Go-ICP: Solving 3D Registration Efficiently and Globally Optimally

Jiaolong Yang [1,2], Hongdong Li [2], Yunde Jia [1]

[1]Beijing Lab of Intelligent Information Technology, Beijing Institute of Technology
[2]Australian National University and NICTA Australia
{yangjiaolong,jiayunde}@bit.edu.cn, hongdong.li@anu.edu.au

Abstract

Registration is a fundamental task in computer vision. The Iterative Closest Point (ICP) algorithm is one of the widely-used methods for solving the registration problem. Based on local iteration, ICP is however well-known to suffer from local minima. Its performance critically relies on the quality of initialization, and only local optimality is guaranteed. This paper provides the very first globally optimal solution to Euclidean registration of two 3D pointsets or two 3D surfaces under the L_2 error. Our method is built upon ICP, but combines it with a branch-and-bound (BnB) scheme which searches the 3D motion space $SE(3)$ efficiently. By exploiting the special structure of the underlying geometry, we derive novel upper and lower bounds for the ICP error function. The integration of local ICP and global BnB enables the new method to run efficiently in practice, and its optimality is exactly guaranteed. We also discuss extensions, addressing the issue of outlier robustness.

1. Introduction

The Iterative Closest Point (ICP) [5, 10, 35] is a well-known algorithm for registering two point sets (in 2D or 3D) under Euclidean transformation. It has been successfully applied to solving numerous real-world applications. The concept of ICP is simple and intuitive. It alternates between estimating geometric transformation (rotation and translation), and estimating the point-wise correspondences. Partly due to its conceptual simplicity, as well as its good performance in practice, ICP is one of the most popular algorithms for registration, widely used in computer vision, and beyond computer vision.

ICP is however also well-known for its suffering from the issue of local minima, due to the local iterative procedure it adopts. Being an iterative method, ICP requires a good initialization to start, without which the algorithm may easily get trapped into local minima. If this situation happens, the solution found by ICP may be far away from the true (optimal) solution, leading to erroneous estimation. More seriously, ICP itself has no way to tell whether or not it has been trapped into a local minimum. Despite that this drawback of local-minima is generally well-known, relatively few papers have tackled this issue explicitly.

This paper is, to the best of our knowledge, the very first that proposes a truly globally optimal solution to ICP type Euclidean registration in 3D. It provides *guaranteed optimality* without the need for a good initialization. In fact, our new method always produces the exact and globally optimal solution (up to *any* desired accuracy), starting from *any* initialization.

We call our new algorithm *the Globally Optimal ICP* (or Go-ICP in short), because it largely resembles the computational structure of a standard ICP. It still relies on the search of closest-points at each iteration. Moreover, a standard (local) ICP is employed as a subroutine in our new algorithm. By exploiting special structure of the underlying geometry of $SE(3)$, and with the help of local ICP, our Go-ICP algorithm works rather efficiently. We have conducted extensive tests on both synthetic data and real data; satisfactory results (both in terms of theoretical optimality and computational efficiency) are obtained for all tests.

Although Go-ICP is specifically designed for 3D Euclidean registration since we take advantage of the geometry of $SE(3)$, the same techniques used in this paper may be inspiring for other cases as well (*e.g.* 2D or affine). Moreover, confining to 3D should not be considered as a limitation, as the 3D case is arguably the most useful case for registration. Our error metric used by Go-ICP follows strictly that of ICP, namely, minimizing the L_2 norm of the vector of residuals. However, with small effort we can extend it to other metrics such as the L_1 norm, Least Median Squares and other variants of ICP as well.

2. Related Work

There has been a large volume of work published on ICP, preventing us from giving a comprehensive list. Below we only list a few most relevant works, that either aimed to achieve optimal ICP or, more generally, addressed optimality in Euclidean registration. For other papers, the reader is referred to a recent survey [8] or [31] and the references therein.

To alleviate the local minima issue, previous work has attempted to enlarge the basin of convergence by smoothing out the objective function. Good performance has been observed from Fitzgibbon's LM-ICP [13] with robust kernels. Probability density based techniques [18, 33, 27, 6] have been used to model the points with Gaussian Mixture Models. Although improved robustness can be archived, the optimization procedures they adopted are still local search. Efforts have been devoted to various heuristic stochastic optimizations, *e.g.* particle swarm optimization [34] and particle filtering [32], to help the registration jump out of local minima. While these methods provide improved results, none of them maintains a deterministic and exact optimality. Another class of methods adopt the heuristic hypothesis-and-test idea. Examples include Hough Transform, RANSAC and alignment-based object recognition [17]. They work well in cluttered scenes (*e.g.* the 4PC-S [1]), but the heuristic nature renders their results not exactly optimal.

Registration methods that come with guaranteed optimality were published in the past, though in a smaller number. For example, in 2D cases, branch-and-bound (BnB) has been used for image pattern matching [7, 26, 29]. A truncated L_2 optimization for optimal geometric fitting is recently addressed in [2]. However, most of these methods are focused on the much simpler 2D case. Extending them to 3D and $SE(3)$ is a non-trivial task.

Li and Hartley [23] presented a rotation-search method for 3D-3D registration. While being globally optimal, their method makes unrealistic assumption such as the two pointsets are of equal size and there is only pure rotation. Branch-and-bound based Euclidean registration was investigated in Olsson *et al.* [28] for cases with known correspondences. Enqvist *et al.* [12] converted the registration problem to graph vertex cover and provided an optimal solution. Applications of BnB to other vision geometry problems may be found in [22, 20].

Methods that make use of local invariant shape descriptors (*e.g.* spin image [19], shape contexts [4], EGI [24]) are mostly heuristic and do not address the optimality issue. One exception is the work of Gelfand *et al.* [14] in which they proposed a globally optimal solution on top of the local descriptors. Their idea is based on pair-wise distance consistency similar to [12]. Their optimization is applied to a relatively small number of local descriptors rather than whole point clouds. In contrast, our method (to be described in this paper) requires no local descriptors and directly works on raw point clouds.

In this paper, we solve the 3D Euclidean registration problem with global optimality guarantee. Our method relies on the idea of $SO(3)$ space search, as proposed in [15, 16] and extended in *e.g.* [30, 3]. Most of the existing work along this line are based on L_∞-norm minimization. For the case of optimal registration under L_2-norm, few results are available.

3. The 3D Registration Problem

The standard ICP algorithm solves an L_2-error minimization problem, defined as follows.

Let two 3D pointsets $\mathcal{X} = \{\mathbf{x}_i\}, i = 1, ..., M$ and $\mathcal{Y} = \{\mathbf{y}_j\}, j = 1, ..., N$, where $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^3$ are point coordinates, be the *data* pointset and *model* pointset respectively. The aim 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:

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

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

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

Given initial transformation $\mathbf{R}$ and $\mathbf{t}$, the ICP algorithm iteratively solves the above minimization via alternating between estimating the transformation in Eq. (1), and finding the closest-point matches by Eq. (2). Due to such iterative nature, ICP can only guarantee the convergence to a local minimum.

4. Method Overview

In this work, we seek a truly globally-optimal solution to 3D registration. We choose to use branch-and-bound to solve the global optimization problem. Our method is 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 above BnB search.
Until convergence.

While the idea of using BnB is straightforward, it is *non-trivial* to apply it for the case of rigid 3D registration. Although existing BnB approaches work successfully for 2D registration, extending the success to 3D has been much challenging (see *e.g.*, [16, 12, 23, 12]).

In order to apply BnB to 3D registration, one must first answer the following questions: (i) how to parameterize and branch the domain of 3D motions, (ii) how to efficiently find an upper bound and lower bound.

Domain parametrization. Recall that our goal is to minimize the error E in Eq. (1) over the domain of all feasible 3D motions (*i.e.* the group of $SE(3)$, defined by $SE(3) = SO(3) \times \mathbb{R}^3$). Each member of $SE(3)$ can be minimally parameterized by six parameters. The angle-axis representation is used in this paper to encode rotation, and we use $\mathbf{R_r}$ to denote the rotation matrix with its angle-axis representation to be $\mathbf{r}$, *i.e.* $\mathbf{R_r} = \exp([\,\mathbf{r}\,]_\times)$ where $\exp(\cdot)$ is the matrix exponential and $[\,\cdot\,]_\times$ denotes the skew-symmetric matrix representation. With this representation, the entire space formed by all 3D rotations can be compactly represented as a solid radius-π ball in 3D. 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 the optimal translation must lie within a bounded cube $[-\xi, \xi]^3$ which may be readily set by choosing a big number as ξ. During BnB searches, initial cubes will be subdivided into smaller sub-cubes C_r, C_t using the octree data-structure and the process is repeated.

Bounding functions. We will present the derivation of our new bounding functions (for the ICP L_2-norm metric) in the next section. Worth mentioning here is a unique feature of the proposed BnB method. That is, we employ, as a subroutine, the conventional ICP algorithm in the BnB search computation. This way, our method enjoys both the efficiency provided by the local ICP search, and the optimality guaranteed by the BnB search.

5. Derive New Bounding Functions

As for any BnB method, finding quality bounds is the key to success. In our method, we need to find the bounds of the particular type of L_2-norm error function used in ICP within a domain $C_r \times C_t$. Next, we will introduce the concept of *uncertainty radius* as a mathematical preparation, then derive our new bounds based on it.

5.1. Uncertainty radius

The intuition behind the concept of *uncertainty radius* is: we want to examine, if we perturb a 3D rigid motion with rotation $\mathbf{r} \in C_r$ and/or translation $\mathbf{t} \in C_t$ applied to a 3D point $\mathbf{x}$, what the uncertainty region of the transformed point will be. We aim to find a ball enclosing such an uncertainty region. Our first result is as follows.
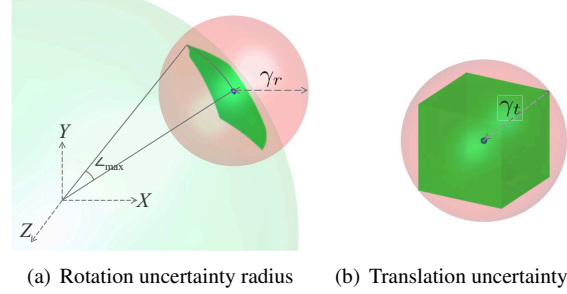

(a) Rotation uncertainty radius   (b) Translation uncertainty radius

Figure 1. 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).

Result 1. *(Rotation uncertainty radius) Given a 3D point* $\mathbf{x}$*. For a rotation cube* C_r *of half side-length* σ_r *with* $\mathbf{r}_0$ *as the center, examining the maximum distance from* $\mathbf{R_r x}$ *to* $\mathbf{R_{r_0} x}$*, we have*

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

Proof. $\|\mathbf{R_r x} - \mathbf{R_{r_0} x}\|$
$$= 2\sin(\angle(\mathbf{R_r x}, \mathbf{R_{r_0} x})/2)\|\mathbf{x}\|$$
$$\leqslant 2\sin(\min(\angle(\mathbf{R_r}, \mathbf{R_{r_0}})/2, \pi/2))\|\mathbf{x}\|$$
$$\leqslant 2\sin(\min(\|\mathbf{r} - \mathbf{r}_0\|/2, \pi/2))\|\mathbf{x}\|$$
$$\leqslant 2\sin(\min(\sqrt{3}\sigma_r/2, \pi/2))\|\mathbf{x}\|.$$

$\square$

The first, and the second inequalities above, are based on Lemma 1, Lemma 2 of paper [16], respectively. For convenience, we summarize both Lemmas in a (single) result shown below.

Result 2. *For any vector* $\mathbf{x}$*, two rotations* $\mathbf{R_r}$ *and* $\mathbf{R_{r_0}}$*, with* $\mathbf{r}$ *and* $\mathbf{r}_0$ *as their angle-axis representations, then we have*

$$\angle(\mathbf{R_r x}, \mathbf{R_{r_0} x}) \leqslant \angle(\mathbf{R_r}, \mathbf{R_{r_0}}) \leqslant \|\mathbf{r} - \mathbf{r}_0\|. \quad (4)$$

The second inequality in Eq. (4) means that, the angular distance between two rotations in the underlying manifold, is less than their vector distance in the angle-axis representation. We call γ_r the rotation uncertainty radius. Similarly, we can derive a translation uncertainty radius γ_t, for a translation cube C_t with half side-length σ_t centered at $\mathbf{t}_0$:

$$\|(\mathbf{x} + \mathbf{t}) - (\mathbf{x} + \mathbf{t}_0)\| = \|\mathbf{t} - \mathbf{t}_0\| \leqslant \sqrt{3}\sigma_t \doteq \gamma_t. \quad (5)$$

See Fig. 1 for illustrations of γ_r and γ_t. Both uncertainty radii are used in deriving the lower bound for our method. Note that γ_r is point-dependent, therefore γ_{ri} refers to the rotation uncertainty radius at $\mathbf{x}_i$.
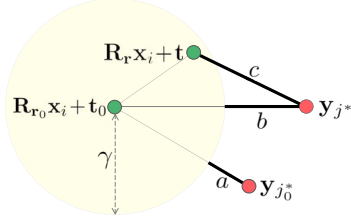
1459

Figure 2. An illustrative figure for the obtained lower bound. It is clear that $a \le b \le c$ while $a = \underline{e_i}$ and $c = e_i(\mathbf{R_r}, \mathbf{t})$. See text for details.

5.2. Bounding the L_2 error

Given a 3D data point $\mathbf{x}_i$, a rotation cube C_r centered at $\mathbf{r}_0$ and a translation cube C_t centered at $\mathbf{t}_0$, an *upper bound* for the per-pixel residual error $e_i(\mathbf{R}, \mathbf{t})$ within the cubes can be trivially found as

$$\overline{e_i} \doteq e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) \geqslant \min_{\forall(\mathbf{r}, \mathbf{t}) \in (C_r \times C_t)} e_i(\mathbf{R_r}, \mathbf{t}). \quad (6)$$

Finding a suitable lower bound for this L_2 residual error appears to be a harder task, especially considering the domain is in $SE(3)$. However, below we will show how a lower bound can be found, using the concept of uncertainty radius.

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

$$
\begin{aligned}
& e_i(\mathbf{R_r}, \mathbf{t}) \\
&= \|\mathbf{R_r}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_{j^*}\| \\
&= \|(\mathbf{R_{r_0}}\mathbf{x}_i + \mathbf{t}_0 - \mathbf{y}_{j^*}) + (\mathbf{R_r}\mathbf{x}_i - \mathbf{R_{r_0}}\mathbf{x}_i) + (\mathbf{t} - \mathbf{t}_0)\| \\
&\geqslant \|\mathbf{R_{r_0}}\mathbf{x}_i + \mathbf{t}_0 - \mathbf{y}_{j^*}\| - (\|\mathbf{R_r}\mathbf{x}_i - \mathbf{R_{r_0}}\mathbf{x}_i\| + \|\mathbf{t} - \mathbf{t}_0\|) \quad (7) \\
&\geqslant \|\mathbf{R_{r_0}}\mathbf{x}_i + \mathbf{t}_0 - \mathbf{y}_{j^*}\| - (\gamma_{r_i} + \gamma_t) \quad (8) \\
&\geqslant \|\mathbf{R_{r_0}}\mathbf{x}_i + \mathbf{t}_0 - \mathbf{y}_{j_0^*}\| - (\gamma_{r_i} + \gamma_t) \quad (9) \\
&= e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{r_i} + \gamma_t),
\end{aligned}
$$

where Eq. (7) follows from the (reverse) triangle inequality, Eq. (8) is based on the uncertainty radii in Eq. (3) and Eq. (5), and Eq. (9) is based on the closest-point definition. Note that, $\mathbf{y}_{j^*}$ is not a fixed point, but changes dynamically as a function of $(\mathbf{r}, \mathbf{t})$ as defined in Eq. (2). Such a closest-point mechanism is consistent with standard ICP.

Now we have reached a lower bound of the per-point residual for $C_r \times C_t$ as

$$\underline{e_i} \doteq \max(e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{r_i} + \gamma_t), 0) \leqslant \min_{\forall(\mathbf{r}, \mathbf{t}) \in (C_r \times C_t)} e_i(\mathbf{R_r}, \mathbf{t}). \quad (10)$$

The geometric explanation of this lower bound is as follows. Since $\mathbf{y}_{j_0^*}$ is the closest point to the center $\mathbf{R_{r_0}}\mathbf{x}_i + \mathbf{t}_0$ of the uncertainty ball with radius $\gamma = \gamma_{r_i} + \gamma_t$, it is also the closest point to (the surface of) the ball and $\underline{e_i}$ is the closest

distance between pointset $\mathcal{Y}$ and the ball. Thus no matter where the transformed data point $\mathbf{R_r}\mathbf{x}_i + \mathbf{t}$ lies inside the ball, its closest distance to pointset $\mathcal{Y}$ will be no less than $\underline{e_i}$. See Fig. 2 for an illustration.

Summing up the squared upper bounds and lower bounds of the per-point residuals in Eq. (6) and Eq. (10) for all the M points, we get the important result below.

Result 3. *(Bounds of the 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_{r_i}s 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*

$$\overline{E} \doteq \sum_{i=1}^{M} \overline{e_i}^2 = \sum_{i=1}^{M} e_i(\mathbf{R_{r_0}}, \mathbf{t}_0)^2, \quad (11)$$

$$\underline{E} \doteq \sum_{i=1}^{M} \underline{e_i}^2 = \sum_{i=1}^{M} \max(e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{r_i} + \gamma_t), 0)^2. \quad (12)$$

This result gives both the upper bound and lower bound of the registration error, based on which we developed our Go-ICP algorithm.

6. The Go-ICP Algorithm

6.1. Nested BnBs

Instead of searching the 6D space of $SE(3)$ with a single BnB which would be less efficient, we propose to use a nested BnB search structure. An outer BnB searches the rotation space of $SO(3)$, while solving the bounds and corresponding optimal translation by calling an inner translational BnB. The bounds for both BnB algorithms can be readily derived according to Sec. 5.2 and will be briefly described as follows. In the outer rotation BnB, for a cube C_r the bounds of the registration error can be chosen as $\overline{E}_r = \min_{\forall \mathbf{t} \in \mathcal{C}_t} \sum_i e_i(\mathbf{R_{r_0}}, \mathbf{t})^2$ and $\underline{E}_r = \min_{\forall \mathbf{t} \in \mathcal{C}_t} \sum_i \max(e_i(\mathbf{R_{r_0}}, \mathbf{t}) - \gamma_{r_i}, 0)^2$ where $\mathcal{C}_t$ is the initial translation cube. To solve $\underline{E}_r$ with the inner translation BnB, the bounds for a translation cube C_t can be chosen as $\overline{E}_t = \sum_i \max(e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - \gamma_{r_i}, 0)^2$ and $\underline{E}_t = \sum_i \max(e_i(\mathbf{R_{r_0}}, \mathbf{t}_0) - (\gamma_{r_i} + \gamma_t), 0)^2$. By setting all the rotation uncertainty radii γ_{r_i} to be zero in $\overline{E}_t$ and $\underline{E}_t$ above, we get the translation BnB to solve $\overline{E}_r$.

A detailed description is given in Algorithm 1 (Go-ICP – the Main Algorithm) and Algorithm 2 (the Translation BnB). The nested BnB structure can be clearly seen: the outer BnB in Algorithm 1 calls the inner BnB in Algorithm 2.

Search strategy and stop criterion. In both BnBs, we perform *best-first-search* strategy. Specifically, each of the BnBs maintains a *priority queue* Q_r, Q_t, respectively. The priority of the cubes in the queue is opposite to their lower bounds, which means that the BnBs always explore the cube

1460

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. Set $E^* = +\infty$.
2 **loop**
3   Read out a cube with lowest lower-bound $\underline{E}_r$ from Q_r.
4   Quit the loop if $E^* - \underline{E}_r < \epsilon$.
5   Divide the cube into 8 sub-cubes.
6   **foreach** sub-cube C_r **do**
7   Compute $\overline{E}_r$ for C_r and corresponding optimal $\mathbf{t}$ by calling Algorithm 2 with $\mathbf{r}_0$, zero uncertainty radius, E^*.
8   **if** $\overline{E}_r < E^*$ **then**
9   Run ICP with initialization of $(\mathbf{r}_0, \mathbf{t})$.
10   Update E^*, $\mathbf{r}^*$, and $\mathbf{t}^*$ with the results of ICP.
11   **end**
12   Compute $\underline{E}_r$ for C_r by calling Algorithm 2 with $\mathbf{r}_0$, $\boldsymbol{\gamma}_r$ and E^*.
13   Discard C_r if $\underline{E}_r \geqslant E^*$; otherwise put it into Q_r.
14   **end**
15 **end**

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 $\boldsymbol{\gamma}_r$, 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. Set $E_t^* = E^*$.
2 **loop**
3   Read out a cube with lowest lower-bound $\underline{E}_t$ from Q_t.
4   Quit the loop if $E_t^* - \underline{E}_t < \epsilon$.
5   Divide the cube into 8 sub-cubes.
6   **foreach** sub-cube C_t **do**
7   Compute $\overline{E}_t$ for C_t with $\mathbf{r}_0, \mathbf{t}_0$ and $\boldsymbol{\gamma}_r$.
8   **if** $\overline{E}_t < E_t^*$ **then**
9   Update $E_t^* = \overline{E}_t, \mathbf{t}^* = \mathbf{t}_0$.
10   **end**
11   Compute $\underline{E}_t$ for C_t with $\mathbf{r}_0, \mathbf{t}_0, \boldsymbol{\gamma}_r, \gamma_t$.
12   Discard C_t if $\underline{E}_t \geqslant E_t^*$; otherwise put it into Q_t.
13   **end**
14 **end**

with smallest lower bound in the queue. Once the difference between so-far-the-best error E^* and the lower bound $\underline{E}$ of current cube is less than a threshold ϵ, the BnB stops.

Proof of convergence. The convergence for both of the algorithms is provable. All we need to do is to show that, as the algorithm iterates, the gap between the lower bound and the upper bound converges uniformly to zero. This is easy to see, as when the side-lengths of all cubes asymptotically diminish to zero, the gap between the two bounds, *i.e.* the uncertainty radii in Eq. (3) and Eq. (5), will be zero too.

Initial error for inner BnB. Here we give some details regarding the initial error setting in Line 1 of Algorithm 2. To speed up the computation of inner BnB, we set the initial E_t^* to be E^* without loss of globally optimal registration based on the following insight. If $\overline{E}_r = E_t^* \geq E^*$, then E^* will not be updated. If $\underline{E}_r = E_t^* \geq E^*$, then C_r contains no better solution. In other words, if the error E_t^* returned by the inner BnB is greater than or equal to so-far-the-best error E^* in the outer BnB, it makes no contribution.

6.2. Integration with local ICP

Lines 9-10 of Algorithm 1 describe our upper-bound refinement procedure based on a standard local ICP. This procedure is done as follows. Whenever the outer BnB finds a cube C_r which has an upper-bound lower than the so-far-the-best function value, it will then call the conventional ICP to start over, from the center of C_r and corresponding t^* as the new initial transformation. This helps ICP jump out of the previous local minima. Once ICP converges, it will arrive at a new local minimum which has a lower function value. This new local minimum is used to update the upper bound.
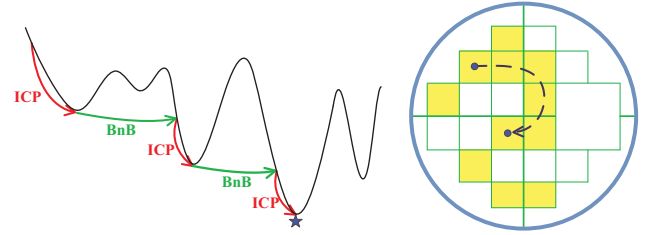

Figure 3. **Left**: BnB and ICP collaboratively update the upper bounds during the search process. **Right**: with the guidance of BnB, ICP only explores un-discarded promising cubes with small lower bounds marked up by BnB.

Figure 3 (left) illustrates the collaborative relationship between ICP and BnB. Under the guidance of the global BnB, the local ICP seems to have a strong "sense of direction". Instead of exploring the domain blindly, ICP converges into local minima one by one with each local minimum having lower error than the previous one, and reaches the global minimum in the end. Since ICP monotonically decreases current-best error E^* (cf. [5]), all points (transformation parameter $\mathbf{r}, \mathbf{t}$) in its search path should have error lower than E^*, which means that lower bounds of the cubes containing these points should be lower than E^*. Thus the search path of the local ICP is entirely confined to the undiscarded, promising cubes with small lower bounds, as illustrated in Fig. 3 (right).

This way, both the global BnB search and the local ICP search are intimately integrated in our method. The former not only helps the latter to jump out of local minima, but also provides a guidance for the latter's next search; the latter accelerates the former's convergence by refining the upper bound, hence improves the overall efficiency.
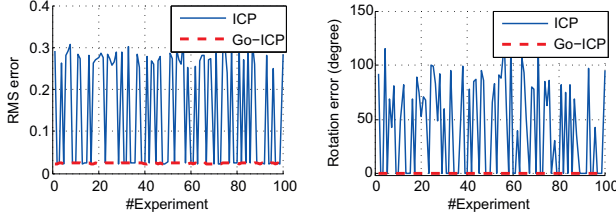
Figure 4. Comparison of the registration error (Left) and rotation error (Right) on random points, by our Go-ICP method, versus ICP initialized with Identity rotation and zero translation. Ground-truth rotation and translation lie randomly within ±100 degrees and ±0.5 respectively.

7. Experiments

This section reports our experimental results of the Go-ICP algorithm, on both synthetic data and real range surfaces. All our codes are implemented in C++, and tested on a standard PC with Intel i7 3.4GHz CPU. To speed up the closest-point computation one may use kd-tree or Distance Transform (DT). Similar to [13], we use 3D Euclidean DT with $300\times300\times300$ grids. In all the experiments reported below, we pre-normalized the pointsets such that all the points are within the domain of $[-1,1]^3$ and the initial transformation domain to explore is set to be $[-\pi,\pi]^3 \times [-0.5,0.5]^3$. Except for the 3D object localization experiment, we set the convergence threshold ϵ to be $0.001 * M$ in all the tests. It is worth mentioning that, thanks to the inner ICP, the convergence threshold can be set reasonably large for BnB to converge fast. We organize our experimental results below according to their purposes.

7.1. Optimality

The purpose of this first experiment is to verify the global optimality of our new Go-ICP algorithm and compare that with standard ICP. We repeat 100 tests on random points. In each test, 100 model points are randomly drawn from the uniform distribution in $[-1,1]^3$; rotation and translation are randomly drawn within ±100 degrees and ±0.5 respectively and applied to the model points to generate data points; zero mean Gaussian noise is added to the points; ICP is initialized with Identity rotation and zero translation.

Figure 4 shows the final reported registration error and rotation errors for the 100 runs. Note that our goal is to minimize the L_2 error while root-mean-square (RMS) error is reported for better comprehension. It is clear that our Go-ICP always outperforms the classic ICP, in terms of having consistently lower residual error. Hence the optimality is confirmed. The first row in Fig. 5 compares the registration results of the first run. Visually inspected, our method yields a more satisfactory registration too.
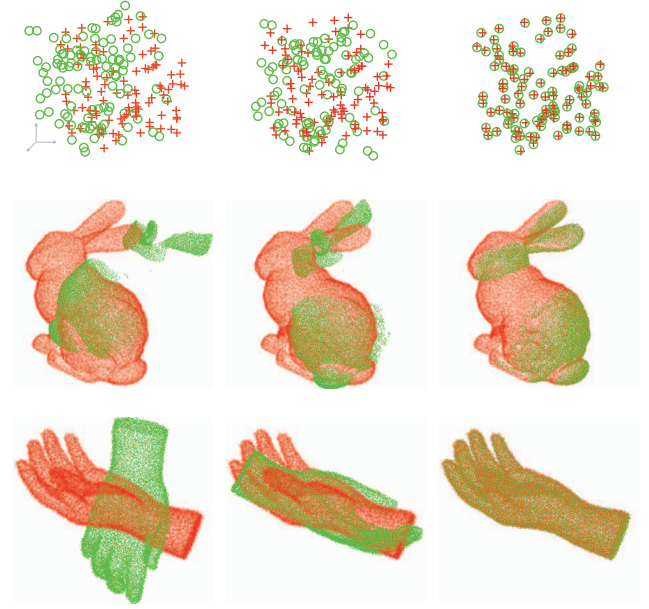

Figure 5. Visual comparison of registration results. **Left:** initial pose. **Middle:** results by ICP. **Right:** results by our Go-ICP.

7.2. Running time

In this experiment, we use the Stanford bunny raw scan data[1] and a dense hand mesh[2] shown in the last two rows of Fig. 5 to test out the real-life efficiency. We test the running time of our method on different numbers of data points (*i.e.* M) by sub-sampling the original data, while the initial poses are fixed.

As presented in Fig. 6, the running time of Go-ICP manifests a linear trend, which is due to our linear convergence threshold setting $w.r.t$ number of data points, and the $O(1)$ closest-point distance retrieval from DT. Overall, the Go-ICP algorithm is efficient. In our experiment, for example, to match 1000 data points to about 30,000–40,000 model points took about 30 seconds for bunny and 15 seconds for the hand.

7.3. Convergence of bounds

To show the convergence of our method and demonstrate the evolution of the bounds, we record the upper and lower bound values of the outer BnB when registering the 1000 data points onto the model pointsets in the previous experiment, and plot them as a function of time as shown in Fig. 7. Note that, the *global* bounds are plotted. The global lower bound, which is the smallest lower bound of the cubes in the queue, is always close to zero because the globally optimal error is close to zero, which is true for the registration problem. It can be seen that, BnB and ICP collaboratively

---

[1] http://graphics.stanford.edu/data/3Dscanrep/
[2] http://fastscan3d.com/download/samples/
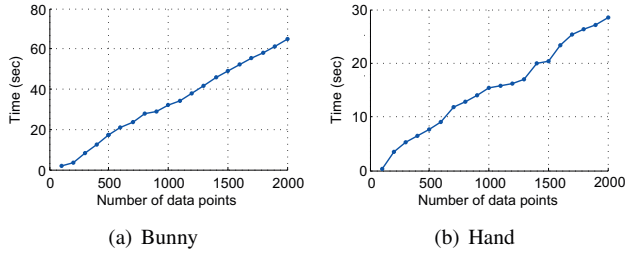
1462

(a) Bunny   (b) Hand

Figure 6. Typical execution time of our method on the bunny and hand *w.r.t.* different numbers of data points (M). Initial poses are shown in the last two rows of Fig. 5.
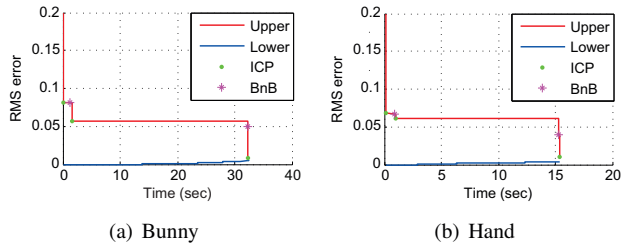

(a) Bunny   (b) Hand

Figure 7. Typical convergence curves of the upper bounds and lower bounds in the outer BnB of our method on bunny and hand ($M = 1000$).

(a) 10% outliers   (b) 20% outliers

Figure 8. Tests of Go-ICP with trimming under outliers. (Better viewed in color)

update the upper bound. ICP refines the better upper-bound found by BnB with local search, and BnB guides ICP to converge into multiple local minima with lower and lower registration errors until the global minimum is reached.

7.4. Outlier handling

This experiment aims to test the performance of our method under outliers. Since ICP is based on least-squares fitting, it is not inherently robust to outliers. However, our Go-ICP can be easily extended to using a robust error function. In this test, we use ICP with trimming [11] and the same trimming strategy in BnB to handle outliers. The extended lower bound is described in Sec. 8.

Different numbers of outliers (10% and 20%) are added into the Bunny data points, and we set the trimming percentage of our Go-ICP to be 20%. Figure 8 visually shows the effectiveness of our trimmed Go-ICP in attaining the optimum despite of the presence of the outliers.

7.5. 3D object localization

In this experiment, we show how our method may be used for model-based 3D object detection, localization and pose estimation, from a cluttered range scan (*e.g.* that obtained by a Kinect, or a laser scanner). The RGB-D database from [21] is used. Our goal is to register the points of a baseball cap to the point cloud of the scene as shown in Fig. 9. Note that this is an extremely hard task, as there
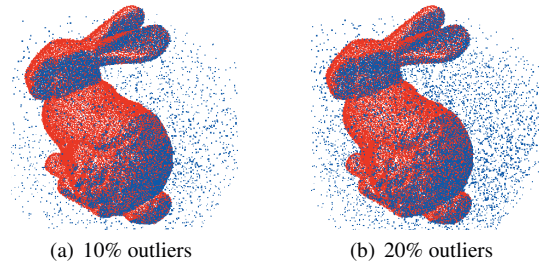
are numerous local minima (with very low registration error) arising when the cap is nestled into the table, the wall, etc. Neither visual information nor local descriptors were used; the inputs were two point clouds only.

We sampled 100 points on the cap, and the convergence threshold was set to be $0.00001 * M = 0.001$. Our Go-ICP successfully localized the cap in the scene with accurate pose estimation within 40 seconds. The final RMS error is 3.9mm.

8. Extensions

This section gives some ideas of how to extend our method to various other variants of ICP. We focus on a few examples, showing mainly how the new lower bound (for $C_r \times C_t$) may be derived for these extended cases.

LM-ICP with robust M-estimator kernel. With little change, our algorithm can be directly adapted to Levenberg-Marquardt ICP [13]. Even the DT data structure in LM-ICP can be shared by the BnB. The new lower-bound function is simply a robust kernelized version of our original lower bound.

Trimmed ICP. In [11], only a subset $\mathcal{P}$ of the data points with smallest closest-point distances is used for motion computation, hence improving the robustness. The new lower bound can be derived as $\underline{E} = \sum_{i \in \mathcal{Q}} \underline{e_i}^2 \leq \sum_{i \in \mathcal{Q}} e_i^2 \leq \sum_{i \in \mathcal{P}} e_i^2 = E$, where $\mathcal{Q}$ is the trimmed subset of $\{e_i\}$ with $\#\mathcal{Q} = \#\mathcal{P}$. In the same spirit, other variants of ICP such as [9] or [25] can be similarly handled.

L_p-norm ICP. The same approach of the work may be extended to other L_p-norm based ICP variants, such as the robustness/sparseness promoting L_1 norm. We leave this as future work.

9. Closing Remarks

We have described a truly globally-optimal solution to Euclidean registration in 3D, under the L_2-norm error
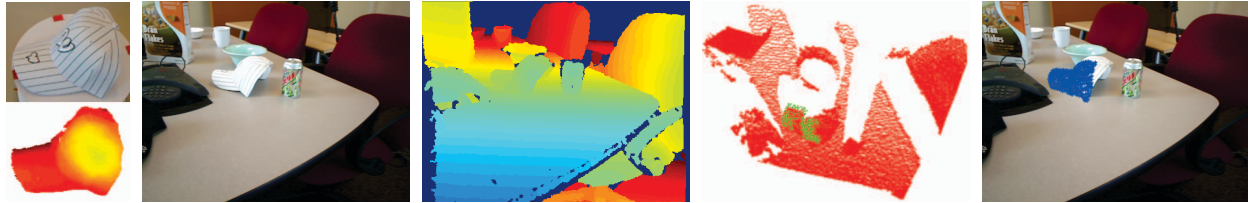
Figure 9. 3D object detection and localization experiment. From left to right: a labeled object and its depth image; an RGB image and a depth image of the scene; registration result showing 3D points and 2D image points. The object is optimally localized, with an accurate pose estimated. Note that no appearance information (*e.g.* feature descriptors) was used, and there exists a large number of local minima.

metric which is the very first solution of this kind. Our algorithm is especially useful for practical scenarios where having an exact optimal solution is highly desirable. Despite being a branch-and-bound based method, it works rather efficiently. For certain applications where real-time performance is not critical, our algorithm can be readily applied, or used as an optimality benchmark.

References

[1] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *SIGGRAPH*, 2008. 2

[2] E. Ask, O. Enqvist, and F. Kahl. Optimal geometric fitting under the truncated L_2-norm. In *CVPR*, 2013. 2

[3] J.-C. Bazin, Y. Seo, and M. Pollefeys. Globally optimal consensus set maximization through rotation search. In *ACCV*, 2012. 2

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *T-PAMI*, 2002. 2

[5] P. Besl and N. McKay. A method for registration of 3-D shapes. *T-PAMI*, 1992. 1, 5

[6] D. Breitenreicher and C. Schnörr. Model-based multiple rigid object detection and registration in unstructured range data. *IJCV*, 2011. 2

[7] T. M. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *CVIU*, 2003. 2

[8] U. Castellani and A. Bartoli. 3D shape registration. *3D Imaging, Analysis, and Applications*, 2012. 2

[9] G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3D object localization. In *CVPR*, 1992. 7

[10] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *ICRA*, 1991. 1

[11] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *ICPR*, 2002. 7

[12] O. Enqvist, K. Josephson, and F. Kahl. Optimal correspondences from pairwise constraints. In *ICCV*, 2009. 2, 3

[13] A. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 2003. 2, 6, 7

[14] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Eurographics Symposium on Geometry Processing*, 2005. 2

[15] R. I. Hartley and F. Kahl. Global optimization through searching rotation space and optimal estimation of the essential matrix. In *ICCV*, 2007. 2

[16] R. I. Hartley and F. Kahl. Global optimization through rotation space search. *IJCV*, 2009. 2, 3

[17] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *T-PAMI*, 1993. 2

[18] B. Jian and B. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *ICCV*, 2005. 2

[19] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *T-PAMI*, 1999. 2

[20] J.-H. Kim, H. Li, and R. Hartley. Motion estimation for multi-camera systems using global optimization. *CVPR*, 2008. 2

[21] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011. 7

[22] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. *ICCV*, 2009. 2

[23] H. Li and R. Hartley. The 3D-3D registration problem revisited. In *ICCV*, 2007. 2, 3

[24] A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3D point clouds. In *CVPR*, 2006. 2

[25] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. In *CAD-Based Vision Workshop*, 1994. 7

[26] D. M. Mount, N. S. Netanyahu, and J. Le Moigne. Efficient algorithms for robust feature matching. *Pattern recognition*, 1999. 2

[27] A. Myronenko and X. Song. Point set registration: coherent point drift. *T-PAMI*, 2010. 2

[28] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *T-PAMI*, 2009. 2

[29] F. Pfeuffer, M. Stiglmayr, and K. Klamroth. Discrete and geometric branch and bound algorithms for medical image registration. *Annals of Operations Research*, 2012. 2

[30] T. Ruland, T. Pajdla, and L. Kruger. Globally optimal hand-eye calibration. In *CVPR*, 2012. 2

[31] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *3DIM*, 2001. 2

[32] R. Sandhu, S. Dambreville, and A. Tannenbaum. Point set registration via particle filtering and stochastic dynamics. *T-PAMI*, 2010. 2

[33] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. *ECCV*, 2004. 2

[34] 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 Transactions on Evolutionary Computation*, 2004. 2

[35] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 1994. 1