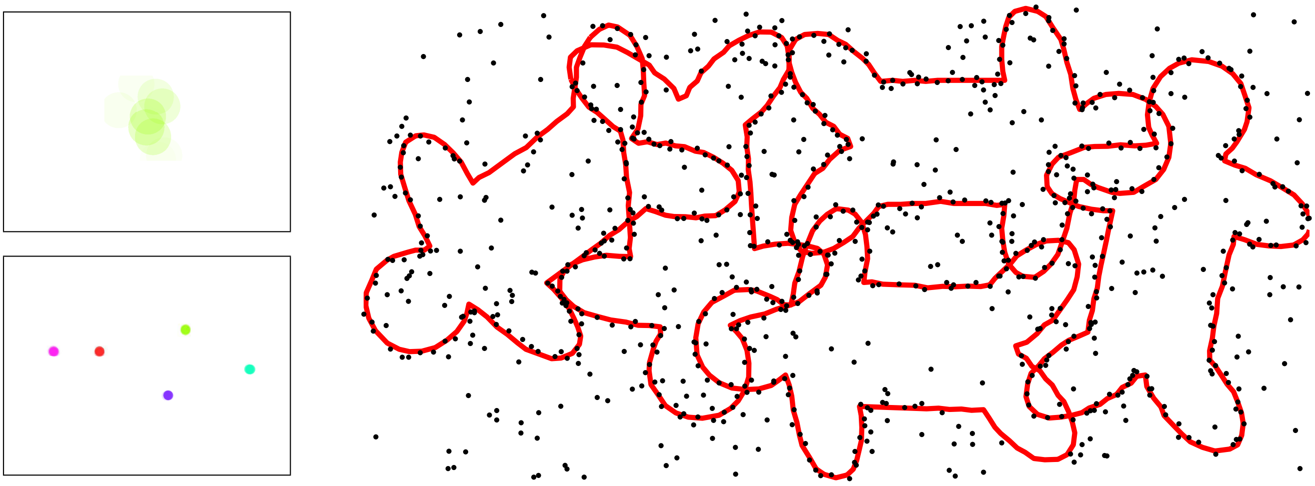

Fig. 7 Despite of a substantial amount of noise and background clutter, poses indicated by x after convex optimization are compactly located (left). A close-up view is shown top left. Dots indicate the center of the model and colors their orientation. The correct object instances in the scene (red shapes) can be determined quickly by a clustering post-processing step.

While for varying σ_m the Newton procedures based on local approximations of the Euclidean group (Sect. 3.2.2) converge slightly faster than the approach presented in this paper, all of them exhibit quadratic convergence. In contrast, ICP and Softassign only converge linearly to the optimal configuration. As a result, they return less accurate registrations under tight run-time constraints (fixed number of iterations).

The superior performance of the Newton schemes is at the cost of more expensive computations for determining the Hessian in each iteration. While ICP requires $O(M \log N)$ computations in each iteration using K-D trees, the evaluation of the gradient and the Hessian of (20) causes costs of $O(MN)$. As a result, a single round of ICP requires about 1 second. In contrast, the computation of the derivatives, using MatLab research code, needs between 8 (linear and quadratic approximation [42]) and 12 seconds (our approach). This difference is primarily due to the higher dimension of the ambient space in which the gradient and the Hessian are computed. We expect however that when using a C-tuned implementation the Newton approaches will considerably catch up with ICP.

4.2.2 Region of Attraction

Fast convergence is immaterial if the algorithm gets stuck or converges to the wrong local minimum. Robustness to poor initializations is therefore important. The region of attraction for ICP [8] has already been analyzed in [37]. We therefore only consider Newton procedures here.

For comparison, we used the same initial setup as [37], i.e. a model of the Stanford Bunny rotated around the z-axis and shifted in the x-y plane by the size of the model. As scene we used a copy of the model placed in the origin. Because we are primarily interested in quadratic and fast convergence and the resulting accuracy after a fixed run-time, we terminated all second-order algorithms after 25 iterations.

We observed that especially for transformations with rotational initialization error, the Newton approach proposed in this work has a significantly larger domain of attraction to the correct solution than the procedures based on local approximations of the Euclidean group, as visualized in Fig. 9. This finding confirms the discussion in Sect. 3.2.4.

5 Industrial Application

In this section, we apply and evaluate our two-stage approach to the real-world bin-picking scenario. To this end, we used both computer-generated data allowing for full control of the evaluation by simulating the scanning device and noise, and real industrial data as shown in Fig. 1.

5.1 Efficient Initialization

Short processing times are important for many industrial applications. We briefly point out properties of our approach enabling fast on-the-fly computations of some steps of the overall approach.

Concerning the preprocessing based on Prop. 1, only the object in position $\mathcal{O}_{Y_k}$ is required to compute the corresponding column vector a_k and to determine if the related indicator variable x_k can be set to zero (i.e. ignored) immediately. Furthermore, each entry of a_k can be computed in parallel. Finally, each entry in a_k is given by (3) that only

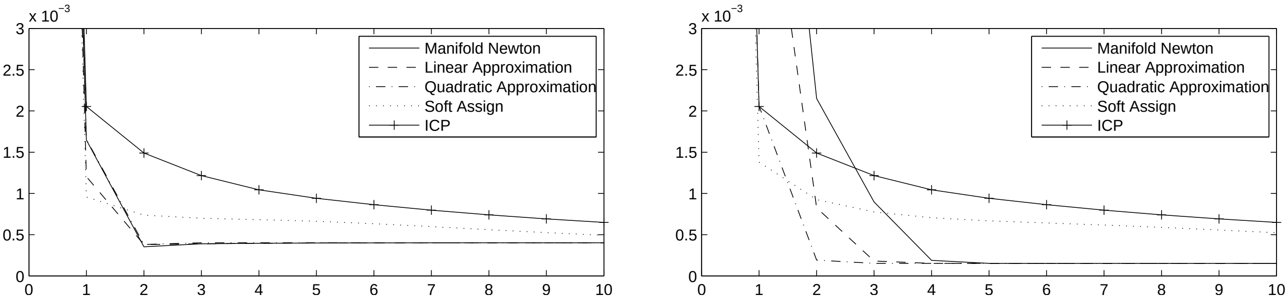

Fig. 8 Evaluation of the performance of Newton algorithms based on linear and quadratic motion approximation [42], the approach proposed in this paper (manifold Newton) as well as ICP [8] and Softassign [43], for different values of σ_m (left: 0.3, right: 0.15). Each plot visualizes the objective function values for subsequent iterates. ICP and Softassign converge linearly while the remaining approaches converge quadratically.

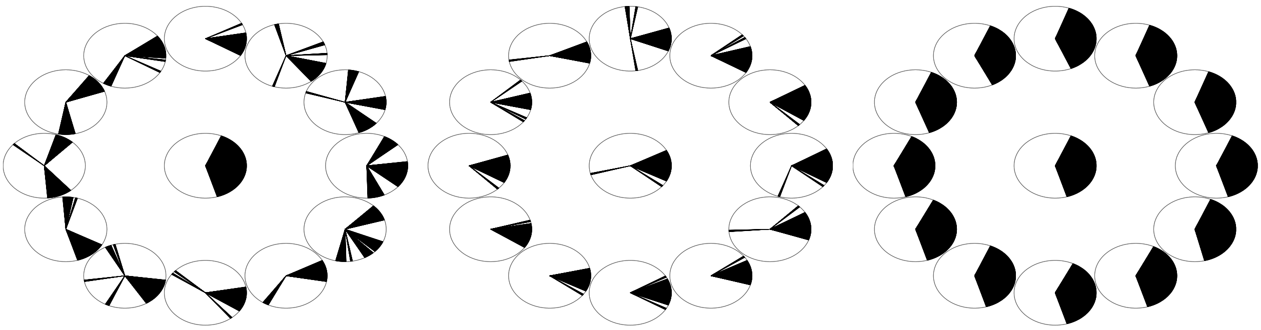

Fig. 9 Evaluation of the *region of quadratic convergence* for Newton's algorithm based on linear (left), quadratic (middle) local approximation [42], and on the intrinsic local approximation (this paper, right), for fixed $\sigma_m = 0.1$. Each circle center together with the circle center in the middle shows the initial translation offset of the model vs. the scene in the $x - y$ plane. Slices in each circle refer to the initial rotation around the z-axis. They are colored black if the model converges to the scene within the first few iterations and otherwise remained white. The results illustrate that the approach proposed in this paper is significantly and uniformly more robust against inaccuracies of initialization.

requires to compute the shortest distance (1). Using precomputed distance maps [37], this evaluation amounts to inspect a look-up table.

As a consequence, the only remaining costly part is the computation of the local neighborhood for which a range of established efficient algorithms and data structures such as kd-trees are available.

5.2 Computer-Generated Data

To evaluate the accuracy of our approach in a fully controlled environment, we generated different “realistic” data sets by simulating the real world scanning device of Fig. 1 and noise, for real objects.

Each object instance was randomly placed in the scene including partially overlapping objects. Additionally, to cover a wide range of applications with different input data, we used both object models exclusively based on edge data as well as models obtained by reference scans, as depicted in Figs. 10, 11, and 12, respectively. Again, we point out that different input formats are uniformly handled by our approach.

	Link	Hook	Mech-part
# candidate instances	1 524 600	2 413 675	394 975
# instances pruning	228	336 681	4 597
# instances optimization	9	47	11
# instances clustering	5	5	5

Table 1 Quantitative evaluation of the initialization phase and of the first processing stage (non-local multiple object detection through convex optimization) for the data sets shown in Figs. 10,11, and 12 where the rows refer to the number of candidate instances in total, the number of non-zero instances remaining after pruning, the number of non-zero instances remaining after convex optimization, and the number of non-zero instances remaining after clustering, respectively

The collection of candidate poses for multiple object detection was compiled by discretizing the space of possible rotations in intervals of $15°$ and ranges of the translation vector such that at least a single point in the scene can be fitted accurately. This resulted in a total of up to $2\,413\,675$ possible candidate instances. Table 1 displays all relevant numbers.

Our current research code does not exploit the measurements listed in Sect. 5.1 to accelerate the initialization phase,

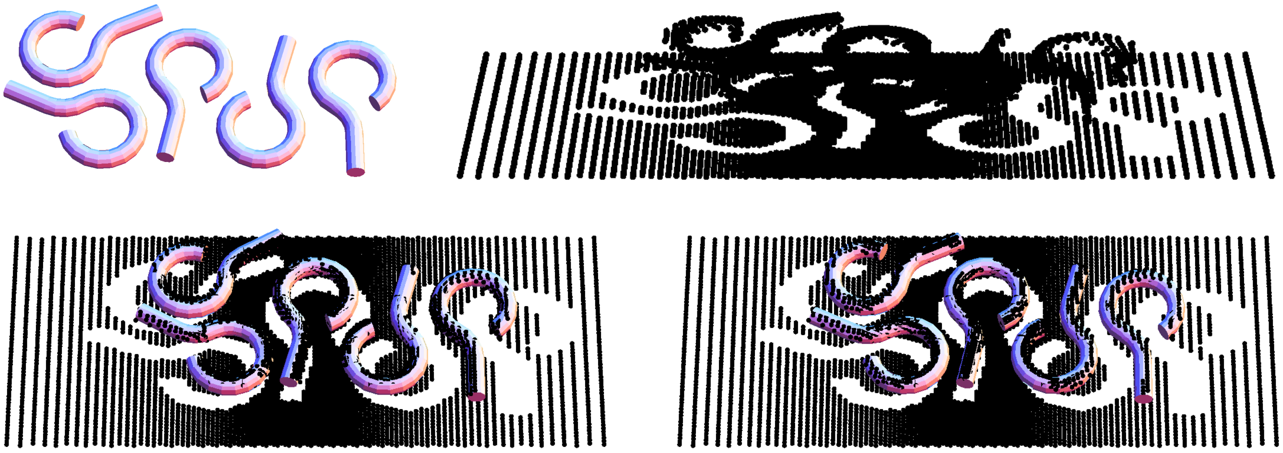

Fig. 10 Object detection and localization with real world objects (top left) in 3D scanning data obtained by simulating a SICK LMS 400 scanning device (top right). While the convex initialization step simultaneously gives proper estimates of the number of objects as well as the corresponding transformations (bottom left), running subsequently few iterations of the geometric optimization approach yields accurate registration results (bottom right).

yet. Rather, we computed the full matrix A off-line which took several minutes.

Parameters δ, eqn. (2), σ, eqn. (3), and μ, eqn. (15), of our approach are set by hand for each scenario. These values reflect the characteristics of the *scenario*, i.e. the noise level and the spacing of the *model* points. Their choice is therefore straightforward and does not require elaborate tuning. We point out that they only depend on the scenario (noise, object models), and *not on the specific given scene (data)* of a fixed scenario to be analyzed.

The elimination of variables in the preprocessing step (Sect. 2.3.1) reduced the dimension by 86% to 99.9% such that the final convex optimization procedure returned multiple objects within few seconds only.

Pose clustering according to Sect. 4.1.2 provides rough initializations used for subsequent fine alignment through geometric optimization. In case of the mechanical part (Figure 11) the deviation of the estimated position from ground truth was at most $5°$ rotation and $\approx 5.3\%$ translation of the model size. Similar results have been obtained for the link (rotation error $\leq 4.5°$, translation error $\leq 1.8\%$) and the hook data set (rotation error $\leq 5°$, translation error $\leq 2.4\%$). The trade-off between the computational costs of the first non-local convex processing stage and the subsequent geometric optimization depends on how finely the pose space is discretized (problems size vs. inaccurate detection) and can certainly be optimized for fixed industrial scenarios.

Running the geometric optimization algorithm (Sect. 3) for each detected object returns a final pose estimate within few iterations. At this second stage of the overall approach, we used an additional background kernel in (17) with a large scale parameter σ_m to cope with structured outliers [17], i.e. nearby objects.

Because geometric optimization converges to a *local* optimum, occlusion configurations may occasionally lead to erroneous updates of the corresponding Newton algorithm. Figure 12 depicts such a scenario where due to locally “looking through holes” no consistent match of the sparse model points to scene points is possible.

5.3 Real World Industrial Data

We applied our approach to the real-world industrial scenarios depicted in Fig. 1, comprising 3D noisy and unstructured scanning data of brake-discs and flanges, respectively. Similar to the synthetic scenario, we set the parameters δ, σ, μ by hand according to the noise level and the spacing of the model points and kept these values for all corresponding experiments. At the second stage, i.e. the refinement, we also used an additional background kernel to cope with structured outliers [17]. Furthermore, we used a machine with a Pentium 4, 3.00 GHz processor.

Based on expert’s knowledge, i.e. knowing that brake-disc objects are never located upside down, we sampled the model at 10 different points for each circle and discretized the space of rotations within the interval of $[-15, 15]$ degrees for each free axis (the model is rotation invariant with respect to the third axis). This resulted in a total of 1420 candidate objects poses that can yield a single scene point and took about 0.2 seconds computation time.

The preprocessing step reduced the problem size by eliminating $\approx 99.5\%$ of the variables immediately. The subsequent global convex optimization determined the remaining 4320 variables in 14 seconds. Even highly occluded model instances are detected accurately as indicated by the blobs

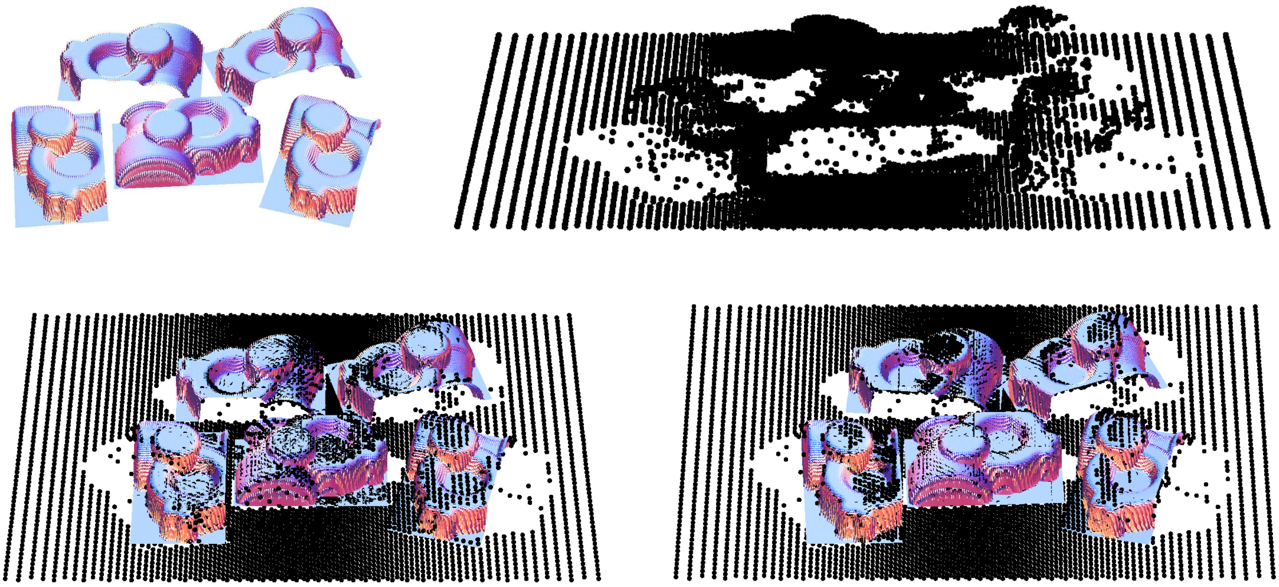

Fig. 11 Object detection and localization with real world objects in 3D scanning data obtained by simulating a SICK LMS 400 scanning device. While the convex initialization step simultaneously gives proper estimates of the number of objects as well as the corresponding transformations (bottom left), running subsequently few iterations of the geometric optimization approach yields accurate registration results (bottom right).

on the right hand side of Fig. 13 marking the hypotheses corresponding to the objects' pose.

Applying the subsequent geometric optimization for at most 5 iterations where each iteration required about 1 second, turned out to be sufficient to accurately locate all object instances located in the bin.

For the complex objects shown in the lower panel of Fig. 1 the approach was able to detect the objects and to determine proper initial pose estimates in the corresponding highly unstructured point set – see Fig. 14, left panel. Again, subsequent geometric optimization determined the final object positions within few iterations.

However, geometric optimization may fail if the initial pose estimate does not fall into the region of convergence of the Newton updates on the manifold, as indicated in Fig. 9. This fact is well known from standard Newton-based optimization in Euclidean spaces, too. We cope with this issue by resorting to *first*-order optimization techniques on the group of Euclidean transformations [12] if the initial Newton updates do not sufficiently decrease the objective function value. The object marked with red in Fig. 14 shows such an example, where the Newton method failed and switching to first-order optimization safely converged, at the cost of a higher number of iterations.

Finally, we demonstrate the robustness of the non-local detection stage with respect to similar looking but *different* objects. Fig. 15 shows a single disc embedded into other discs, and whose radius of the inner ring is slightly larger than that of all other discs. Although this disc is very similar to the other discs, multiple object detection through convex programming only return this single object instance.

6 Conclusion and Discussion

We presented a novel two-stage approach for the model-based detection and localization of multiple objects in industrial bin-picking scenarios from noisy, unstructured and sparse point measurements.

We formulated the problem of finding good initialization hypothesis in terms of a global convex objective function that reflects geometric constraints and provides a basis for efficient preprocessing techniques that drastically reduce the problem size. We evaluated state-of-the-art sparse solvers for the corresponding large-scale convex programs and demonstrated promising performance in terms of accuracy of multiple object detections and in view of industrial runtime constraints. We pointed out techniques for considerably speeding up the preprocessing stage for which, obviously, modern graphics hardware may be used to achieve further accelerations.

To refine the hypothesis individually, we suggested a Newton algorithm that fully exploits the intrinsic geometry of the underlying space of Euclidean transformations and exhibits fast convergence to a local optimum as well as a significantly enlarged region of attraction.

Although the presented approach is designed to handle single rigid models, it can be extended to cope with multiple rigid object models straightforwardly.

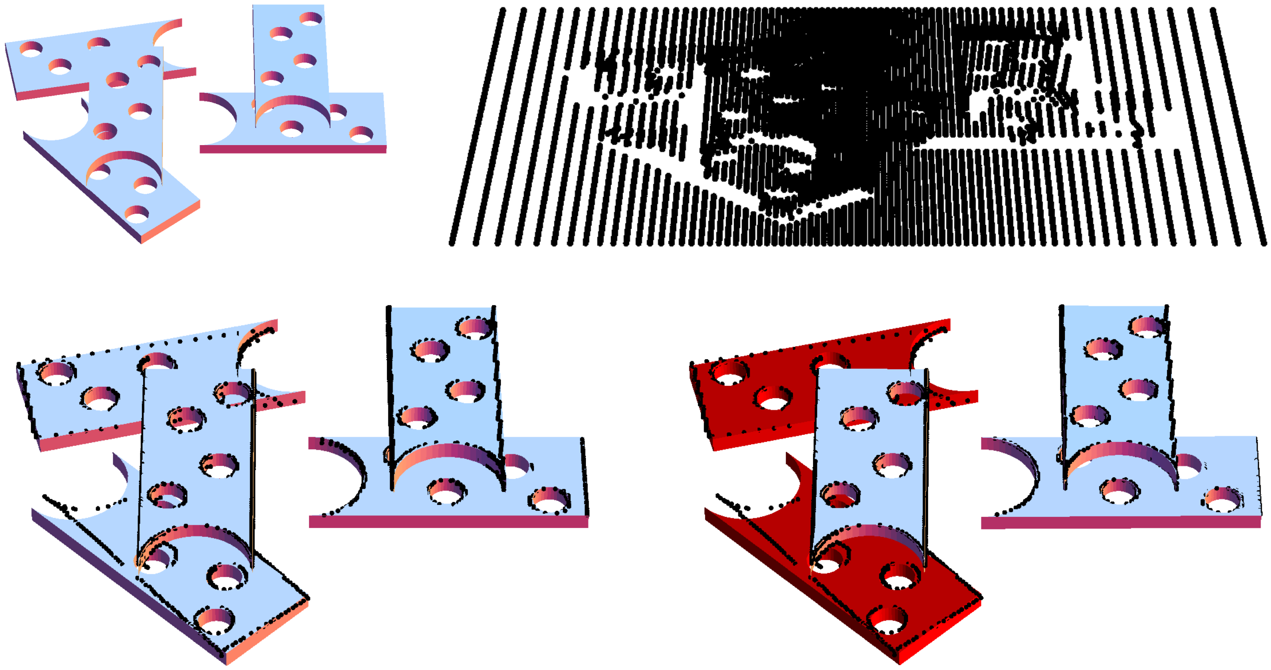

Fig. 12 Object detection and localization with real world objects in 3D scanning data obtained by simulating a SICK LMS 400 scanning device. Due to the use of edge images, wrong edge detections due to noise ("looking through holes") yields Newton's algorithm to fail to converge. While the convex initialization step simultaneously gives proper estimates of the number of objects as well as the corresponding transformations (bottom left), running subsequently the geometric optimization approach fails to converge for the two objects marked red. The reason is that the ability of "looking through holes" complicates the objective function and narrows down the region of attraction to the local minimum.

Further work also includes to work out criteria for selecting the discretization of the pose space automatically. Too coarse discretization yields inaccurate initial pose estimates for the subsequent geometric optimization procedure. Too fine discretization leads to unnecessarily large problem sizes. A convenient feature for the user therefore would be to derive this parameter from given object models directly.

A more straightforward extension concerns the interplay between first- and second-order optimization methods on the manifold of Euclidean transforms in order to optimized the speed of convergence while guaranteeing convergence to a local optimum. As discussed above, the latter sometimes requires to temporarily switch from second- to first-order methods. This objective can be accomplished by adopting numerical trust-region strategies to the manifold setting.

Finally, evaluation of the objective functional proposed in this work is slightly more expensive than related work based on sophisticated extensions of ICP. Here, it is apparent that our approach might benefit from established techniques for accelerating multiple kernel evaluations.

Acknowledgments

The authors would like to thank the VMT Vision Machine Technic Bildverarbeitungssysteme GmbH, a company of the Pepperl+Fuchs Group, for supporting this research work, and to Stefania Petra, Jan Lellmann and Florian Becker from our IPA group for continuously discussing with us various optimization issues.

A Christoffel Symbols Defining the Connection $\overline{\nabla}$

The non-zero Christoffel symbols of (30) are

$$\varGamma^3_{12} = \varGamma^1_{23} = \varGamma^2_{31} = \frac{1}{2}, \tag{44a}$$

$$\varGamma^2_{13} = \varGamma^3_{21} = \varGamma^1_{32} = -\frac{1}{2}, \tag{44b}$$

$$\varGamma^6_{15} = \varGamma^4_{26} = \varGamma^5_{34} = 1, \tag{44c}$$

$$\varGamma^5_{16} = \varGamma^6_{24} = \varGamma^4_{35} = -1. \tag{44d}$$

References

1. R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub. Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA J. Numer. Anal.*, 22(3):359 – 390, 2002.
2. D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recogn.*, 13:111–122, 1981.
3. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:509–522, 2002.

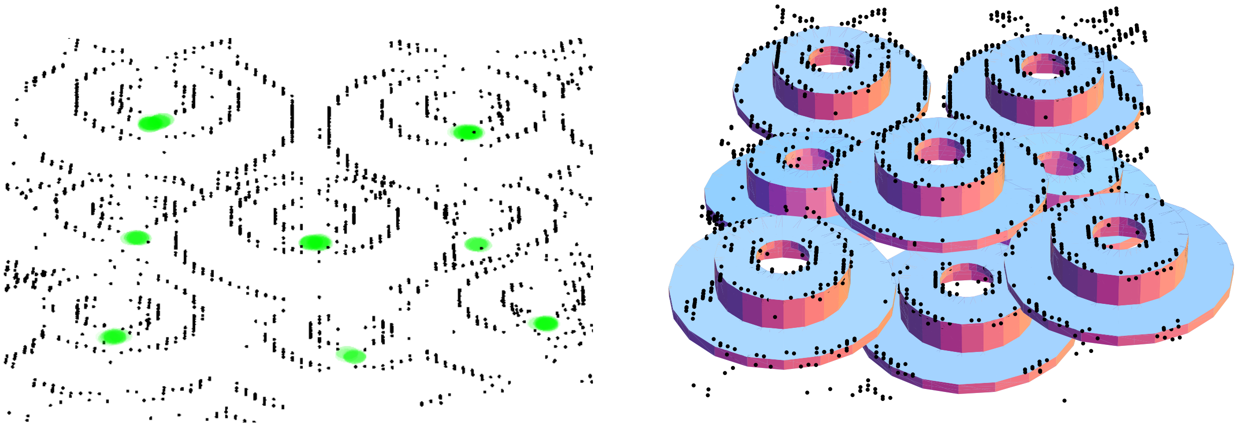

Fig. 13 Object detection and localization with real world 3D scanning data. Pose clusters of all object instances recovered by convex global optimization are displayed as blobs in the left panel. Selecting a representative of each compact cluster as initialization enables to infer the unique number and localization of objects by subsequent geometric optimization (right panel).

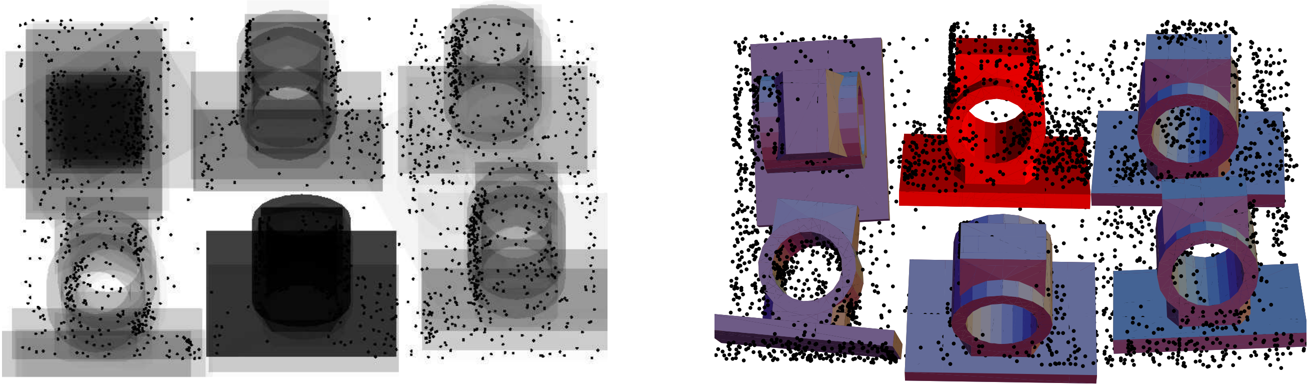

Fig. 14 Detection and localization of complex objects in unstructured range data. Poses returned by the first convex global optimization stage cluster tightly and are displayed in the left panel, where more likely candidate poses are shown less transparent. Selecting a representative of each cluster as initialization enables to accurately locate the objects through subsequent geometric optimization – cf. right panel and the lower panel of Fig. 1. The object marked with red shows an example, where Newton's method failed and switching to first-order optimization is required to ensure convergence.

4. S. Benhimane and E. Malis. A new approach to vision-based robot control with omni-directional cameras. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 526–531, 2006.
5. K. Bennett and E. Parrado-Hernández. The interplay of optimization and machine learning research. *J. Mach. Learning Res.*, 7:1265–1281, 2006.
6. D. Bertsimas and R. Weismantel. *Optimization over Integers*. Dynamic Ideas, 2005.
7. P. J. Besl and R. C. Jain. Three-dimensional object recognition. *ACM Comput. Surv.*, 17(1):75–145, 1985.
8. P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:239–256, 1992.
9. E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.*, 10:1196–1211, 2000.
10. Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. *Int. J. Comp. Vision*, 70(2):109–131, 2006.
11. D. Breitenreicher and C. Schnörr. Intrinsic second-order geometric optimization for robust point set registration without correspondence. In *7th Int. Workshop on Energy Minimization Methods in Comp. Vision and Pattern Recogn.*, 2009.
12. D. Breitenreicher and C. Schnörr. Robust 3D object registration without explicit correspondence using geometric integration. *Machine Vision and Applications*, 21(5):601-611, 2010.
13. T. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, 66(5):1632–1648, 2006.
14. S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.
15. R. T. Chin and C. R. Dyer. Model-based recognition in robot vision. *ACM Comp. Surv.*, 18(1):67–108, 1986.
16. C. S. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *Int. J. Comp. Vision*, 25(1):63–85, 1997.
17. H. Chui and A. Rangarajan A feature registration framework using mixture models *IEEE Workshop Math. Methods in Biomed. Image Anal.*: 190–197, 2000
18. T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.

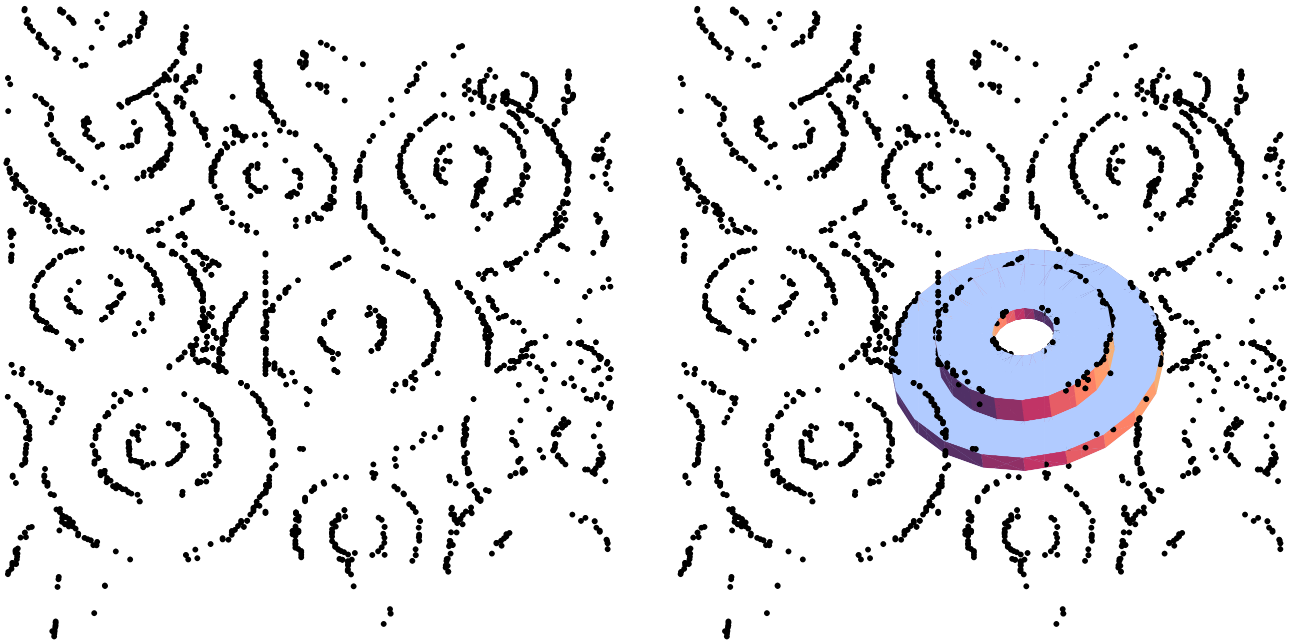

Fig. 15 Robustness of object detection. A single disc that only slightly differs from all other discs (slightly larger inner ring radius) is reliably returned as single object instance by the first convex programming stage.

19. M. P. do Carmo. *Riemannian Geometry*. Birkhäuser Boston, 1992.
20. D. Donoho. Compressed sensing. *IEEE Trans. Information Theory*, 52:1289–1306, 2006.
21. D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable revovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inf. Theory*, 52:6–18, 2006.
22. T. Drummond and R. Cipolla. Real-time tracking of complex structures with on-line camera calibration. *Image Vision Comp.*, 20(5-6):427–433, 2002.
23. A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20:303–353, 1999.
24. 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, 1981.
25. A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image Vision Comp.*, 21(13-14):1145–1153, 2003.
26. A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proc. Europ. Conf. Comp. Vision*, 2004.
27. N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Proc. Symp. Geom. Processing*, 2005.
28. G. Golub and C. Van Loan. *Matrix Computations*. The John Hopkins Univ. Press, 3rd edition, 1996.
29. M. Greenspan. Geometric probing of dense range data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:495–508, 2002.
30. R. I. Hartley and F. Kahl. Global optimization through rotation space search. *Int. J. Comp. Vision*, 82(1):64–79, 2009.
31. B. Jian and B. C. Vemuri. A robust algorithm for point set registration using mixture of Gaussians. In *Proc. Int. Conf. Comp. Vision*, 2005.
32. A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21:433–449, 1999.
33. S. Krishnan, P. Y. Lee, J. B. Moore, and S. Venkatasubramanian. Optimisation-on-a-manifold for global registration of multiple 3D point sets. *Int. J. Intell. Syst. Technol. Appl.*, 3(3/4):319 – 340, 2007.
34. I. Lavva, E. Hameiri, and I. Shimshoni. Robust methods for geometric primitive recovery and estimation from range images. *IEEE Trans. Syst. Man and Cybernetics - Part B - Cybernetics*, 38(3):826–845, 2008.
35. H. Li and R. Hartley. The 3D-3D registration problem revisited. In *Proc. Int. Conf. Comp. Vision*, 2007.
36. Y. Matsushima. *Differentiable Manifolds*. Marcel Dekker, INC. New York, 1972.
37. N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas. Registration of point cloud data from a geometric optimization perspective. In *Proc. Sym. Geom. Process.*, 2004.
38. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24:227–234, 1995.
39. Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.
40. C. Olson. Efficient pose clustering using a randomized algorithm. *Int. J. Comp. Vision*, 23(2):131–147, 1997.
41. C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for Euclidean registration problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):783–794, 2009.
42. H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu. Geometry and convergence analysis of algorithms for registration of 3D shapes. *Int. J. Comp. Vision*, 67(3):277–296, 2006.
43. A. Rangarajan, H. Chui, and F. L. Bookstein. The softassign procrustes matching algorithm. In *Proc. Int. Conf. Inf. Process. Med. Imaging*, 1997.
44. L. Reyes, G. Medioni, and E. Bayro. Registration of 3D points using geometric algebra and tensor voting. *Int. J. Comp. Vision*, 75(3):351–369, 2007.
45. R. Rockafellar and R.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der math. Wissenschaften*. Springer, 1998.
46. S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. 3rd Int. Conf. on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

47. J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comp.*, 25:578–596, 2007.
48. L. Shang and M. Greenspan. Pose determination by potential well space embedding. In *Proc. 6th Int. Conf. 3-D Digital Imaging and Modeling*, pages 297–304, 2007. IEEE Computer Society.
49. Q. Shi, N. Xi, Y. Chen, and W. Sheng. Registration of point clouds for 3D shape inspection. In *Int. Conf. Intell. Robots Syst.*, 2006.
50. R. Subbarao and P. Meer. Nonlinear mean shift over Riemannian manifolds. *Int. J. Comp. Vision*, 84:1–20, 2009.
51. C. J. Taylor and D. J. Kriegman. Minimization on the Lie group SO(3) and related manifolds. Technical Report 9405, Center for Systems Science, Dept. of Electrical Engineering, Yale University, 1994.
52. M. Teboulle. A unified continuous optimization framework for center-based clustering methods. *J. Mach. Learn. Res.*, 8:65–102, 2007.
53. J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Trans. Inf. Theory*, 52:1030–1051, 2006.
54. Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *Proc. Europ. Conf. Comp. Vision*, volume III, pages 558–569, 2004.
55. M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learning*, 1(1-2):1–305, 2008.
56. F. Wang, B. C. Vemuri, A. Rangarajan, I. M. Schmalfuss, and S. J. Eisenschenk. Simultaneous nonrigid registration of multiple point sets and atlas construction. In *Proc. Europ. Conf. Comp. Vision*, 2006.
57. C. Wiedemann, M. Ulrich, and C. Steger. Recognition and tracking of 3D objects. In *Pattern Recogn.*, 2008.
58. H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Comp. Sci. Eng.*, 4:10–21, 1997.
59. L. Zhu, J. Barhak, V. Shrivatsan, and R. Katz. Efficient registration for precision inspection of free-form surfaces. *Int. J. Adv. Manuf. Technol.*, 32:505–515, 2007.