Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration

Jiaolong Yang, *Student Member, IEEE*, Hongdong Li, *Member, IEEE*,
Dylan Campbell, *Student Member, IEEE*, and Yunde Jia, *Member, IEEE*

Abstract—The Iterative Closest Point (ICP) algorithm is one of the most widely used methods for point-set registration. However, being based on local iterative optimization, ICP is known to be susceptible to local minima. Its performance critically relies on the quality of the initialization and only local optimality is guaranteed. This paper presents the first globally optimal algorithm, named Go-ICP, for Euclidean (rigid) registration of two 3D point-sets under the L_2 error metric defined in ICP. The Go-ICP method is based on a branch-and-bound scheme that searches the entire 3D motion space $SE(3)$. By exploiting the special structure of $SE(3)$ geometry, we derive novel upper and lower bounds for the registration error function. Local ICP is integrated into the BnB scheme, which speeds up the new method while guaranteeing global optimality. We also discuss extensions, addressing the issue of outlier robustness. The evaluation demonstrates that the proposed method is able to produce reliable registration results regardless of the initialization. Go-ICP can be applied in scenarios where an optimal solution is desirable or where a good initialization is not always available.

Index Terms—3D point-set registration, global optimization, branch-and-bound, *SE(3)* space search, iterative closest point

✦

1 INTRODUCTION

POINT-SET registration is a fundamental problem in computer and robot vision. Given two sets of points in different coordinate systems, or equivalently in the same coordinate system with different poses, the goal is to find the transformation that best aligns one of the point-sets to the other. Point-set registration plays an important role in many vision applications. Given multiple partial scans of an object, it can be applied to merge them into a complete 3D model [1], [2]. In object recognition, fitness scores of a query object with respect to existing model objects can be measured with registration results [3], [4]. In robot navigation, localization can be achieved by registering the current view into the global environment [5], [6]. Given cross-modality data acquired from different sensors with complementary information, registration can be used to fuse the data [7], [8] or determine the relative poses between these sensors [9], [10].

Among the numerous registration methods proposed in literature, the Iterative Closest Point (ICP) algorithm [11], [12], [13], introduced in the early 1990s, is the most well-known algorithm for efficiently registering two 2D or 3D point-sets under Euclidean (rigid) transformation. Its concept is simple and intuitive: given an initial transformation (rotation and translation), it alternates between building closest-point correspondences under the current transformation and estimating the transformation with these correspondences, until convergence. Appealingly, point-to-point ICP is able to work directly on the raw point-sets, regardless of their intrinsic properties (such as distribution, density and noise level). Due to its conceptual simplicity, high usability and good performance in practice, ICP and its variants are very popular and have been successfully applied in numerous real-world tasks ([7], [14], [15], for example).

However, ICP is also known for its susceptibility to the problem of local minima, due to the non-convexity of the problem as well as the local iterative procedure it adopts. Being an iterative method, it requires a good initialization, without which the algorithm may easily become trapped in a local minimum. If this occurs, the solution may be far from the true (optimal) solution, resulting in erroneous estimation. More critically, there is no reliable way to tell whether or not it is trapped in a local minimum.

To deal with the issue of local minima, previous efforts have been devoted to widening the basin of convergence [16], [17], performing heuristic and non-deterministic global search [18], [19] and utilizing other methods for coarse initial alignment [20], [21], etc. However, global optimality cannot be guaranteed with these approaches. Furthermore, some methods, such as those based on feature matching, are not always reliable or even applicable when the point-sets are not sampled densely from smooth surfaces.

This work is, to the best of our knowledge, the first to propose a globally optimal solution to the Euclidean registration problem defined by ICP in 3D. The proposed method always produces the exact and globally optimal solution, up to the desired accuracy. Our method is named the *Globally Optimal ICP*, abbreviated to *Go-ICP*.

We base the Go-ICP method on the well-established branch-and-bound (BnB) theory for global optimization.

- J. Yang is with the Beijing Lab of Intelligent Information Technology, Beijing Institute of Technology, Beijing, China, and with the Australian National University, Canberra, ACT 0200, Australia.
  E-mail: yangjiaolong@bit.edu.cn.
- H. Li and D. Campbell are with the Australian National University, Canberra, ACT 0200, Australia, and with the National ICT, Australia.
  E-mail: {hongdong.li, dylan.campbell}@anu.edu.au.
- Y. Jia is with the Beijing Lab of Intelligent Information Technology, Beijing Institute of Technology, Beijing, China. E-mail: jiayunde@bit.edu.cn.

Nevertheless, choosing a suitable domain parametrization for building a tree structure in BnB and, more importantly, deriving efficient error bounds based on the parametrization are both non-trivial. Our solution is inspired by the $SO(3)$ space search technique proposed in [22] as well as [23]. We extend it to $SE(3)$ space search and derive novel bounds of the 3D registration error. Another feature of the Go-ICP method is that we employ, as a subroutine, the conventional (local) ICP algorithm within the BnB search procedure. The algorithmic structure of the proposed method can be summarized as follows.

Use BnB to search the space of $SE(3)$
    Whenever a better solution is found, call ICP initialized at this solution to refine (reduce) the objective function value. Use ICP's result as an updated upper bound to continue the BnB.
Until convergence.

Our error metric strictly follows that of the original ICP algorithm, that is, minimizing the L_2 norm of the closest-point residual vector. We also show how a trimming strategy can be utilized to handle outliers. With small effort, one can also extend the method with robust kernels or robust norms. A preliminary version of this work was presented as a conference paper [24].

1.1 Previous Work

There is a large volume of work published on ICP and other registration techniques, precluding us from giving a comprehensive list. Therefore, we will focus below on some relevant Euclidean registration works addressing the local minimum issue in 2D or 3D. For other papers, the reader is referred to two surveys on ICP variants [6], [25], a recent survey on 3D point cloud and mesh registration [26], an overview of 3D registration [27] and the references therein.

1.1.1   Robustified Local Methods

To improve the robustness of ICP to poor initializations, previous work has attempted to enlarge the basin of convergence by smoothing out the objective function. Fitzgibbon [16] proposed the LM-ICP method where the ICP error was optimized with the Levenberg–Marquardt algorithm [28]. Better convergence than ICP was observed, especially with the use of robust kernels.

It was shown by Jian and Vemuri [29] that if the point-sets are represented with Gaussian mixture models (GMMs), ICP is related to minimizing the Kullback-Leibler divergence of two GMMs. Although improved robustness to outliers and poor initializations could be achieved by GMM-based techniques [17], [29], [30], [31], the optimization was still based on local search. Earlier than these works, Rangarajan et al. [32] presented a SoftAssign algorithm which assigned Gaussian weights to the points and applied deterministic annealing on the Gaussian variance. Granger and Pennec [33] proposed an algorithm named Multi-scale EM-ICP where an annealing scheme on GMM variance was also used. Biber and Straßer [34] developed the normal distributions transform (NDT) method, where Gaussian models were defined for uniform cells in a spatial grid. Magnusson et al. [35] experimentally showed that NDT was more robust to poor initial alignments than ICP.

Some methods extend ICP by robustifying the distance between points. For example, Sharp et al. [36] proposed the additional use of invariant feature descriptor distance; Johnson and Kang [37] exploited color distances to boost the performance.

1.1.2   Global Methods

To address the local minima problem, global registration methods have also been investigated. A typical family adopts stochastic optimization such as genetic algorithms [19], [38], particle swam optimization [39], particle filtering [18] and simulated annealing schemes [1], [40]. While the local minima issue is effectively alleviated, global optimality cannot be guaranteed and initializations still need to be reasonably good as otherwise the parameter space is too large for the heuristic search.

Another class of global registration methods introduces shape descriptors for coarse alignment. Local descriptors, such as spin images [3], shape contexts [4], integral volume [41] and point feature histograms [20] are invariant under specific transformations. They can be used to build sparse feature correspondences, based on which the best transformation can be found with random sampling [20], greedy algorithms [3], Hough Transforms [42] or BnB algorithms [41], [43]. Global shape descriptors, such as extended Gaussian images (EGI) [21], can be used to find the best transformation maximizing descriptor correlation. These methods are often robust and can efficiently register surfaces where the descriptor can be readily computed.

Random sampling schemes such as RANSAC [44] can also be used to register raw point clouds directly. Irani and Raghavan [45] randomly sampled 2-point bases to align 2D point-sets using similarity transformations. For 3D, Aiger et al. [46] proposed a 4PCS algorithm that sampled coplanar 4-points, since congruent coplanar four-point sets can be efficiently extracted with affine invariance.

1.1.3   Globally Optimal Methods

Registration methods that guarantee optimality have been published in the past, albeit in a smaller number. Most of them are based on BnB algorithms. For example, geometric BnB has been used for 2D image pattern matching [47], [48], [49]. These methods share a similar structure with ours: given each transformation sub-domain, determine for each data point the uncertainty region, based on which the objective function bounds are derived and the BnB search is applied. However, despite uncertainty region computation with various 2D transformations has been extensively explored, extending them to 3D is often impractical due to the heightened complexity [47].

For 3D registration, Li and Hartley [23] proposed using a Lipschitzized L_2 error function that was minimized by BnB. However, this method makes unrealistic assumptions that the two point-sets are of equal size and that the transformation is pure rotation. Olsson et al. [50] obtained the optimal solution to simultaneous point-to-point, point-to-line and point-to-plane registration using BnB and bilinear relaxation of rotation quaternions. This method, although related to ours, requires known correspondences. Recently, Bustos et al. [51] proposed searching $SO(3)$ space for optimal 3D

geometric matching, assuming known translation. Efficient run-times were achieved using stereographic projection techniques.

Some optimal 3D registration methods assume a small number of putative correspondences, and treat registration as a correspondence outlier removal problem. For example, to minimize the overall pairwise distance error, Gelfand et al. [41] applied BnB to assign one best corresponding model point for each data point. A similar idea using pairwise consistency was proposed by Enqvist et al. [52], where the inlier-set maximization was formulated as an NP-hard graph vertex cover problem and solved using BnB. Using angular error, Bazin et al. [43] solved a similar correspondence inlier-set maximization problem via $SO(3)$ space search assuming known translation. Enqvist and Kahl [53] optimally solved camera pose in $SE(3)$ via BnB. However, the key insight is that with pre-matched correspondences, their pairwise constraint (also used in [52]) enabled a single translation BnB in $\mathbb{R}^3$ to solve the $SE(3)$ problem.

In this paper, we optimally solve the 3D Euclidean registration problem with both rotation and translation. The proposed Go-ICP method is able to work directly on raw sparse or dense point-sets (which may be sub-sampled only for reasons of efficiency), without the need for a good initialization or putative correspondences. The method is related to the idea of $SO(3)$ space search, as proposed in [22], [23] and extended in [43], [54], [55], etc. We extend the 3-dimensional $SO(3)$ search to six-dimensional $SE(3)$ search, which is much more challenging.

2 PROBLEM FORMULATION

In this paper we define the L_2-norm registration problem in the same way as in the standard point-to-point ICP algorithm. Let two 3D point-sets $\mathcal{X} = \{\mathbf{x}_i\}, i = 1, \ldots, N$ and $\mathcal{Y} = \{\mathbf{y}_j\}, j = 1, \ldots, M$, where $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^3$ are point coordinates, be the *data* point-set and the *model* point-set respectively. The goal is to estimate a rigid motion with rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$, which minimizes the following L_2-error E:

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} e_i(\mathbf{R}, \mathbf{t})^2 = \sum_{i=1}^{N} \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_{j^*}\|^2, \quad (1)$$

where $e_i(\mathbf{R}, \mathbf{t})$ is the per-point residual error for $\mathbf{x}_i$. Given $\mathbf{R}$ and $\mathbf{t}$, the point $\mathbf{y}_{j^*} \in \mathcal{Y}$ is denoted as the optimal correspondence of $\mathbf{x}_i$, which is the closest point to the transformed $\mathbf{x}_i$ in $\mathcal{Y}$, i.e.,

$$j^* = \underset{j \in \{1, \ldots, M\}}{\operatorname{argmin}} \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_j\|. \quad (2)$$

Note the short-hand notation used here: j^* varies as a function of $(\mathbf{R}, \mathbf{t})$ and also depends on $\mathbf{x}_i$.

Equations (1) and (2) actually form a well-known *chicken-and-egg* problem: if the true correspondences are known *a priori*, the transformation can be optimally solved in closed-form [56], [57]; if the optimal transformation is given, correspondences can also be readily found. However, the joint problem cannot be trivially solved. Given an initial transformation $(\mathbf{R}, \mathbf{t})$, ICP iteratively solves the problem by
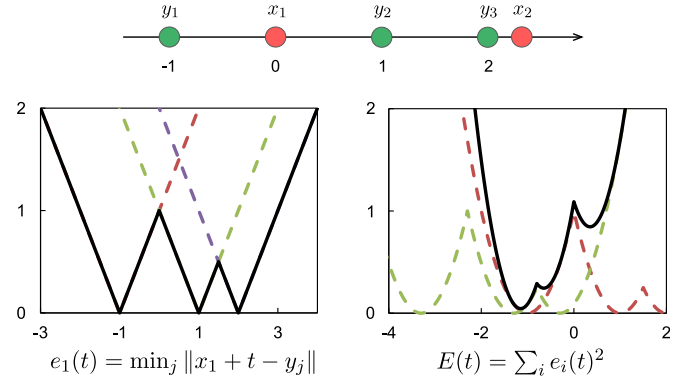

Fig. 1. Nonconvexity of the registration problem. *Top*: two 1D point-sets $\{x_1, x_2\}$ and $\{y_1, y_2, y_3\}$. *Bottom-left*: residual error (closest-point distance) for x_1 as a function of translation t; the three dashed curves are $\|x_1 + t - y_j\|$ with $j=1, 2, 3$ respectively. *Bottom-right*: the overall L_2 registration error; the two dashed curves are $e_i(t)^2$ with $i=1, 2$ respectively. The residual error functions are nonconvex, thus the L_2 error function is also nonconvex.

alternating between estimating the transformation with (1), and finding closest-point matches with (2). Such an iterative scheme guarantees convergence to a local minimum [11].

2.1 (Non-)Convexity Analysis

It is easy to see from (1) that the transformation function denoted by $T_x(p)$ affinely transforms a point x with parameters p, thus the residual function $e(p) = d(T_x(p))$ is convex provided that *domain D_p is a convex set (Condition 1)* and $d(x) = \inf_{y \in \mathcal{Y}} \|x - y\|$ is convex. Moreover, it has been shown in [58] and further in [50] that $d(x)$ is convex if and only if $\mathcal{Y}$ *is a convex set (Condition 2)*. For registration with pure translation, Condition 1 can be satisfied as the domain D_p is $\mathbb{R}^3$. However, $\mathcal{Y}$ is often a discrete point-set sampled from complex surfaces and is thus rarely a convex set, violating Condition 2. Therefore, $e(p)$ is nonconvex. Fig. 1 shows a 1D example. For registration with rotation, even Condition 1 cannot be fulfilled, as the rotation space induced by the quadratic orthogonality constraints $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ is clearly not a convex set.

2.2 Outlier Handling

As is well known, L_2-norm least squares fitting is susceptible to outliers. A small number of outliers may lead to erroneous registration, even if the global optimum is achieved. There are many strategies to deal with outliers [16], [25], [29], [59], [60]. In this paper, a trimmed estimator is used to gain outlier robustness similar to [60]. To streamline the presentation and mathematical derivation, we defer the discussion to Section 5.3. For now we assume there are no outliers and focus on minimizing (1).

3 THE BRANCH AND BOUND ALGORITHM

The BnB algorithm is a powerful global optimization technique that can be used to solve nonconvex and NP-hard problems [61]. Although existing BnB methods work successfully for 2D registration, extending them to search $SE(3)$ and solve 3D rigid registration has been much more challenging [23], [47]. In order to apply BnB to 3D registration, we must consider i) how to parametrize and branch

the domain of 3D motions (Section 3.1), and *ii*) how to efficiently find upper bounds and lower bounds (Section 4).

3.1 Domain Parametrization

Recall that our goal is to minimize the error E in (1) over the domain of all feasible 3D motions (the $SE(3)$ group, defined by $SE(3) = SO(3) \times \mathbb{R}^3$). Each member of $SE(3)$ can be minimally parameterized by 6 parameters (3 for rotation and 3 for translation).

Using the *angle-axis representation*, each rotation can be represented as a 3D vector $\mathbf{r}$, with axis $\mathbf{r}/\|\mathbf{r}\|$ and angle $\|\mathbf{r}\|$. We use $\mathbf{R_r}$ to denote the corresponding rotation matrix for $\mathbf{r}$. The 3×3 matrix $\mathbf{R_r} \in SO(3)$ can be obtained by the matrix exponential map as

$$\mathbf{R_r} = \exp([\mathbf{r}]_\times) = \mathbf{I} + \frac{[\mathbf{r}]_\times \sin\|\mathbf{r}\|}{\|\mathbf{r}\|} + \frac{[\mathbf{r}]_\times^2 (1 - \cos\|\mathbf{r}\|)}{\|\mathbf{r}\|^2}, \quad (3)$$

where $[\cdot]_\times$ denotes the skew-symmetric matrix representation

$$[\mathbf{r}]_\times = \begin{bmatrix} 0 & -r^3 & r^2 \\ r^3 & 0 & -r^1 \\ -r^2 & r^1 & 0 \end{bmatrix}, \quad (4)$$

where r^i is the ith element in $\mathbf{r}$. Equation (3) is also known as the *Rodrigues' rotation formula* [62]. The inverse map is given by the matrix logarithm as

$$[\mathbf{r}]_\times = \log \mathbf{R_r} = \frac{\|\mathbf{r}\|}{2 \sin\|\mathbf{r}\|}(\mathbf{R_r} - \mathbf{R_r^T}), \quad (5)$$

where $\|\mathbf{r}\| = \arccos\left((\text{trace}(\mathbf{R_r}) - 1)/2\right)$. With the angle-axis representation, the entire 3D rotation space can be compactly represented as a solid radius-π ball in $\mathbb{R}^3$. Rotations with angles less than (or, equal to) π have unique (or, two) corresponding angle-axis representations on the interior (or, surface) of the ball. For ease of manipulation, we use the minimum cube $[-\pi, \pi]^3$ that encloses the π-ball as the rotation domain.

For the translation part, we assume that the optimal translation lies within a bounded cube $[-\xi, \xi]^3$, which may be readily set by choosing a large number for ξ.

During BnB search, initial cubes will be subdivided into smaller sub-cubes C_r, C_t using the *octree data-structure* and the process is repeated. Fig. 2 illustrates our domain parametrization.

4 BOUNDING FUNCTION DERIVATION

For our 3D registration problem, we need to find the bounds of the L_2-norm error function used in ICP within a domain $C_r \times C_t$. Next, we will introduce the concept of an *uncertainty radius* as a mathematical preparation, then derive our bounds based on it.

4.1 Uncertainty Radius

Intuitively, we want to examine the uncertainty region of a 3D point $\mathbf{x}$ perturbed by an arbitrary rotation $\mathbf{r} \in C_r$ or a translation $\mathbf{t} \in C_t$. We aim to find a ball, characterized by an uncertainty radius, that encloses such an uncertainty region. We will use the first two lemmas of [63] in the following
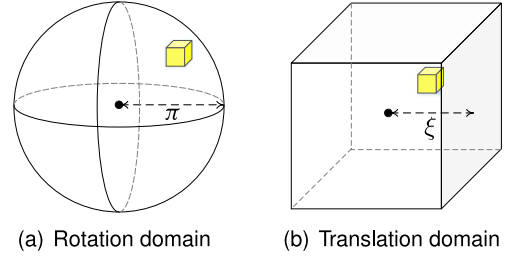

(a) Rotation domain   (b) Translation domain

Fig. 2. $SE(3)$ space parameterization for BnB. *Left*: the rotation space $SO(3)$ is parameterized in a solid radius-π ball with the angle-axis representation. *Right*: the translation is assumed to be within a 3D cube $[-\xi, \xi]^3$ where ξ can be readily set. The octree data-structure is used to divide (branch) the domains and the yellow box in each diagram represents a sub-cube.

derivation. For convenience, we summarize both lemmas in a single Lemma shown below.

Lemma 1. *For any vector $\mathbf{x}$ and two rotations $\mathbf{R_r}$ and $\mathbf{R_{r_0}}$ with $\mathbf{r}$ and $\mathbf{r_0}$ as their angle-axis representations, we have*

$$\angle(\mathbf{R_r x}, \mathbf{R_{r_0} x}) \leqslant \angle(\mathbf{R_r}, \mathbf{R_{r_0}}) \leqslant \|\mathbf{r} - \mathbf{r_0}\|, \quad (6)$$

where $\angle(\mathbf{R_r}, \mathbf{R_{r_0}}) = \arccos\left((\text{trace}(\mathbf{R_r^T R_{r_0}}) - 1)/2\right)$ *is the angular distance between rotations.*

The second inequality in (6) means that the angular distance between two rotations on the $SO(3)$ manifold is less than the Euclidean vector distance of their angle-axis representations in $\mathbb{R}^3$. Based on this Lemma, uncertainty radii are given as follows.

Theorem 1 (Uncertainty radius). *Given a 3D point $\mathbf{x}$, a rotation cube C_r of half side-length σ_r with $\mathbf{r_0}$ as the center and examining the maximum distance from $\mathbf{R_r x}$ to $\mathbf{R_{r_0} x}$, we have $\forall \mathbf{r} \in C_r$,*

$$\|\mathbf{R_r x} - \mathbf{R_{r_0} x}\| \leqslant 2\sin\left(\min(\sqrt{3}\sigma_r/2, \pi/2)\right)\|\mathbf{x}\| \doteq \gamma_r. \quad (7)$$

Similarly, given a translation cube C_t with half side-length σ_t centered at $\mathbf{t_0}$, we have $\forall \mathbf{t} \in C_t$,

$$\|(\mathbf{x} + \mathbf{t}) - (\mathbf{x} + \mathbf{t_0})\| \leqslant \sqrt{3}\sigma_t \doteq \gamma_t. \quad (8)$$

Proof. Inequality (7) can be derived from

$$\|\mathbf{R_r x} - \mathbf{R_{r_0} x}\| \quad (9)$$

$$= 2\sin\left(\angle(\mathbf{R_r x}, \mathbf{R_{r_0} x})/2\right)\|\mathbf{x}\| \quad (10)$$

$$\leqslant 2\sin\left(\min(\angle(\mathbf{R_r}, \mathbf{R_{r_0}})/2, \pi/2)\right)\|\mathbf{x}\| \quad (11)$$

$$\leqslant 2\sin\left(\min(\|\mathbf{r} - \mathbf{r_0}\|/2, \pi/2)\right)\|\mathbf{x}\| \quad (12)$$

$$\leqslant 2\sin\left(\min(\sqrt{3}\sigma_r/2, \pi/2)\right)\|\mathbf{x}\| \quad (13)$$

where (10) is illustrated in Fig. 3. Inequalities (11), (12) are based on Lemma 1, and (13) is from the fact that $\mathbf{r}$ resides in the cube.

Inequality (8) can be trivially derived via $\|(\mathbf{x} + \mathbf{t}) - (\mathbf{x} + \mathbf{t_0})\| = \|\mathbf{t} - \mathbf{t_0}\| \leqslant \sqrt{3}\sigma_t$. □

We call γ_r the rotation uncertainty radius, and γ_t the translation uncertainty radius. They are depicted in Fig. 4. Note that γ_r is point-dependent, thus we use γ_{ri} to denote the rotation uncertainty radius at $\mathbf{x}_i$ and the vector $\boldsymbol{\gamma}_r$ to
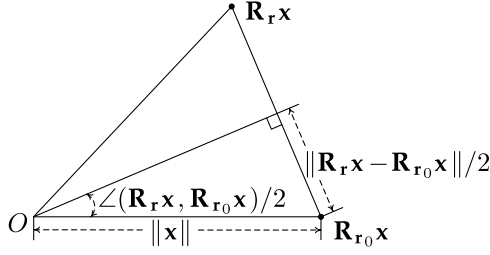
Fig. 3. Distance computation from $\mathbf{R_r x}$ to $\mathbf{R_{r_0} x}$ used in the derivation of the rotation uncertainty radius.

Fig. 4. Uncertainty radii at a point. *Left:* rotation uncertainty ball for C_r (in red) with center $\mathbf{R_{r_0} x}$ (blue dot) and radius γ_r. *Right:* translation uncertainty ball for C_t (in red) with center $\mathbf{x} + \mathbf{t}_0$ (blue dot) and radius γ_t. In both diagrams, the uncertainty balls enclose the range of $\mathbf{R_r x}$ or $\mathbf{x} + \mathbf{t}$ (in green).

represent all γ_{ri}. Based on the uncertainty radii, the bounding functions are derived in the following section.

4.2 Bounding the L_2 Error

Given a rotation cube C_r centered at $\mathbf{r}_0$ and a translation cube C_t centered at $\mathbf{t}_0$, we will first derive valid bounds of the residual $e_i(\mathbf{R}, \mathbf{t})$ for a single point $\mathbf{x}_i$.

The upper bound of e_i can be easily chosen by evaluating the error at any $(\mathbf{r}, \mathbf{t}) \in C_r \times C_t$. Finding a suitable lower bound for the L_2 error is a harder task. From Section 4.1 we know that, with rotation $\mathbf{r} \in C_r$ (or, translation $\mathbf{t} \in C_t$), a transformed point $\mathbf{x}_i$ will lie in the uncertainty ball centered at $\mathbf{R_{r_0} x}_i$ (or, $\mathbf{x}_i + \mathbf{t}_0$) with radius γ_{ri} (or, γ_t). For both rotation and translation, it therefore lies in the uncertainty ball centered at $\mathbf{R_{r_0} x}_i + \mathbf{t}_0$ with radius $\gamma_{ri} + \gamma_t$. Now we need to consider the smallest residual error that is possible for $\mathbf{x}_i$. We have the following theorem, which is the cornerstone of the proposed method.

Theorem 2 (Bounds of per-point residuals). *For a 3D motion domain $C_r \times C_t$ centered at $(\mathbf{r}_0, \mathbf{t}_0)$ with uncertainty radii γ_{ri} and γ_t, the upper bound $\overline{e_i}$ and the lower bound $\underline{e_i}$ of the optimal registration error $e_i(\mathbf{R_r}, \mathbf{t})$ at $\mathbf{x}_i$ can be chosen as*

$$\overline{e_i} \doteq e_i(\mathbf{R_{r_0}}, \mathbf{t}_0), \tag{14}$$

$$\underline{e_i} \doteq \max\big(e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{ri} + \gamma_t), 0\big). \tag{15}$$

Proof. The validity of $\overline{e_i}$ is obvious: error e_i at the specific point $(\mathbf{r}_0, \mathbf{t}_0)$ must be larger than the minimal error within the domain, i.e. $e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) \geqslant \min_{\forall (\mathbf{r}, \mathbf{t}) \in (C_r \times C_t)} e_i(\mathbf{R_r}, \mathbf{t})$. We now focus on proving the correctness of $\underline{e_i}$.

As defined in (2), the model point $\mathbf{y}_{j^*} \in \mathcal{Y}$ is closest to $(\mathbf{R_r x}_i + \mathbf{t})$. Let $\mathbf{y}_{j_0^*}$ be the closest model point to $\mathbf{R_{r_0} x}_i + \mathbf{t}_0$. Observe that, $\forall (\mathbf{r}, \mathbf{t}) \in (C_r \times C_t)$,

$$e_i(\mathbf{R_r}, \mathbf{t}) = \|\mathbf{R_r x}_i + \mathbf{t} - \mathbf{y}_{j^*}\| \tag{16}$$

$$= \|(\mathbf{R_{r_0} x}_i + \mathbf{t}_0 - \mathbf{y}_{j^*}) + (\mathbf{R_r x}_i - \mathbf{R_{r_0} x}_i) + (\mathbf{t} - \mathbf{t}_0)\| \tag{17}$$

$$\geqslant \|\mathbf{R_{r_0} x}_i + \mathbf{t}_0 - \mathbf{y}_{j^*}\| - (\|\mathbf{R_r x}_i - \mathbf{R_{r_0} x}_i\| + \|\mathbf{t} - \mathbf{t}_0\|) \tag{18}$$

$$\geqslant \|\mathbf{R_{r_0} x}_i + \mathbf{t}_0 - \mathbf{y}_{j^*}\| - (\gamma_{ri} + \gamma_t) \tag{19}$$

$$\geqslant \|\mathbf{R_{r_0} x}_i + \mathbf{t}_0 - \mathbf{y}_{j_0^*}\| - (\gamma_{ri} + \gamma_t) \tag{20}$$

$$= e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{ri} + \gamma_t), \tag{21}$$

where (17) trivially involves introducing two auxiliary elements $\mathbf{R_{r_0} x}$ and $\mathbf{t}_0$, (18) follows from the reverse triangle inequality[1], (19) is based on the uncertainty radii in (7) and (8), and (20) is from the closest-point definition.

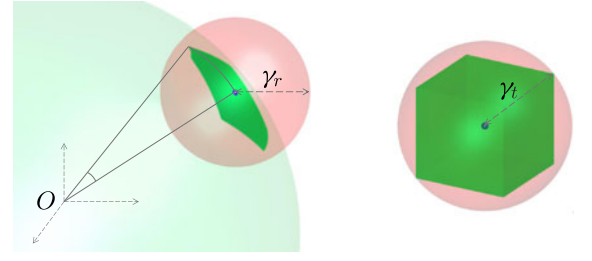1. $|x + y| = |x - (-y)| \geqslant |x| - |-y| = |x| - |y|$

Note that $\mathbf{y}_{j^*}$ is not fixed, but changes dynamically as a function of $(\mathbf{R_r}, \mathbf{t})$ as defined in (2).

According to the above derivation, the residual error $e_i(\mathbf{R_r}, \mathbf{t})$ after perturbing a data point $\mathbf{x}_i$ by a 3D rigid motion composed of a rotation $\mathbf{r} \in C_r$ and a translation $\mathbf{t} \in C_t$ will be at least $e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{ri} + \gamma_t)$. Given that a closest point distance should be non-negative, a valid lower bound $\underline{e_i}$ for $C_r \times C_t$ is $\max\big(e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{ri} + \gamma_t), 0\big) \leqslant \min_{\forall(\mathbf{r},\mathbf{t}) \in (C_r \times C_t)} e_i(\mathbf{R_r}, \mathbf{t})$. $\square$

The geometric explanation for $\underline{e_i}$ is as follows. Since $\mathbf{y}_{j_0^*}$ is closest to the center $\mathbf{R_{r_0} x}_i + \mathbf{t}_0$ of the uncertainty ball with radius $\gamma = \gamma_{ri} + \gamma_t$, it is also closest to the surface of the ball and $\underline{e_i}$ is the closest distance between point-set $\mathcal{Y}$ and the ball. Thus, no matter where the transformed data point $\mathbf{R_r x}_i + \mathbf{t}$ lies inside the ball, its closest distance to point-set $\mathcal{Y}$ will be no less than $\underline{e_i}$. See Fig. 5 for a geometric illustration.

Summing the squared upper and lower bounds of per-point residuals in (14) and (15) for all M points, we get the L_2-error bounds in the following corollary.

Corollary 3 (Bounds of L_2 error). *For a 3D motion domain $C_r \times C_t$ centered at $(\mathbf{r}_0, \mathbf{t}_0)$ with uncertainty radii γ_{ri} and γ_t, the upper bound $\overline{E}$ and the lower bound $\underline{E}$ of the optimal L_2 registration error E^* can be chosen as*

$$\overline{E} \doteq \sum_{i=1}^{M} \overline{e_i}^2 = \sum_{i=1}^{M} e_i(\mathbf{R_{r_0}}, \mathbf{t}_0)^2, \tag{22}$$

$$\underline{E} \doteq \sum_{i=1}^{M} \underline{e_i}^2 = \sum_{i=1}^{M} \max\big(e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{ri} + \gamma_t), 0\big)^2. \tag{23}$$
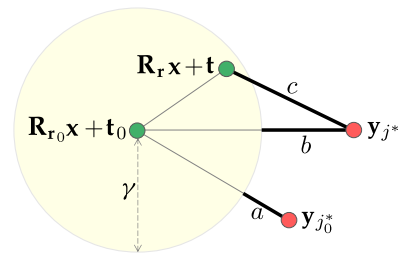

Fig. 5. Deriving the lower bound. Any transformed data point $\mathbf{R_r x} + \mathbf{t}$ lies within the uncertainty ball (in yellow) centered at $\mathbf{R_{r_0} x} + \mathbf{t}_0$ with radius $\gamma = \gamma_r + \gamma_t$. Model points $\mathbf{y}_{j^*}$ and $\mathbf{y}_{j_0^*}$ are closest to $\mathbf{R_r x} + \mathbf{t}$ and $\mathbf{R_{r_0} x} + \mathbf{t}_0$ respectively. It is clear that $a \leq b \leq c$ where $a = \underline{e_i}$ and $c = e_i(\mathbf{R_r}, \mathbf{t})$. See text for more details.

5   THE GO-ICP ALGORITHM

Now that the domain parametrization and bounding functions have been specified, we are ready to present the Go-ICP algorithm concretely.

5.1   Nested BnBs

Given Corollary 3, a direct 6D space BnB (i.e. branching each 6D cube into $2^6 = 64$ sub-cubes and bounding errors for them) seems to be straightforward. However, we find it prohibitively inefficient and memory consuming, due to the huge number of 6D cubes and point-set transformation operations.

Instead, we propose using a nested BnB search structure. An outer BnB searches the rotation space of $SO(3)$ and solves the bounds and corresponding optimal translations by calling an inner translation BnB. In this way, we only need to maintain two queues with significantly fewer cubes. Moreover, it avoids redundant point-set rotation operations for each rotation region, and takes the advantage that translation operations are computationally much cheaper.

The bounds for both the BnBs can be readily derived according to Section 4.2. In the outer rotation BnB, for a rotation cube C_r the bounds can be chosen as

$$\overline{E}_r = \min_{\forall \mathbf{t} \in \mathcal{C}_t} \sum_i e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t})^2, \tag{24}$$

$$\underline{E}_r = \min_{\forall \mathbf{t} \in \mathcal{C}_t} \sum_i \max\big(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}) - \gamma_{ri}, 0\big)^2, \tag{25}$$

where $\mathcal{C}_t$ is the initial translation cube. To solve the lower bound $\underline{E}_r$ in (25) with the inner translation BnB, the bounds for a translation cube C_t can be chosen as

$$\overline{E}_t = \sum_i \max\big(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) - \gamma_{ri}, 0\big)^2, \tag{26}$$

$$\underline{E}_t = \sum_i \max\big(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) - (\gamma_{ri} + \gamma_t), 0\big)^2. \tag{27}$$

By setting all the rotation uncertainty radii γ_{ri} in (26) and (27) to zero, the translation BnB solves $\overline{E}_r$ in (24). A detailed description is given in Algorithm 1 and Algorithm 2.

Search strategy and stop criterion. In both BnBs, we use a best-first search strategy. Specifically, each of the BnBs maintains a priority queue; the priority of a cube is opposite to its lower bound. Once the difference between so-far-the-best error E^* and the lower bound $\underline{E}$ of the current cube is less than a threshold ϵ, the BnB stops. Another possible strategy is to set $\epsilon = 0$ and terminate the BnBs when the remaining cubes are sufficiently small.

5.2   Integration with the ICP Algorithm

Lines 9-11 of Algorithm 1 show that whenever the outer BnB finds a cube C_r that has an upper bound lower than the current best function value, it will call conventional ICP, initialized with the center rotation of C_r and the corresponding best translation.

Fig. 6 illustrates the collaborative relationship between ICP and BnB. Under the guidance of the global BnB, ICP converges into local minima one by one, with each local minimum having lower error than the previous one, and ultimately reaches the global minimum. Since ICP monotonically decreases the current-best error E^* (cf. [11]), the search
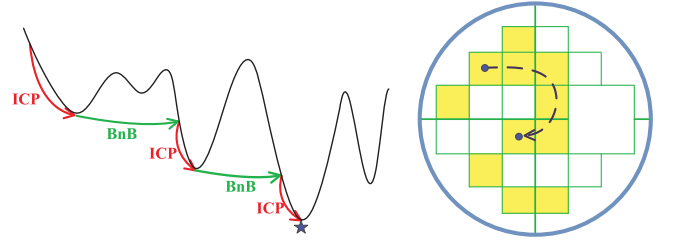

Fig. 6. Collaboration of BnB and ICP. *Left*: BnB and ICP collaboratively update the upper bounds during the search process. *Right*: with the guidance of BnB, ICP only explores un-discarded, promising cubes with small lower bounds marked up by BnB.

path of the local ICP is confined to un-discarded, promising sub-cubes with small lower bounds, as illustrated in Fig. 6.

In this way, the global BnB search and the local ICP search are intimately integrated in the proposed method. The former helps the latter jump out of local minima and guides the latter's next search; the latter accelerates the former's convergence by refining the upper bound, hence improving the efficiency.

5.3   Outlier Handling with Trimming

In statistics, trimming is a strategy to obtain a more robust statistic by excluding some of the extreme values. It is used in Trimmed ICP [60] for robust point-set registration. Specifically, in each iteration, only a subset $\mathcal{S}$ of the data points that have the smallest closest distances are used for motion computation. Therefore, the registration error will be

$$E^{Tr} = \sum_{i \in \mathcal{S}} e_i(\mathbf{R}, \mathbf{t})^2. \tag{28}$$

To robustify our method with trimming, it is necessary to derive new upper and lower bounds of (28). We have the following corollary.

Algorithm 1. Go-ICP—The Main Algorithm: BnB Search for Optimal Registration in $SE(3)$

Input: Data and model points; threshold ϵ; initial cubes $\mathcal{C}_r, \mathcal{C}_t$.
Output: Globally minimal error E^* and corresponding $\mathbf{r}^*, \mathbf{t}^*$.
1 Put $\mathcal{C}_r$ into priority queue Q_r.
2 Set $E^* = +\infty$.
3 **loop**
4   Read out a cube with lowest lower-bound $\underline{E}_r$ from Q_r.
5   Quit the loop if $E^* - \underline{E}_r < \epsilon$.
6   Divide the cube into 8 sub-cubes.
7   **foreach** sub-cube C_r **do**
8   Compute $\overline{E}_r$ for C_r and corresponding optimal $\mathbf{t}$ by calling Algorithm 2 with $\mathbf{r}_0$, zero uncertainty radii, and E^*.
9   **if** $\overline{E}_r < E^*$ **then**
10   Run ICP with the initialization $(\mathbf{r}_0, \mathbf{t})$.
11   Update E^*, $\mathbf{r}^*$, and $\mathbf{t}^*$ with the results of ICP.
12   **end**
13   Compute $\underline{E}_r$ for C_r by calling Algorithm 2 with $\mathbf{r}_0$, γ_r and E^*.
14   **if** $\underline{E}_r \geqslant E^*$ **then**
15   Discard C_r and continue the loop;
16   **end**
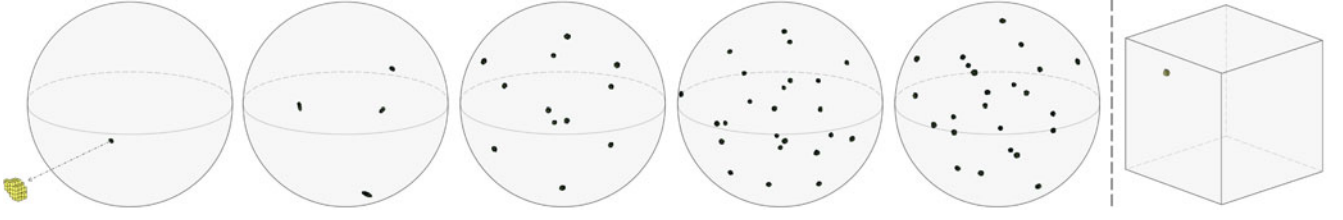17   Put C_r into Q_r.
18   **end**
19 **end**

Fig. 7. Remaining cubes of the BnBs. The first five figures show the remaining cubes in the rotation π-ball of the rotation BnBs, for an irregular tetrahedron, a cuboid with three different side-lengths, a regular tetrahedron, a regular cube, and a regular octahedron respectively. The last figure shows a typical example of remaining cubes of a translation BnB, for the irregular tetrahedron. (Best viewed when zoomed in.)

Algorithm 2. BnB Search for Optimal Translation given Rotation

---

> **Input:** Data and model points; threshold ϵ; initial cube $\mathcal{C}_t$; rotation $\mathbf{r}_0$; rotation uncertainty radii γ_r, so-far-the-best error E^*.
> **Output:** Minimal error E_t^* and corresponding $\mathbf{t}^*$.

1   Put $\mathcal{C}_t$ into priority queue Q_t.
2   Set $E_t^* = E^*$.
3   **loop**
4   Read out a cube with lowest lower-bound $\underline{E}_t$ from Q_t.
5   Quit the loop if $E_t^* - \underline{E}_t < \epsilon$.
6   Divide the cube into 8 sub-cubes.
7   **foreach** sub-cube C_t **do**
8   Compute $\overline{E}_t$ for C_t by (26) with $\mathbf{r}_0, \mathbf{t}_0$ and γ_r.
9   **if** $\overline{E}_t < E_t^*$ **then**
10   Update $E_t^* = \overline{E}_t, \mathbf{t}^* = \mathbf{t}_0$.
11   **end**
12   Compute $\underline{E}_t$ for C_t by (27) with $\mathbf{r}_0, \mathbf{t}_0, \gamma_r$ and γ_t.
13   **if** $\underline{E}_t \geqslant E_t^*$ **then**
14   Discard C_t and continue the loop.
15   **end**
16   Put C_t into Q_t.
17   **end**
18 **end**

Corollary 4 (Bounds of the trimmed L_2 error). *The upper bound $\overline{E^{Tr}}$ and lower bound $\underline{E^{Tr}}$ of the registration error with trimming for the domain $C_r \times C_t$ can be chosen as*
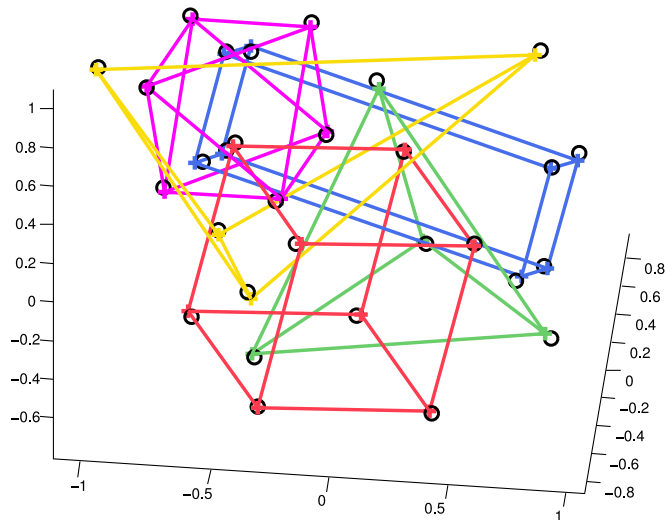
$$\overline{E^{Tr}} \doteq \sum_{i \in \mathcal{P}} \overline{e_i}^2, \tag{29}$$

$$\underline{E^{Tr}} \doteq \sum_{i \in \mathcal{Q}} \underline{e_i}^2, \tag{30}$$

where $\overline{e_i}$, $\underline{e_i}$ are bounds of the per-point residuals defined in (14), (15) respectively, and $\mathcal{P}$, $\mathcal{Q}$ are the trimmed point-sets having smallest values of $\overline{e_i}$, $\underline{e_i}$ respectively, with $|\mathcal{P}| = |\mathcal{Q}| = |\mathcal{S}| = K$.

Proof. The upper bound in (29) is chosen trivially. To see the validity of the lower bound in (30), observe that $\forall (\mathbf{r}, \mathbf{t}) \in C_r \times C_t$,

$$\underline{E^{Tr}} = \sum_{i \in \mathcal{Q}} \underline{e_i}^2 \leq \sum_{i \in \mathcal{S}} \underline{e_i}^2 \leq \sum_{i \in \mathcal{S}} e_i(\mathbf{R_r}, \mathbf{t})^2 = E^{Tr}. \tag{31}$$

$\square$

Based on this corollary, the corresponding bounds in the nested BnB can be readily derived. As proved in [60], iterations of Trimmed ICP decrease the registration error monotonically to a local minimum. Thus it can be directly integrated into the BnB procedure.

Fast trimming. A straightforward yet inefficient way to do trimming is to sort the residuals outright and use the K smallest ones. In this paper, we employ the Introspective Selection algorithm [64] which has $O(N)$ performance in both the worst case and average case.

Other robust extensions. In the same spirit as trimming, other ICP variants such as [59], [65] can be handled. The

Fig. 8. A clustered scene (black circles) and the registration results of Go-ICP for the five shapes.

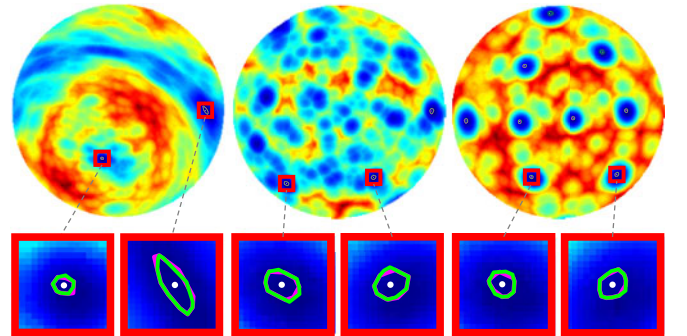
Fig. 9. Remaining rotation domains of the outer rotation BnB on 2D slices of the π-ball, for the synthetic points. Results using the DT and the kd-tree are within magenta and green polygons, respectively. The white dots denote optimal rotations. From left to right: a cuboid, a regular tetrahedron and a regular cube. The colors on the slices indicate registration errors evaluated via inner translation BnB: red for high error and blue for low error. (Best viewed when zoomed in)
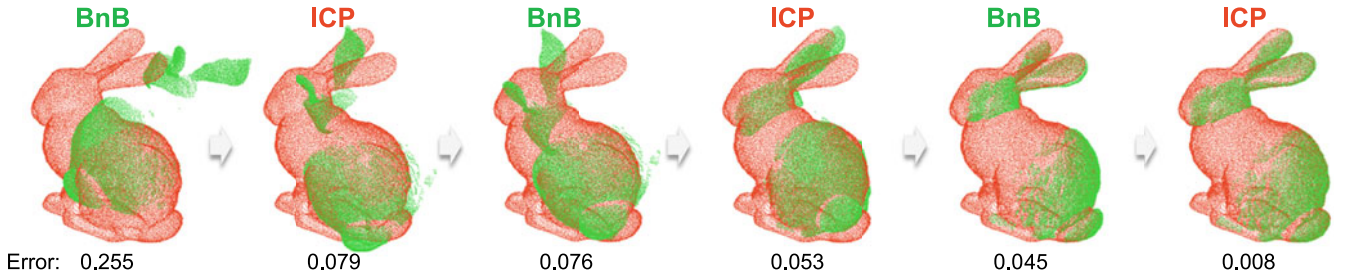
Fig. 10. Evolution of Go-ICP registration for the bunny dataset. The model point-set and data point-set are shown in red and green respectively. BnB and ICP collaboratively update the registration: ICP refines the solution found by BnB and BnB guides ICP into the convergence basins of multiple local minima with increasingly lower registration errors.

method can also be adapted to LM-ICP [16], where the new lower-bound is simply a robust kernelized version of the current one. It may also be extended to ICP variants with L_p-norms [66], such as the robustness-promoting L_1-norm.

6 EXPERIMENTS

We implemented the method[2] in C++ and tested it on a standard PC with an Intel i7 3.4 GHz CPU. In the experiments reported below, the point-sets were pre-normalized such that all the points were within the domain of $[-1, 1]^3$. Although the goal was to minimize the L_2 error in (1), the root-mean-square (RMS) error is reported for better comprehension.

Closest-point distance computation. To speed up the closest distance computation, a kd-tree data structure can be used. We also provide an alternative solution that is used more often in the experiments – a 3D Euclidean distance transform (DT) [16] used to compute closest distances for fast bound evaluation.[3] A DT approximates the closest-point distances in the real-valued space by distances of uniform grids, and pre-computes them for constant-time retrieval (details about our DT implementation are included in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2015.2513405). Despite the DT can introduce approximation errors thus the convergence gap may not be exactly ϵ, in the following experiments our method works very well with a $300 \times 300 \times 300$ DT for optimal registration. Naturally, higher resolutions can be used when necessary.

6.1 Optimality

To verify the correctness of the derived bounds and the optimality of Go-ICP, we first use a convergence condition similar to [63] for the BnBs. Specifically, we set the the threshold of a BnB to be 0, and specify a smallest cube size at which the BnB stops dividing a cube. In this way, we can examine the uncertainty in the parameter space after the BnB stops. Both the DT and kd-tree are tested in these experiments.

6.1.1 Synthetic Points

We first tested the method on a synthetically generated scene with simple objects. Specifically, five 3D shapes were

created: an irregular tetrahedron, a cuboid with three different side-lengths, a regular tetrahedron, a regular cube, and a regular octahedron. Note that the latter four shapes have self-symmetries. All the shapes were then placed together, each with a random transformation, to generate clustered scenes. Zero-mean Gaussian noise with standard deviation $\sigma = 0.01$ was added to the scene points. We created such a scene as shown in Fig. 8, and applied Go-ICP to register the vertices of each shape to the scene points.

To test the rotation BnB, we set the parameter domain to be $[-\pi, \pi]^3 \times [-1, 1]^3$ and the minimal volume of a rotation cube to $1.5E{-}5$ (~ 1 degree uncertainty). The lower bound of a rotation cube was set to be the global lower bound of the invoked translation BnB. Thus the threshold of translation BnB is not very important and we set it to a small value ($0.0001 \times N$ where N is the data point number). The initial errors E_t^* of translation BnBs were set to infinity.

In all tests, Go-ICP produced correct results with both the DT and kd-tree. The remaining rotation cubes using the DT and kd-tree respectively are almost visually indistinguishable, and Fig. 7 shows the results using the DT. It is interesting to see that the remaining cubes formed one cluster for the irregular tetrahedron, four clusters for the cuboid, 12 clusters for the regular tetrahedron, and 24 clusters for the regular cube and octahedron. These results conform to the
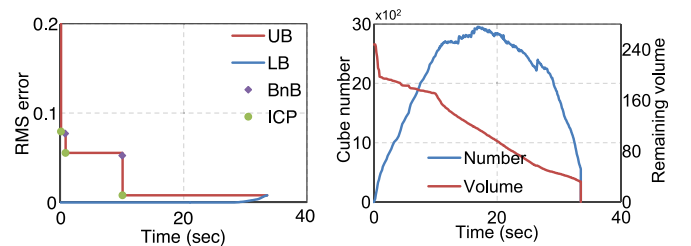

Fig. 11. Evolution of the bounds (left) and cubes (right) in the rotation BnB with a DT on the bunny point-sets. See text for details.
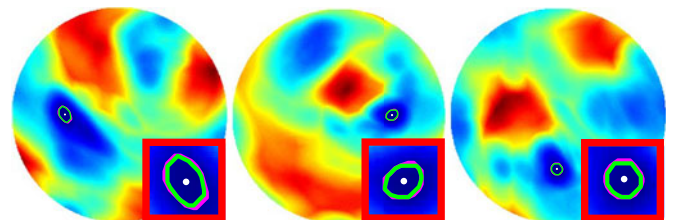

Fig. 12. Remaining rotation domains of the outer rotation BnB on 2D slices of the π-ball, for the bunny point-sets. The three slices pass through the optimal rotation and the X-, Y-, Z-axes respectively. See also the caption of Fig. 9. (Best viewed when zoomed in.)

---

2. Source code and demo can be found on the author's webpage.

3. Local ICP is called infrequently so we simply use a kd-tree for it. The refined upper-bounds from the found ICP solutions are evaluated via the DT for consistency.
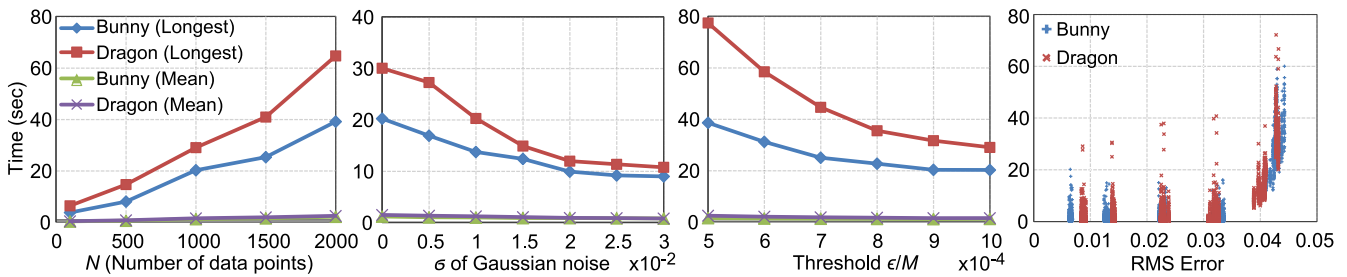
Fig. 13. Running time of the Go-ICP method with DTs on the bunny and dragon point-sets with respect to different factors. The evaluation was conducted on 10 data point-sets with 100 random poses (i.e., $1,000$ pairwise registrations).

geometric properties of these shapes, and validated the derived bounds. Investigating shape self-similarity would be a practical application of the algorithm.

Moreover, Fig. 9 shows some typical remaining rotation domains on 2D slices of the rotation π-ball.[4] The non-convexity of the problem can be clearly seen from the presence of many local minima. It can also been seen that the remaining rotation domains using a DT and kd-tree are highly consistent, and the optima are well contained by them.

The translation BnB can be easily verified by running it with rotations picked from the remaining rotation cubes. The threshold was set to be 0, and the minimal side-length of a translation cube was set to be 0.01. The last figure of Fig. 7 shows a typical result.

More results of remaining cubes can be found in the supplementary material, available online.

6.1.2 Real Data

Similar experiments were conducted on real data. We applied our method to register a bunny scan bun090 from the Stanford 3D dataset[5] to the reconstructed model. Since model and data point-sets are of similar spatial extents, we set the parameter domain to be $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$ which is large enough to contain the optimal solution. We randomly sampled 500 data points, and did similar tests to those on the synthetic points. The translation BnB threshold was set to $0.001 \times N$, and the remaining rotation cubes from the outer rotation BnB were similar to the first figure in Fig. 7 (i.e., one cube cluster). Fig. 12 shows the results on three slices of the rotation π-ball.

Additionally, we recorded the bound and cube evolutions in the rotation BnB which are presented in Fig. 11. It can be seen that BnB and ICP collaboratively update the global upper bound. Corresponding transformations for each global upper bound found by BnB and ICP are shown in Fig. 10. Note that in the fourth image the pose has been very close to the optimal one, which indicates that ICP may fail even if reasonably good initialization is given.

Although the convergence condition used in this section worked successfully, we found that using a small threshold ϵ of the bounds to terminate a BnB also works well in practice. It is more efficient and produces satisfactory results. In the following experiments, we used this strategy for the BnBs.

6.2 "Partial" to "Full" Registration

In this section, we test the performance of Go-ICP by registering partially scanned point clouds to full 3D model point clouds. The bunny and dragon models from the Stanford 3D dataset were used for evaluation. All 10 partial scans of the bunny dataset were used as data point-sets. For the dragon model, we selected 10 scans generated from different view points as data point-sets. The reconstructed bunny and dragon models were used as model point-sets.

For each of these 20 scans, we first performed 100 tests with random initial rotations and translations. The transformation domain to explore for Go-ICP was set to be $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$. We sampled $N = 1,000$ data points from each scan, and set the convergence threshold ϵ to be $0.001 \times N$. As expected, *Go-ICP achieved 100 percent correct registration on all the 2,000 registration tasks on the bunny and dragon models*, with both the DT and kd-tree. All rotation errors were less than 2 degrees and all translation errors were less than 0.01. With a DT, the mean/longest running times of Go-ICP, in the 1,000 tests on 1,000 data points and 20,000-40,000 model points, were 1.6 s/22.3 s for bunny and 1.5 s/28.9 s for dragon. Fig. 14 shows the running time histograms The running times with a kd-tree were typically 40-50 times longer than that with the DT. The solutions from using the DT and the kd-tree respectively were highly
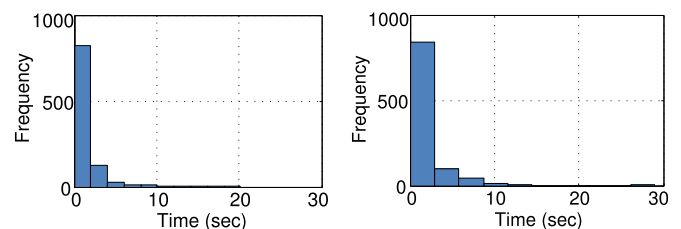

Fig. 14. Running time histograms of Go-ICP with DTs for the bunny (left) and dragon (right) point-sets.
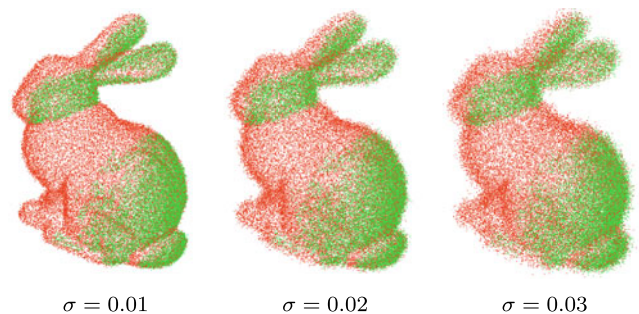

$\sigma = 0.01$ $\sigma = 0.02$ $\sigma = 0.03$

Fig. 15. Registration with different levels of Gaussian noise.

4. We chose the slices passing two randomly-selected optimal rotations plus the origin. Due to shape symmetry there may exist more than two optimal rotations on one slice.

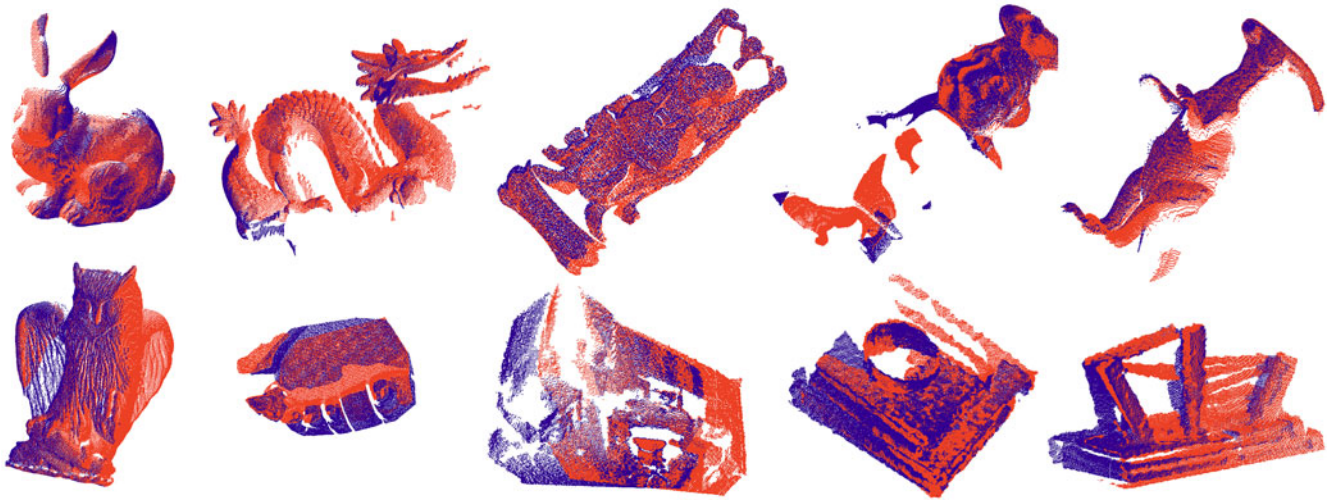5. http://graphics.stanford.edu/data/3Dscanrep/

Fig. 16. Registration with partial overlap. Go-ICP with the trimming strategy successfully registered the 10 point-set pairs with 100 random relative poses for each of them. The point-sets in red and blue are denoted as point-set A and point-set B, respectively. The trimming settings and running times are presented in Table 1.

consistent (the largest rotation difference was below 1 degree). See the supplementary material for detailed result and running time comparisons for the DT and the kd-tree.

We then analyzed the running time of the proposed method under various settings using the DT. We analyzed the influence of each factor by varying it while keeping others fixed. Default factor settings: number of data points $N = 1,000$, no added Gaussian noise (i.e. standard deviation $\sigma = 0$) and convergence threshold $\epsilon = 0.001 \times N$.

Effect of number of points. In this experiment, the running time was tested for different numbers of points. Since the DT was used for closest-point distance retrieval, the number of model points does not significantly affect the speed of our method. To test the running time with respect to different numbers of data points, we randomly sampled the data point-set. As presented in Fig. 13, the running time manifested a linear trend since closest-point distance retrieval was $O(1)$ and the convergence threshold varied linearly with the number of data points.

Effect of noise. We examined how the noise level impacted the running time by adding Gaussian noise to both the data and model point-sets. The registration results on the corrupted bunny point-sets are shown in Fig. 15. We found that, as shown in Fig. 13, the running time decreased as the noise level increased (until $\sigma = 0.02$). This is because the Gaussian noise (especially that added to the model points)

smoothed out the function landscape and widened the convergence basin of the global minimum, which made it easier for Go-ICP to find a good solution.

Effect of convergence threshold. We further investigated the running time with respect to the convergence threshold of the BnB loops. We set the threshold ϵ to depend linearly on N, since the registration error is a sum over the N data points. Fig. 13 shows that the smaller the threshold was, the slower our method performed. In our experiments, $\epsilon = 0.001 \times N$ was adequate to get a 100 percent success rate for the bunny and dragon point-sets. For cases when the local minima are small or close to the global minimum, the threshold can be set smaller.

Effect of optimal error. We also tested the running time *w.r. t.* the optimal registration error. To increase the error, Gaussian noise was added to the data point-set *only*. As shown in Fig. 13, the running time remained almost constant when the RMS error was less than 0.03. This is because the gap between the global lower bound and the optimal error was less than ϵ. Therefore, the running time depended primarily on when the global minimum was found, that is, the termination depended on the *decrease of*
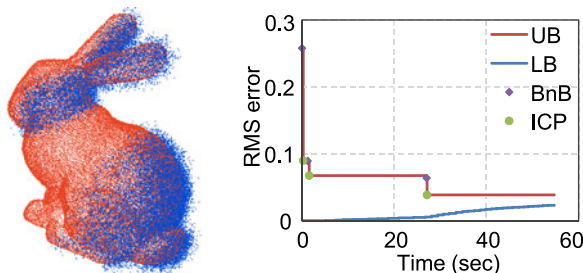

Fig. 17. Registration with high optimal error. *Left:* Gaussian noise was added to the data point-set to increase the RMS error. *Right:* the global minimum was found at about 25 s with a DT; the remainder of the time was devoted solely to increasing the lower bound.

TABLE 1
Running Time (in Seconds) of Go-ICP with DTs for the Registration of the Partially Overlapping Point-Sets in Fig. 16

	$A \rightarrow B$		$B \rightarrow A$	
	ρ	mean/max time	ρ	mean/max time
Bunny	10 percent	0.81/10.7	10 percent	0.49/7.25
Dragon	20 percent	2.99/43.5	40 percent	8.72/72.4
Buddha	10 percent	0.71/11.3	10 percent	0.60/14.8
Chef	20 percent	0.45/4.47	30 percent	0.52/3.79
Dinosaur	10 percent	2.03/23.5	10 percent	1.65/26.1
Owl	40 percent	12.5/87.5	40 percent	13.4/75.0
Denture	30 percent	6.74/74.7	30 percent	4.24/68.1
Room	30 percent	9.82/73.3	30 percent	18.4/107.3
Bowl	20 percent	3.19/20.3	30 percent	3.52/25.3
Loom	30 percent	8.64/67.2	20 percent	5.96/44.6

100 random relative poses were tested for each point-set pair and 1,000 data points were used. ρ is the trimming percentage.
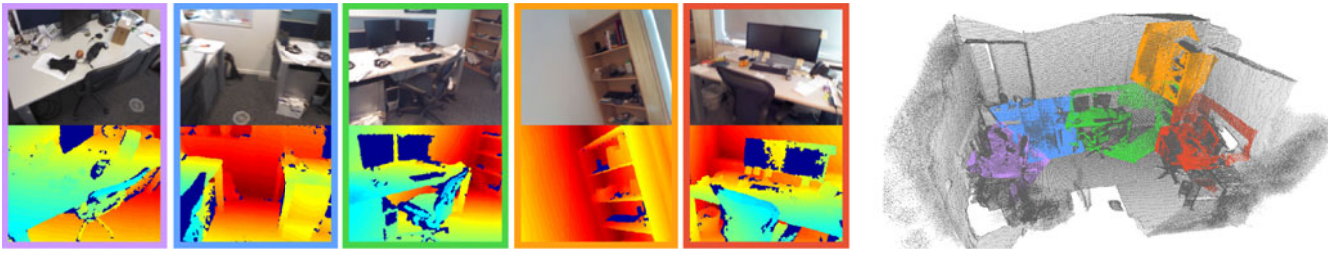
Fig. 18. Camera localization experiment. *Left*: 5 (out of 100) color and depth image pairs of the scene. (The color images were not used.) *Right*: Corresponding registration results. Note that the scene contains many similar structures, and the depth images only cover small portions of the scene, which make the 3D registration tasks very challenging.

the upper bound. However, it took longer to converge when the final RMS error was higher. Fig. 17 shows the bound evolution for bunny when the RMS error was increased to ~ 0.04. As can be seen, the global minimum was found at about 25 s, with the remainder of the time devoted to *increasing the lower bound*.

6.3 Registration with Partial Overlap

In this section, we tested the proposed method on partially overlapping point-sets. The data points in regions that are not overlapped by the other model point-set should be treated as outliers, as their correspondences are missing. Trimming was employed to deal with outliers as described in Section 5.3.

We used 10 point-set pairs shown in Fig. 16 to test Go-ICP with trimming. These point-sets were generated by different scanners and with different noise levels. The bunny, dragon and buddha models are from the Stanford 3D dataset. The chef and dinosaur models are from [67]. The denture was generated with a structured light 3D scanner.[6] The owl status is from [66] and the room scans are from [68]. The bowl and loom point-sets were collected by us with a Kinect. The overlapping ratio of the point-set pairs are between 50-95 percent.

For each of the 10 point-set pairs, we generated 100 random relative poses, and registered the two point-sets to each other. This lead to 2,000 registration tasks. The translation domain to explore for Go-ICP was set to be $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$. We chose the trimming percentages ρ as in Table 1, sampled $N = 1{,}000$ data points for each registration, and set all the convergence thresholds to $\epsilon = 0.001 \times K$ where $K = (1 - \rho) \times N$. Our method correctly registered the point-sets in all these tasks. All the rotation errors were less than 5 degrees and translation errors were less than 0.05 compared to the manually-set ground truths. The running times using DTs are presented in Table 1. In general, it takes the method a longer time compared to the outlier-free case due to 1) the emergence of additional local minima induced by the outliers and 2) the time-consuming trimming operations.

Choosing trimming percentages. In these experiments, each parameter ρ was chosen by visually observing the two point-sets and roughly guessing their non-overlapping ratios. The results were not very sensitive to ρ (e.g. setting ρ as $5\%, 10\%$ and 20% all led to a successfully registration for bunny). If no rough guess is available, one can gradually

increase ρ until a measure such as the inlier number or RMS error attains a set value, or apply the automatic overlap estimation proposed in [62]. We also plan to test other outlier handling strategies (cf. Section 5.3) in future.

6.4 More Applications

In this section, we present several additional scenarios where Go-ICP can be applied to achieve global optimality. Future efforts can be taken to extend the method and build complete real-world systems. In the following experiments, the transformation domain for exploration was set to be $[-\pi, \pi]^3 \times [-1, 1]^3$.

3D object localization. The proposed method is useful for model-based 3D object detection, localization and pose estimation from relatively large scenes. To experimentally verify this, we tested our method on one sequence of the camera localization dataset [68]. Fig. 18 shows a sample color and depth image pair, and a 3D model of the office scene. Our goal was to estimate the camera poses by registering the point clouds of the depth images onto the 3D scene model. We evenly sampled the sequence taken by a smoothly moving camera to 100 depth images. Each depth image was then sampled to 400-600 points. We set our method to seek a solution with the registration error smaller than $0.0001 \times N$, and the method registered the 100 point-sets with the mean/longest running time of 32 s/178 s using a DT. The rotation errors and translation errors were all below 5 degrees and 10 cm. Fig. 18 shows five typical registration results.

We then used the RGB-D Object Dataset [69], with the goal of registering the points of a baseball cap to a point cloud of the scene, as shown in Fig. 19. We sampled $N = 100$ points from the cap model, and set the trimming percentage and threshold to be $\rho = 10\%$ and $\epsilon = 0.00003 \times K$ respectively. Go-ICP successfully localized the cap in 42 seconds with a DT.

Camera extrinsic calibration. In the work of Yang et al. [9], the sparse point-set from a color camera, obtained by single view 3D reconstruction, was registered onto the dense
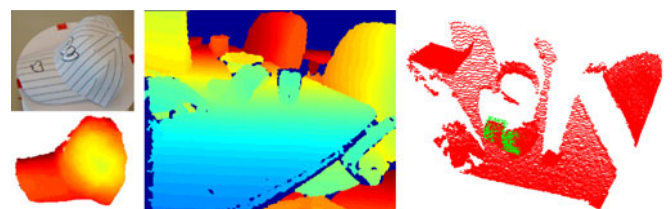
Fig. 19. 3D object localization experiment. *Left*: a labelled object and its depth image to generate the data point-set. *Middle*: a scene depth image to generate the model point-set. *Right*: the registration result.
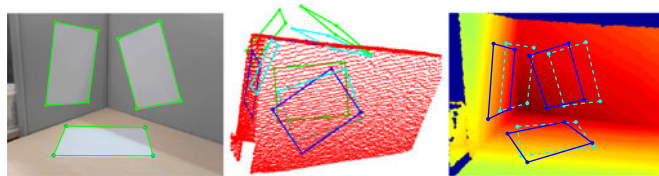
Fig. 20. RGB-D extrinsic calibration experiment. *Left*: the color image with extracted line segments for single view 3D reconstruction. *Middle*: the initial 3D registration (in green), the result of ICP (in cyan) and the result of Go-ICP (in blue) (the lines are for visualization purposes only). *Right*: the depth image with a projection of the registered 3D points from ICP (in cyan) and Go-ICP (in blue).

point-set from a depth camera to obtain the camera relative pose. Fig. 20 shows an example where 12 points are reconstructed. We found that ICP often failed to find the correct registration when the pose difference between the cameras was reasonably large. To the best of our knowledge, few methods can perform such *sparse-to-dense registration* reliably without human intervention, due to the difficulty of building putative correspondences. Setting ϵ to be $0.00001 \times N$, Go-ICP with a DT found the optimal solution in less than 1s. Note that the surfaces are not exactly perpendicular to each other.

7 CONCLUSION

We have introduced a globally optimal solution to Euclidean registration in 3D, under the L_2-norm closest-point error metric originally defined in ICP. The method is based on the branch-and-bound algorithm, thus global optimality is guaranteed regardless of the initialization. The key innovation is the derivation of registration error bounds based on the $SE(3)$ geometry.

The proposed Go-ICP algorithm is especially useful when an exactly optimal solution is highly desired or when a good initialization is not reliably available. For practical scenarios where real-time performance is not critical, the algorithm can be readily applied or used as an optimality benchmark.

REFERENCES

[1] G. Blais and M. D. Levine, "Registering multiview range data to create 3D computer objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 820–824, Aug. 1995.
[2] D. F. Huber and M. Hebert, "Fully automatic registration of multiple 3D data sets," *Image Vision Comput.*, vol. 21, no. 7, pp. 637–650, 2003.
[3] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.
[4] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.
[5] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM–3D mapping outdoor environments," *J. Field Robot.*, vol. 24, nos. 8/9, pp. 699–722, 2007.

[6] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auton. Robots*, vol. 34, no. 3, pp. 133–148, 2013.
[7] T. Makela, P. Clarysse, O. Sipila, N. Pauna, Q. C. Pham, T. Katila, and I. E. Magnin, "A review of cardiac image registration methods," *IEEE Trans. Med. Imaging*, vol. 21, no. 9, pp. 1011–1021, Sep. 2002.
[8] W. Zhao, D. Nister, and S. Hsu, "Alignment of continuous video onto 3d point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1305–1318, Aug. 2005.
[9] J. Yang, Y. Dai, H. Li, H. Gardner, and Y. Jia, "Single-shot extrinsic calibration of a generically configured RGB-D camera rig from scene constraints," in *Proc. Int. Symp. Mixed Augmented Reality*, 2013, pp. 181–188.
[10] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 3936–3943.
[11] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
[12] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1991, vol. 3, pp. 2724–2729.
[13] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision*, vol. 13, no. 2, pp. 119–152, 1994.
[14] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.
[15] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2006, vol. 1, pp. 519–528.
[16] A. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image Vision Comput.*, vol. 21, no. 13, pp. 1145–1153, 2003.
[17] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. Eur. Conf. Comput. Vision*, 2004, pp. 558–569.
[18] R. Sandhu, S. Dambreville, and A. Tannenbaum, "Point set registration via particle filtering and stochastic dynamics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1459–1473, Aug. 2010.
[19] L. Silva, O. R. P. Bellon, and K. L. Boyer, "Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 762–776, May 2005.
[20] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3212–3217.
[21] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2006, vol. 1, pp. 1297–1304.
[22] R. I. Hartley and F. Kahl, "Global optimization through searching rotation space and optimal estimation of the essential matrix," in *Proc. Int. Conf. Comput. Vision*, 2007, pp. 1–8.
[23] H. Li and R. Hartley, "The 3D-3D registration problem revisited," in *Proc. Int. Conf. Comput. Vision*, 2007, pp. 1–8.
[24] J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *Proc. Int. Conf. Comput. Vision*, 2013, pp. 1457–1464.
[25] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Int. Conf. 3-D Digital Imaging Model.*, 2001, pp. 145–152.
[26] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. Visual. Comput. Graphics*, vol. 19, no. 7, pp. 1199–1217, Jul. 2013.
[27] U. Castellani and A. Bartoli, "3D shape registration," in *3D Imaging, Analysis, and Applications*. New York, NY, USA: Springer, 2012, pp. 221–264.
[28] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*. New York, NY, USA: Springer, 1978, pp. 105–116.
[29] B. Jian and B. Vemuri, "A robust algorithm for point set registration using mixture of Gaussians," in *Proc. Int. Conf. Comput. Vision*, 2005, vol. 2, pp. 1246–1251.

[30] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[31] D. Campbell and L. Petersson, "An adaptive data representation for robust point-set registration and merging," in *Proc. Int. Conf. Comput. Vision*, 2015, pp. 4292–4300.

[32] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. Goldman-Rakic, and J. Duncan, "A robust point-matching algorithm for autoradiograph alignment," *Med. Image Anal.*, vol. 1, no. 4, pp. 379–398, 1997.

[33] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *Proc. Eur. Conf. Comput. Vision*, 2002, pp. 418–432.

[34] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 3, pp. 2743–2748.

[35] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3D registration reliability and speed-a comparison of ICP and NDT," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3907–3912.

[36] G. C. Sharp, S. W. Lee, and D. K. Wehe, "ICP registration using invariant features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 90–102, Jan. 2002.

[37] A. E. Johnson and B. K. Sing, "Registration and integration of textured 3D data," *Image Vision Comput.*, vol. 17, no. 2, pp. 135–147, 1999.

[38] C. Robertson and R. B. Fisher, "Parallel evolutionary registration of range data," *Comput. Vision Image Understanding*, vol. 87, no. 1, pp. 39–50, 2002.

[39] M. P. Wachowiak, R. Smolíková, Y. Zheng, J. M. Zurada, and A. S. Elmaghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 289–301, Jun. 2004.

[40] C. Papazov and D. Burschka, "Stochastic global optimization for robust point set registration," *Comput. Vision Image Understanding*, vol. 115, no. 12, pp. 1598–1609, 2011.

[41] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," *Proc. Eurographics Symp. Geometry Process.*, 2005, vol. 2, no. 3, pp. 5–14.

[42] O. J. Woodford, M.-T. Pham, A. Maki, F. Perbet, and B. Stenger, "Demisting the Hough transform for 3D shape recognition and registration," *Int. J. Comput. Vision*, vol. 106, no. 3, pp. 332–341, 2014.

[43] J.-C. Bazin, Y. Seo, and M. Pollefeys, "Globally optimal consensus set maximization through rotation search," in *Proc. Asian Conf. Comput. Vision*, 2012, pp. 539–551.

[44] 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*, vol. 24, pp. 381–395, 1981.

[45] S. Irani and P. Raghavan, "Combinatorial and experimental results for randomized point matching algorithms," *Comput. Geometry*, vol. 12, no. 1, pp. 17–31, 1999.

[46] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *ACM Trans. Graphics*, vol. 27, no. 3, p. 85, 2008.

[47] T. M. Breuel, "Implementation techniques for geometric branch-and-bound matching methods," *Comput. Vision Image Understanding*, vol. 90, no. 3, pp. 258–294, 2003.

[48] D. M. Mount, N. S. Netanyahu, and J. Le Moigne, "Efficient algorithms for robust feature matching," *Pattern Recog.*, vol. 32, no. 1, pp. 17–38, 1999.

[49] F. Pfeuffer, M. Stiglmayr, and K. Klamroth, "Discrete and geometric branch and bound algorithms for medical image registration," *Ann. Oper. Res.*, vol. 196, no. 1, pp. 737–765, 2012.

[50] C. Olsson, F. Kahl, and M. Oskarsson, "Branch-and-bound methods for euclidean registration problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 783–794, May 2009.

[51] A. P. Bustos, T.-J. Chin, and D. Suter, "Fast rotation search with stereographic projections for 3D registration," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2014, pp. 3930–3937.

[52] O. Enqvist, K. Josephson, and F. Kahl, "Optimal correspondences from pairwise constraints," in *Proc. Int. Conf. Comput. Vision*, 2009, pp. 1295–1302.

[53] O. Enqvist and F. Kahl, "Robust optimal pose estimation," in *Proc. Eur. Conf. Comput. Vision*, 2008, pp. 141–153.

[54] T. Ruland, T. Pajdla, and L. Kruger, "Globally optimal hand-eye calibration," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2012, pp. 1035–1042.

[55] J. Yang, H. Li, and Y. Jia, "Optimal essential matrix estimation via inlier-set maximization," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 111–126.

[56] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A*, vol. 4, no. 4, pp. 629–642, 1987.

[57] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.

[58] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[59] G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie, "From accurate range imaging sensor calibration to accurate model-based 3D object localization," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 1992, pp. 83–89.

[60] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm," *Image Vision Comput.*, vol. 23, no. 3, pp. 299–309, 2005.

[61] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Oper. Res.*, vol. 14, no. 4, pp. 699–719, 1966.

[62] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[63] R. I. Hartley and F. Kahl, "Global optimization through rotation space search," *Int. J. Comput. Vision*, vol. 82, no. 1, pp. 64–79, 2009.

[64] D. R. Musser, "Introspective sorting and selection algorithms," *Softw.: Practice Experience*, vol. 27, no. 8, pp. 983–993, 1997.

[65] T. Masuda and N. Yokoya, "A robust method for registration and segmentation of multiple range images," *Proc. CAD-Based Vision Workshop*, 1994, pp. 106–113.

[66] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Proc. Eurographics Symp. Geometry Process.*, 2013, pp. 113–123.

[67] A. S. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1584–1601, Oct. 2006.

[68] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2013, pp. 2930–2937.

[69] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1817–1824.
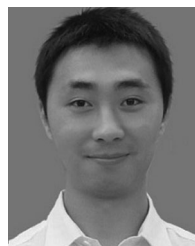
Jiaolong Yang received the BS degree in computer science from the Beijing Institute of Technology (BIT), Beijing, China, in 2010. He is currently working toward the dual PhD degree at the Beijing Laboratory of Intelligent Information Technology, BIT, and the Research School of Engineering, Australian National University (ANU). He received a Chinese Government Scholarship by the China Scholarship Council to study at ANU from September 2013 to September 2015. His current research interests include 3D and geometric computer vision. He is a student member of the IEEE.
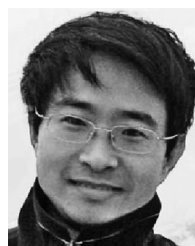
Hongdong Li is a faculty member with the Research School of Engineering, The Australian National University (ANU) and National ICT Australia. He joined the ANU in 2004 after graduated from Zhejiang University in China. His current research interests include vision geometry, 3D computer vision, and optimization. He serves various programme committees for major computer vision conferences, including being an area chair for ICCV'13, ECCV'14, CVPR'15, and ICCV'15. He is a regular reviewer for *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *International Journal of Computer Vision*. He was a winner for the CVPR'12 Best Paper Award, ICPR'10 Best Student Paper Award, and ICIP'14 Best Student Paper Award. He is a member of the IEEE.

Dylan Campbell received the BE degree in mechatronic engineering from the University of New South Wales in Sydney, Australia, in 2012. He is currently working toward the PhD degree in computer vision at the Australian National University and National ICT Australia. His research interests include place recognition, global localisation, mobile robotics, 3D reconstruction, multisensor perception, and global optimization. He is a student member of the IEEE.

Yunde Jia received the MS and PhD degrees in mechatronics from the Beijing Institute of Technology (BIT), Beijing, China, in 1986 and 2000, respectively. He is currently a professor of computer science with BIT and serves as the director of the Beijing Laboratory of Intelligent Information Technology, School of Computer Science. He has previously served as the executive dean of the School of Computer Science, BIT, from 2005 to 2008. He was a visiting scientist at Carnegie Mellon University, Pittsburgh, Pennsylvania, from 1995 to 1997, and a Visiting Fellow at the Australian National University, Acton, Australia, in 2011. His current research interests include computer vision, media computing, and intelligent systems. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.