Efficient Global Point Cloud Registration by Matching Rotation Invariant Features Through Translation Search

Yinlong Liu[1,2*][0000−0002−6468−8233], Chen Wang[1,2*][0000−0003−2922−5345],
Zhijian Song[1,2**][0000−0001−9873−216X], and Manning
Wang[1,2**][0000−0002−9255−3897]

[1] Digital Medical Research Center, School of Basic Medical Science, Fudan
University, Shanghai 200032, China
{yinlongliu15, wangchen17, zjsong, mnwang}@fudan.edu.cn
[2] Shanghai Key Laboratory of Medical Imaging Computing and Computer Assisted
Intervention, Shanghai 200032, China

Abstract. Three-dimensional rigid point cloud registration has many
applications in computer vision and robotics. Local methods tend to fail,
causing global methods to be needed, when the relative transformation
is large or the overlap ratio is small. Most existing global methods utilize
BnB optimization over the 6D parameter space of $SE(3)$. Such methods
are usually very slow because the time complexity of BnB optimization is
exponential in the dimensionality of the parameter space. In this paper,
we decouple the optimization of translation and rotation, and we propose
a fast BnB algorithm to globally optimize the 3D translation parameter
first. The optimal rotation is then calculated by utilizing the global opti-
mal translation found by the BnB algorithm. The separate optimization
of translation and rotation is realized by using a newly proposed rota-
tion invariant feature. Experiments on challenging data sets demonstrate
that the proposed method outperforms state-of-the-art global methods
in terms of both speed and accuracy.

Keywords: point cloud registration · global optimization · rotation in-
variant feature

1 Introduction

Three-dimensional rigid point cloud registration is a common problem in fields
such as computer vision, robotics, and computer-assisted intervention[1–4]. Tra-
ditional local methods suffice only when the relative transformation is small and
there is a large proportion of true overlap[1–3]. In contrast, a global method is
needed when there is a large relative transformation or when the overlap pro-
portion is small. Example applications include loop closing in simultaneous lo-
calization and mapping (SLAM)[1] and spatial registration in computer-assisted

---

⋆ The first two authors contributed equally to this work.
⋆⋆ The corresponding authors are Zhijian Song and Manning Wang.

intervention[5]. In recent years, there has been a surge in the use of branch-and-bound (BnB) optimization to globally register 3D point clouds. However, the time complexity of BnB optimization is exponential in the dimensionality of the problem. Most existing global methods optimize an objective function in the parameter space of $SE(3)$, which has a dimensionality of six; thus, they are usually very slow.

Motivated by [6], in which $SE(3)$ is decoupled into $SO(3)$ and R^3 and the rotation and translation are optimized separately in $SO(3)$ and R^3, respectively, we also optimize the rotation and translation separately in lower-dimensional spaces to achieve high efficiency via a newly proposed rotation invariant feature. In the method presented in [6], BnB optimization is first used to globally optimize the rotation to align translation invariant features, namely, the surface normal distributions, constructed from the original point clouds. Once the globally optimal rotation in $SO(3)$ has been obtained, BnB optimization in R^3 is performed to calculate the translation. In contrast to [6], we propose a new rotation invariant feature (RIF) that allows us to first globally optimize the translation in R^3 to align the features computed for the original point clouds. The RIF we propose is a triple constructed from a pair of points. The first two elements of the RIF are the distances from the two points to the origin, and the third element is the distance between the two points. Conceptually, the elements of this RIF are the edge lengths of the triangle formed by the two points and the origin, which are obviously invariant with respect to the rotation of the two points around the origin. We maximize an objective function defined in terms of the consensus set between the two RIF sets constructed from the two point clouds to be registered and derive an upper bound on this objective function over the parameter space of translation in R^3. An efficient BnB optimization algorithm is developed based on this upper bound to calculate the translation with guaranteed global optimality. Experiments using real data demonstrate the efficiency and accuracy of the proposed 6D rigid point cloud registration method under challenging real-world conditions, such as large relative transformation and partial overlap.

2   Related Work

Given two 3D point clouds $\mathcal{X}$ and $\mathcal{Y}$, performing 3D rigid registration between them is the process of finding an optimal transformation $\mathbf{T} \in SE(3)$ to align them, as follows:

$$\mathbf{T}^* = \underset{\mathbf{T} \in SE(3)}{\arg\max} \quad O(\mathcal{Y}, \mathbf{T}(\mathcal{X})) \tag{1}$$

where $\mathcal{X}$ is the moving point cloud, $\mathcal{Y}$ is the reference point cloud, and O is a function that measures the alignment. If some correspondences between these two point clouds are known, then the transformation can be robustly calculated analytically. However, this registration becomes a difficult optimization problem if no correspondence is known because the objective function O is highly non-convex.

2.1   Local Methods

As a non-convex optimization problem, point cloud registration is first locally solved with iterative approaches, the most popular of which is the iterative closest point (ICP) method[7]. The ICP method starts from an initial transformation and iterates between finding correspondences with the current transformation and updating the transformation with the newly established correspondences. The optimization scheme of the ICP algorithm is of the expectation maximization (EM) type and can converge only to a local optimum. In addition, the original ICP algorithm finds point-to-point correspondences in accordance with the L2 distance by minimizing the average distance between the corresponding point pairs found during iteration. The objective function and optimization scheme of the ICP algorithm make it very sensitive to outliers, and its convergence basin is very small.

A great number of variants have been proposed to improve the convergence and robustness of the ICP algorithm[8]. Another line of research for improving the convergence is to replace the objective function of the ICP algorithm with a new objective function defined in terms of the probability density[9]. In these methods, the probability densities are constructed from the original point clouds by means of kernels, such as the Gaussian kernel, and the difference between the two probability densities constructed from the two point clouds to be registered is minimized. This kind of objective function can be made much smoother than the ICP objective function can, thereby enlarging the basin of convergence. However, all these methods can still converge only to a local optimum.

2.2   Global Methods

The first attempts at global point cloud registration used heuristic methods, such as simulated annealing[10] or particle swarm optimization[11]. Methods of this kind have an increased probability of reaching the global optimum regardless of the initialization conditions. However, global optimality cannot be guaranteed.

The current trend is to solve the global point cloud registration problem using guaranteed global optimization methods. Most of these methods use BnB optimization, and almost every newly developed method involves deriving a new bound on the objective function. [12] is a pioneer work on global point cloud registration in which the point cloud registration problem is parameterized in terms of both transformation and correspondence and a simplified problem under rotation alone is solved by combining BnB optimization with Lipschitz optimization. Go-ICP[13] is the first practical global point cloud registration method. Based on a lemma presented in [14], which states that when a vector is rotated through two rotations, the angular difference between the two new vectors is no greater than the angular difference between the two rotations, Go-ICP establishes a geometric bound on a point rotated through an arbitrary rotation in a cubic branch of $SO(3)$. Go-ICP optimizes the same objective function as that of the traditional ICP algorithm via BnB optimization, and a trimming technique is used to address outliers. [15] proposes a tighter bound and optimizes a more robust

objective function based on a consensus set, and this bound is further tightened in [16]. Another way to address outliers is to define the objective function in terms of the probability density, as is done in GOGMA[17], which minimizes the L2 distance between the two probability densities constructed from the original point clouds with Gaussian kernels.

The above global point cloud registration methods optimize an objective function in the six-dimensional space of $SE(3)$. Asymmetric point matching (APM) optimizes an objective function defined in terms of both transformation and point correspondence matrix[18]. Although the dimensionality of the original parameter space is very high, a bound is developed in a lower-dimensional space, which has the same dimensionality as that of the spatial transformation, by assuming a point-to-point correspondence. This assumption makes it difficult for APM to register partially overlapping point clouds or point clouds with gross outliers. The advantage of this method is that it can perform affine registration.

The time complexity of BnB optimization is exponential in the dimensionality of the problem. Therefore, existing global point cloud registration methods are usually very slow, primarily because the parameter space of $SE(3)$ has six dimensions. One way to improve the time efficiency is to decouple $SE(3)$ into $SO(3)$ and R^3 and then optimize the 3D rotation and the 3D translation separately. [6] utilizes translation invariant features to first optimize the 3D rotation and then optimizes the 3D translation using the calculated globally optimal rotation; however, the process of constructing the translation invariant features is very time consuming.

2.3   Contribution

This paper also optimizes rotation and translation separately to achieve high efficiency. Our first contribution is that we propose a simple RIF that allows us to first globally search for the translation between two point clouds to align the features constructed from the original point clouds. We define a robust objective function based on a consensus set and derive a tight bound to achieve fast BnB optimization for the translation search. Second, we develop a 6D point cloud registration algorithm on the basis of the global translation search. Experiments on challenging real data demonstrate the superiority of the proposed method over state-of-the-art global methods in terms of both run time and accuracy.

3   Method

3.1   Rotation Invariant Feature

Let $\mathcal{Y} = \{\mathbf{y}_j\}_{j=1}^Y$ and $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^X$ be two 3D point clouds, which are related by a 3D rigid transformation from $\mathcal{X}$ to $\mathcal{Y}$. For a pair of corresponding points $\mathbf{y}_j$ and $\mathbf{x}_i$, we have

$$\mathbf{y}_j = \mathbf{R}^*(\mathbf{x}_i + \mathbf{t}^*) \tag{2}$$

where $\mathbf{t}^* \in R^3$ and $\mathbf{R}^* \in SO(3)$ are the translation and rotation, respectively, from $\mathcal{X}$ to $\mathcal{Y}$.

For a pair of points $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$ from the moving point cloud, we propose the construction of a triple $\{\|\mathbf{x}_{i1}\|, \|\mathbf{x}_{i2}\|, \|\mathbf{x}_{i1} - \mathbf{x}_{i2}\|\}$, where $\| \cdot \|$ denotes the Euclidean norm in R^3. As shown in Fig. 1, the three elements of this triple are the edge lengths of the triangle formed by the two points $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ and the origin. Obviously, this triple is invariant with respect to rotation around the origin, and thus, we call it a RIF. This means that for any $\mathbf{R} \in SO(3)$, we have

$$\left\{ \begin{array}{c} \|\mathbf{R}\mathbf{x}_{i1}\| \\ \|\mathbf{R}\mathbf{x}_{i2}\| \\ \|\mathbf{R}(\mathbf{x}_{i1} - \mathbf{x}_{i2})\| \end{array} \right\} = \left\{ \begin{array}{c} \|\mathbf{x}_{i1}\| \\ \|\mathbf{x}_{i2}\| \\ \|\mathbf{x}_{i1} - \mathbf{x}_{i2}\| \end{array} \right\} \tag{3}$$

Figure 1 (b-d) show the changes in RIFs with the translation of points.

(a)   (b)   (c)   (d)
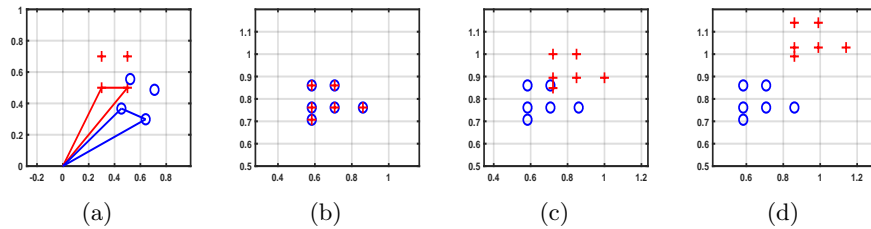
Fig. 1. Illustration of RIFs and how they change with translation. Only the first two dimensions of the RIFs are illustrated because the third dimension does not change with the translation of the points. (a) Four reference points (blue circles) and four moving points (red crosses). There is a relative rotation but no relative translation between the two point clouds. The two triangles represent two RIFs constructed from corresponding point pairs from the reference and moving point clouds. (b) RIFs constructed from the reference and moving clouds in (a). The two sets of RIFs match each other because there is no relative translation between the reference point cloud and the moving point cloud. (c), (d) The RIFs when there are relative translations of (0.1, 0.1) and (0.2, 0.2), respectively, between the two point clouds.

3.2   Objective Function for the Translation Search

We use $\mathbf{p}_{i1,i2} = \{\|\mathbf{x}_{i1}\|, \|\mathbf{x}_{i2}\|, \|\mathbf{x}_{i1} - \mathbf{x}_{i2}\|\}$ to denote the RIF constructed from a pair of moving points $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$ and use $\mathbf{q}_{j1,j2} = \{\|\mathbf{y}_{j1}\|, \|\mathbf{y}_{j2}\|, \|\mathbf{y}_{j1} - \mathbf{y}_{j2}\|\}$ to denote the RIF constructed from a pair of reference points $\{\mathbf{y}_{j1}, \mathbf{y}_{j2}\}$. Following equation (3), if the two point pairs are related by a translation $\mathbf{t}$ and a rotation $\mathbf{R}$, we have

$$\mathbf{q}_{j1,j2} = \left\{ \begin{array}{c} \|\mathbf{y}_{j1}\| \\ \|\mathbf{y}_{j2}\| \\ \|\mathbf{y}_{j1} - \mathbf{y}_{j2}\| \end{array} \right\} = \left\{ \begin{array}{c} \|\mathbf{R}(\mathbf{x}_{i1} + \mathbf{t})\| \\ \|\mathbf{R}(\mathbf{x}_{i2} + \mathbf{t})\| \\ \|\mathbf{R}(\mathbf{x}_{i1} - \mathbf{x}_{i2})\| \end{array} \right\} = \left\{ \begin{array}{c} \|\mathbf{x}_{i1} + \mathbf{t}\| \\ \|\mathbf{x}_{i2} + \mathbf{t}\| \\ \|\mathbf{x}_{i1} - \mathbf{x}_{i2}\| \end{array} \right\} := F(\mathbf{p}_{i1,i2}, \mathbf{t}) \tag{4}$$

We find that the two RIFs are related by the translation $\mathbf{t}$, and we define the function $F(\mathbf{t})$ to express this relationship. The parameter space of the function $F(\mathbf{t})$ is R^3. This means that if two point clouds are related by a translation $\mathbf{t}$ and a rotation $\mathbf{R}$, then the RIFs constructed from corresponding point pairs are related only by the translation $\mathbf{t}$ through the function $F(\mathbf{t})$.

Let $\mathcal{Q} = \{\mathbf{q}_n\}_{n=1}^{Q}$ be the set of RIFs constructed from point pairs $\{\mathbf{y}_{j1}, \mathbf{y}_{j2}\}$, where $\mathbf{y}_{j1}, \mathbf{y}_{j2} \in \mathcal{Y}$ and $j1 \neq j2$. Let $\mathcal{P} = \{\mathbf{p}_m\}_{m=1}^{P}$ be the set of RIFs constructed from point pairs $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}\}$, where $\mathbf{x}_{i1}, \mathbf{x}_{i2} \in \mathcal{X}$ and $i1 \neq i2$. The problem of finding the optimal translation from $\mathcal{X}$ to $\mathcal{Y}$ becomes the following optimization problem:

$$\mathbf{t}^* = \underset{\mathbf{t} \in R^3}{\arg\max} \quad E(\mathcal{Q}, F(\mathcal{P}, \mathbf{t})) \tag{5}$$

where $E(\mathbf{t})$ is a function that measures the alignment between two RIF sets. By solving the optimization problem in (5) over R^3, we can find the optimal $\mathbf{t}^*$, which forms part of the solution to the optimization problem in (1) over $SE(3)$.

To make the objective function robust to outliers, we define $E(\mathbf{t})$ on the basis of a consensus set as follows:

$$E(\mathcal{Q}, F(\mathcal{P}, \mathbf{t})) = \sum_m \max_n \lfloor \|F(\mathbf{p}_m, \mathbf{t}) - \mathbf{q}_n\|_\infty \leq \varepsilon \rfloor \tag{6}$$

where $\lfloor \cdot \rfloor$ is an indicator function that returns 1 if the condition $\cdot$ is true and 0 otherwise. $\| \cdot \|_\infty$ is the L-infinity norm in R^3 and ε is the inlier threshold. The size of both $\mathcal{Q}$ and $\mathcal{P}$ is very large. From the definition of a RIF, we can see that the third element of a RIF, which is the distance between the two points in the pair, is invariant with respect to translation. This means that the third elements of the corresponding RIFs in and $\mathcal{Q}$ and $\mathcal{P}$ are equal to each other. In practice, we do not need to match the entirety of $\mathcal{Q}$ and $\mathcal{P}$. Instead, we match a subset of the RIFs in $\mathcal{Q}$ and $\mathcal{P}$, the third elements of which fall within a specific range. We denote these two subsets by $\mathcal{Q}' = \{\mathbf{q}_n\}_{n=1}^{Q'}$ and $\mathbf{P}' = \{\mathbf{p}_m\}_{m=1}^{P'}$, and the actual objective function used in this paper is

$$E(\mathcal{Q}', F(\mathcal{P}', \mathbf{t})) = \sum_m \max_n \lfloor \|F(\mathbf{p}_m, \mathbf{t}) - \mathbf{q}_n\|_\infty \leq \varepsilon \rfloor \tag{7}$$

3.3   Bounds and Branch-and-Bound-Based Algorithm

To maximize (7) via BnB optimization, we need an upper bound on this objective function in a branch of the parameter space. In our BnB algorithm, we search for the optimal translation in a cube in R^3 and iteratively divide the parameter space into sub-cubes. As shown in Fig. 2, for a translation cube $\mathbb{T}$ with a diagonal length of $2r$ and centred at $\mathbf{t}_c$, we have

$$\|\mathbf{x} + \mathbf{t}_c\| - r \leq \|\mathbf{x} + \mathbf{t}\| \leq \|\mathbf{x} + \mathbf{t}_c\| + r \tag{8}$$

(a) (b)
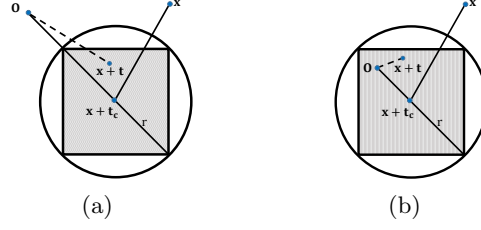
Fig. 2. Bounds on the distance from the origin to a point translated by a translation cube. Two-dimensional coordinates are used for clearer illustration. The translation cube is $\mathbb{T}$, and the possible positions of point $\mathbf{x}$ after being translated by an arbitrary $\mathbf{t} \in \mathbb{T}$ are represented by the grey square. The diagonal length of $\mathbb{T}$ is $2r$, and its centre is at $\mathbf{t}_c$. Therefore, the distance from the origin to an arbitrary point in the grey square is between $\|\mathbf{x} + \mathbf{t}_c\| - r$ and $\|\mathbf{x} + \mathbf{t}_c\| + r$. (a) and (b) illustrate the two cases in which the origin is located outside and inside, respectively, of the circle centred at $\mathbf{t}_c$ with a radius of r.

Let $\mathbf{p}_m$ be the RIF constructed from a point pair $\{\mathbf{x}_{m1}, \mathbf{x}_{m2}\}$ from the moving point cloud and let $\mathbf{q}_n$ be the RIF constructed from a point pair $\{\mathbf{y}_{n1}, \mathbf{y}_{n2}\}$ from the reference point cloud. Then, for any $\mathbf{t} \in \mathbb{T}$, using formula (8) , we have

$$\|F(\mathbf{p}_m, \mathbf{t}) - \mathbf{q}_n\|_\infty = \left\| \left\{ \begin{array}{c} \|\mathbf{x}_{m1} + \mathbf{t}\| - \|\mathbf{y}_{n1}\| \\ \|\mathbf{x}_{m2} + \mathbf{t}\| - \|\mathbf{y}_{n2}\| \\ \|\mathbf{x}_{m1} - \mathbf{x}_{m2}\| - \|\mathbf{y}_{n1} - \mathbf{y}_{n2}\| \end{array} \right\} \right\|_\infty := C(\mathbf{t}) \quad (9)$$

where $\|\mathbf{x}_{m1} - \mathbf{x}_{m2}\| - \|\mathbf{y}_{n1} - \mathbf{y}_{n2}\|$ is constant with respect to $\mathbf{t}$. Because we can compute the bounds on $\|\mathbf{x}_{m1} + \mathbf{t}\|$ and $\|\mathbf{x}_{m2} + \mathbf{t}\|$ from equation (8), we can easily compute the bounds on the absolute values of $\|\mathbf{x}_{m1} + \mathbf{t}\| - \|\mathbf{y}_{n1}\|$ and $\|\mathbf{x}_{m2} + \mathbf{t}\| - \|\mathbf{y}_{n2}\|$. Then we can compute the upper and lower bounds on $\|F(\mathbf{p}_m, \mathbf{t}) - \mathbf{q}_n\|_\infty$, and we denote this lower bound by $\underline{C}(\mathbf{t})$. Thus, we have

$$\|F(\mathbf{p}_m, \mathbf{t}) - \mathbf{q}_n\|_\infty \geq \underline{C}(\mathbf{t}) \quad (10)$$

It follows that

$$\lfloor \|F(\mathbf{p}_m, \mathbf{t}) - \mathbf{q}_n\|_\infty \leq \varepsilon \rfloor \leq \lfloor \underline{C}(\mathbf{t}) \leq \varepsilon \rfloor \quad (11)$$

Then, we can define the upper-bound function as

$$\overline{E}(\mathbb{T}) = \sum_m \max_n \lfloor \underline{C}(\mathbf{t}) \leq \varepsilon \rfloor \quad (12)$$

For any $\mathbf{t} \in \mathbb{T}$, we have

$$E(\mathcal{Q}', F(\mathcal{P}', \mathbf{t})) \leq \overline{E}(\mathbb{T}) \quad (13)$$

Utilizing the upper bound given by formula (12), we can search the translation space to find the globally optimal translation that maximizes the objective function (7). The algorithm is outlined in **Algorithm 1**.

Algorithm 1: Globally Optimal Translation Search Based on RIFs

---

Input: Point sets $\mathcal{X}$ and $\mathcal{Y}$
Output: Optimal translation $\mathbf{t}^*$ between $\mathcal{X}$ and $\mathcal{Y}$.

1 Construct two sets of RIF triples, $\mathcal{P}'$ and $\mathcal{Q}'$.
2 Initialize translation branch $\mathbb{T}_0$;$E^* \leftarrow 0$,$\mathbf{t}^* \leftarrow \mathbf{0}$.Insert $\mathbb{T}_0$ into queue q with
   priority $\overline{E}(\mathbb{T}_0)$.
3 while q *is not empty* **do**
4   Obtain the highest-priority cube $\mathbb{T}$ from q.
5   **if** $\overline{E}(\mathbb{T}) = E^*$ **then**
6 | Terminate.
7   **end**
8 $\mathbf{t}_c \leftarrow$ centre of cube $\mathbb{T}$
9   **if** $E(\mathbf{t}_c) > E^*$ **then**
10 | Update $E^* \longleftarrow E(\mathbf{t}_c), \mathbf{t}^* \leftarrow \mathbf{t}_c$
11   **end**
12   Subdivide $\mathbb{T}$ into eight cubes $\{\mathbb{T}_d\}_{d=1}^8$.
13   **for** *each* $\mathbb{T}_d$ **do**
14   **if** $\overline{E}(\mathbb{T}_d) > E^*$ **then**
15 | Insert $\mathbb{T}_d$ into q with priority $\overline{E}(\mathbb{T}_d)$.
16   **end**
17   **end**
18 end
19 Obtain the optimal translation $\mathbf{t}^*$.

3.4   Six-Dimensional Registration

Once the globally optimal translation $\mathbf{t}^*$ between $\mathcal{X}$ and $\mathcal{Y}$ has been found, it is easy to calculate the optimal rotation $\mathbf{R}^*$ between the two point clouds. For example, we could use the global rotation search methods presented in [15] or [16]; however, the optimal rotation can actually be obtained in a much easier way. When we obtain the optimal $\mathbf{t}^*$ via **Algorithm 1**, we also obtain a set of consensus RIFs between $\mathcal{P}'$ and $\mathcal{Q}'$, which provide us with potential correspondences between points in the original point clouds $\mathcal{X}$ and $\mathcal{Y}$. Of course, there will be some outliers among these correspondences, and the problem of estimating $\mathbf{R}^*$ becomes a robust estimation problem, which can be solved quickly and reliably. In this paper, we estimate $\mathbf{R}^*$ by means of the RANdom Sample Consensus (RANSAC) algorithm[19]. We refer to our 6D rigid registration method as GoTS. Many global optimization methods sacrifice accuracy for speed and global optimality, and a subsequent local refinement is performed to improve accuracy. Similarly, a local refinement procedure can be added to GoTS; we refer to the algorithm with local refinement as GoTS$^+$.

4   Experiments and Results

In this section, we report the results of evaluating the performance of GoTS and GoTS$^+$ using many different point clouds and compare the proposed al-

gorithms against the following three global methods: Go-ICP[13], Glob-GM[15] and APM[18]. We did not perform comparisons with the methods of [6] and [17], which use GPUs. GoTS and GoTS$^+$ were implemented in MATLAB. The code for the other methods was obtained from the authors. In GoTS, a range is set for the third elements of the RIFs that are used to construct $\mathcal{P}'$ and $\mathcal{Q}'$ for the translation search. We set the lower bound of this range to 65% of the shortest extension of the original point cloud in the X, Y and Z directions, and set the range width to 0.01 times inlier threshold. In the supplementary material, we studied the registration time with respect to the range width used in choosing the RIFs for translation search. The original point clouds used in these experiments are very dense, and we down-sampled them using the *pcdownsample* function of MATLAB with a specified *gridsize*. In the experiments with synthetic data in section 4.1 and 4.2, we scaled the points to make them fall within a cube of $[-1,1]^3$ and set 0.01 as the inlier threshold for GoTS; in the experiment with real data in section 4.3 and 4.4, we set *gridsize* as the inlier threshold for GoTS. In Glob-GM, the matchlist and local-refinement options were turned on. In Go-ICP, a distance transformation was used to speed up the nearest-neighbour distance calculation. All experiments were performed on a computer with a 2.80 GHz Intel(R) Core(TM) i7-7700HQ CPU and 16 GB of RAM. The run time for each method includes the method-specific pre-processing time.

4.1   Run-Time Comparison with Other Global Methods

We first compared the run times of GoTS for registering point clouds with different numbers of points to the run times of the other three methods. In this experiment, we used the *bunny* point cloud from the Stanford 3D Scanning Repository and down-sampled the original data to point clouds with different numbers of points. The down-sampled point cloud was scaled to fall within a cube of $[-1,1]^3$ and was regarded as the moving point cloud, and it was transformed to a new position via a random transformation to form the reference point cloud. Therefore, we knew the ground-truth transformation from the moving point cloud to the reference point cloud and the point correspondence between the two point clouds. The moving point cloud was then registered to the reference point cloud with each of the four methods. For each number of points, we performed 20 registrations with different random ground-truth transformations, and the average run times with respect to the number of points are shown in Fig. 3(a) and Fig. 3(b). The run time of Glob-GM and APM exceeded 1000 seconds when registering 400 points. To better illustrate the difference between GoTS and Go-ICP, we show the run times of only these two methods in Fig. 3(b). We can see that GoTS and Go-ICP are much faster than the other two methods and that GoTS is approximately four times as fast as Go-ICP. In this experiment, all methods could successfully register the moving and reference point clouds.

Figure 3 (c) shows the evolution of the upper and lower bounds of **Algorithm 1** during one registration between 1000 moving points and 1000 reference points, in which it took only 5.2 seconds for GoTS to converge.
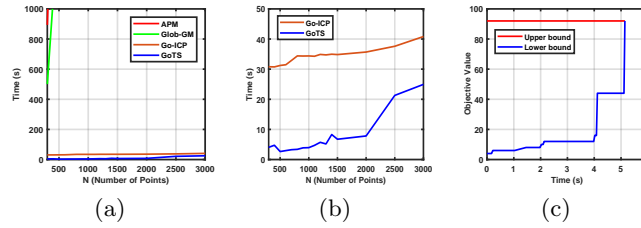
Fig. 3. (a) Run times of GoTS, Go-ICP, APM and Glob-GM with respect to the number of points. (b) Run times of only GoTS and Go-ICP with respect to the number of points, to better illustrate the run time difference between these two methods. (c) Evolution of the upper and lower bounds in the global optimal translation search method (**Algorithm 1**) as a function of time when registering 1000 points.

4.2   Robustness to Outliers

In this experiment, we evaluated the robustness of GoTS to outliers and compared it to the robustness of the other three methods. Outliers are commonly encountered in point cloud registration; in this section, we study only the influence of gross outliers, which do not belong to the true object represented by the point cloud. Gross outliers may originate from imperfections of the device used to obtain the point cloud. In the registration of partially overlapping surfaces, the points in the region that is not overlapped by the other point cloud can also be regarded as outliers; experiments focusing on this scenario are reported in the next section.

We down-sampled the *bunny* data from the Stanford 3D Scanning Repository (http://graphics.stanford.edu/data/3Dscanrep/) with a *gridsize* of 0.0121 to obtain 500 points and then scaled these points to fall within a cube of $[-1,1]^3$. The scaled data points were used as the clean moving point cloud. Uniformly distributed random outlier points were added to this clean point cloud, and the resulting point cloud was then transformed via a random transformation to form a reference point cloud. Different numbers of outliers were added to generate reference point clouds with five different outlier percentages: 10%, 20%, 30%, 40% and 50%. Figure 4(c) and (d) show examples of reference point clouds with 10% and 50% outliers, respectively. We ran each algorithm 20 times for each outlier ratio.

Table 1 reports the root mean square errors on the inlier points for each method. Both GoTS and Go-ICP achieved high accuracy, indicating that they are both very robust to outliers. Although the errors of Go-ICP were smaller than those of GoTS, the accuracy of GoTS is sufficiently high in practice. The errors of GoTS$^+$ show that two point clouds can be perfectly aligned by means of a subsequent local refinement, which requires less than one second of additional time. The registration error of Glob-GM is much higher than that of GoTS, and it increases as the outlier ratio increases. Considering that all the points lie in a cube with an edge length of 2, the registration errors of Glob-GM indicate

that it fails in aligning the two point clouds in many cases. The errors of APM are even higher; the reason for these large errors may be that the assumption of a one-to-one correspondence that is adopted in APM is not valid in these experiments with outliers.

Table 1. Root mean square error between corresponding inlier points after registration.

Method	GoTS	GoTS$^+$	Go-ICP	Glob-GM	APM
10%	1.03E-02	1.38E-15	8.95E-06	1.10E-01	2.17E-01
20%	1.20E-02	9.72E-16	7.51E-06	1.21E-01	-
30%	1.14E-02	9.50E-15	4.65E-06	1.14E-01	-
40%	1.09E-02	1.38E-15	4.12E-06	1.12E-01	-
50%	1.17E-02	1.19E-15	6.22E-06	1.41E-01	-

The average run times of the four algorithms with respect to the outlier ratio are shown in Fig. 4(a). The run time of APM increased rapidly as we added more outliers to the 500 inlier points and exceeded 1000 seconds when the outlier ratio was 20%. The run time of Glob-GM decreased when the outlier ratio was above 20%; this occurred because the algorithm terminated earlier at an incorrect solution in which the two point clouds were not aligned. Again, we plot the average run times of GoTS and Go-ICP separately in Fig. 4(b) to better illustrate the difference between them.
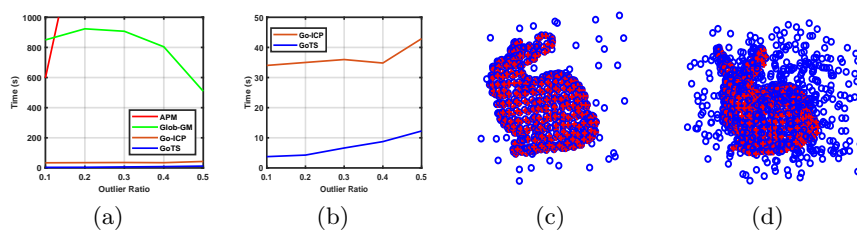
Fig. 4. (a) Run times of the four methods with respect to the outlier ratio. (b) Run times of only GoTS and Go-ICP with respect to the outlier ratio, to better illustrate the difference between them. (c), (d) Reference point clouds with 10% outliers and 50% outliers, respectively.

4.3   Partially Overlapping Registration of Real 3D Scans

In this section, we report the results of testing the performance of the proposed method on partially overlapping point clouds using real data. We compared GoTS and GoTS$^+$ to Go-ICP and Glob-GM in this experiment. APM was not used here, but we can expect that its registration error would be large in this experiment because of its difficulty in dealing with outliers.

We used bun000, bun045, bun090, ArmadilloStand60, ArmadilloStand30 and ArmadilloStand0 from the Stanford 3D Scanning Repository in this experiment. The first three point clouds are scans of *bunny* from three different directions, and the last three are scans of *Armadillo* from three different directions. The original point clouds were down-sampled to clouds consisting of 1000 to 2000 points, all of which lay in a cube with an edge length of 0.3. The ground-truth transformations between each pair of scans of the same object are provided in the data set. Using the four methods, we performed registrations between four pairs of point clouds: ArmadilloStand60/ArmadilloStand30, ArmadilloStand30/ArmadilloStand0, bun045/bun000 and bun090/bun000. The point cloud pairs before and after registration are shown in the supplementary material.

Table 2. Results of registering *Armadillo* scans using four different methods.

	AmadilloStandd60/30				ArmadilloStand30/0			
Method	GoTS	GoTS$^+$	Go-ICP	Glob-GM	GoTS	GoTS$^+$	Go-ICP	Glob-GM
Tran[m]	0.0037	0.0039	**0.0023**	0.28	0.0065	0.0063	**0.0005**	0.014
Rot[°]	0.94	**0.84**	4.13	172	1.82	**1.70**	1.73	38.10
Time[s]	**27.11**	27.33	30.67	447	**17.00**	17.22	32.08	3136

Table 3. Results of registering *bunny* scans using four different methods.

	bun045/000				bun090/000			
Method	GoTS	GoTS$^+$	Go-ICP	Glob-GM	GoTS	GoTS$^+$	Go-ICP	Glob-GM
Tran[m]	0.0058	0.0057	**0.0024**	0.025	**0.0069**	0.0074	0.10	0.14
Rot[°]	3.95	**3.94**	4.10	19.41	3.82	**3.55**	90.24	137.02
Time[s]	**14.02**	14.25	30.44	2361	**56.39**	56.62	603.63	868

The translation and rotation errors together with the run time of each method are listed in Table 2 and Table 3. In these experiments, the inlier threshold of Glob-GM was set to 0.05 for the *Armadillo* data and to 0.029 for the *bunny* data because Glob-GM could not terminate within 1000 seconds when we used an inlier threshold equal to the *gridsize* of the down-sampling function. We can see that GoTS achieved high accuracy in these challenging registrations. In particular, the translation errors achieved by the global optimal translation search method in **Algorithm 1** are all very small. Go-ICP failed on bun090/000, which is a difficult case because the relative transformation between the two point clouds is very large and the overlap ratio is small. The run time of Go-ICP was extremely long in this case, while in the other three cases, the run time of Go-ICP was slightly longer than or twice as long as that of GoTS. Carefully tuning the trimming ratio may improve the accuracy of Go-ICP, but in practice, the best trimming ratio cannot be known before registration. Glob-GM failed in all four cases, and its run times were very long. The number of RIFs generated for the global translation search was between 150 and 160 in these experiments.

4.4 Registration of Scans of *LivingRoom*

We tested the performance of the proposed method on large-scale field data using "livingRoom.mat" from the MATLAB example "3-D Point Cloud Registration and Stitching", which consists of a series of 3D point sets obtained by continuously scanning a living room. We used GoTS, Go-ICP and Glob-GM to register the point clouds of the first and sixth frames of these data. The original point clouds consist of more than 300,000 points, and all these points lie in a cube with an edge length of 4 metres. They were down-sampled with a *gridsize* of 0.1 metre.

(a) Init   (b) GoTS   (c) Go-ICP   (d) Glob-GM
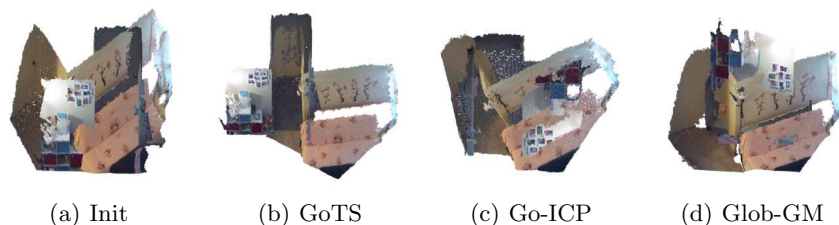
Fig. 5. *LivingRoom* point clouds before and after registration.

The results are listed in Table 4. GoTS successfully registered the two point clouds in 25.90 seconds. In contrast, Go-ICP failed to register the point clouds when the mean square error threshold was set to 0.05. When we used a mean square error threshold of 0.01, the algorithm could not terminate within half an hour. For Glob-GM, when we set the inlier threshold to 0.1, the algorithm could not terminate within half an hour. Then, we set the inlier threshold to 0.8;

the algorithm terminated after 633.8 seconds, but the result was incorrect. The numbers of RIFs generated from the reference and moving point clouds were 1434 and 1866, respectively. More results can be found in the supplementary material.

Table 4. Results of registering *livingRoom* point clouds.

Method	GoTS	Go-ICP	Glob-GM
Tran[m]	**0.1047**	1.158	0.4555
Rot[°]	**4.382**	158.3	97.59
Time[s]	**25.90**	489.6	633.8

5   Conclusion

We introduce a fast global 3D rigid point cloud registration method based on the decoupling of translation and rotation optimization via a newly proposed rotation invariant feature. We first globally optimize the translation by using a BnB algorithm to match sets of rotation invariant features constructed from the two point clouds, and we then calculate the optimal rotation using the optimized translation. Decoupling the optimization of the translation and rotation makes the proposed algorithm more efficient than existing methods. Experiments on real data demonstrate the superiority of the proposed algorithm in cases of partial overlap, large angular differences and high outlier ratios. All code will be available on the web (http://www.fudanmiccai.org).

Acknowledgements

References

1. T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH, vol. 34, pp. 598-626, 2015.

2. P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," IN-TERNATIONAL JOURNAL OF ROBOTICS RESEARCH, vol. 31, pp. 647-663, 2012.

3. R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking," in International Symposium on Mixed and Augmented Reality, 2011, pp. 127-136.

4. J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image regis-tration methods with accuracy evaluation," IMAGE AND VISION COMPUTING, vol. 25, pp. 578-596, 2007.

5. X. Xu and M. Wang, "A Surface-Based Spatial Registration Method Based on Sense Three-Dimensional Scanner (vol 28, pg 157, 2017)," JOURNAL OF CRAN-IOFACIAL SURGERY, vol. 28, p. 1581-1581, 2017.

6. J. Straub, T. Campbell, J. P. How, and J. W. I. Fisher, "Efficient Global Point Cloud Alignment using Bayesian Nonparametric Mixtures," in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2403-2412.

7. P. J. BESL and N. D. MCKAY, "A METHOD FOR REGISTRATION OF 3-D SHAPES," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 14, pp. 239-256, 1992.

8. D. Chetverikov, D. Stepanov and P. Krsek, "Robust euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm," IMAGE AND VISION COMPUTING, vol. 23, pp. 299-309, 2005.

9. B. Jian and B. C. Vemuri, "Robust Point Set Registration Using Gaussian Mixture Models," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 33, pp. 1633-1645, 2011.

10. S. KIRKPATRICK, C. D. GELATT and M. P. VECCHI, "OPTIMIZATION BY SIMULATED ANNEALING," SCIENCE, vol. 220, pp. 671-680, 1983.

11. M. P. Wachowiak, R. Smolikova, Y. F. Zheng, J. M. Zurada, and A. S. El-maghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimization," IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, vol. 8, pp. 289-301, 2004.

12. H. Li and R. Hartley, "The 3D-3D registration problem revisited," in IEEE Inter-national Conference on Computer Vision, 2007, pp. 1947-1954.

13. J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A Globally Optimal Solu-tion to 3D ICP Point-Set Registration," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 38, pp. 2241-2254, 2016.

14. R. I. Hartley and F. Kahl, "Global Optimization through Rotation Space Search," INTERNATIONAL JOURNAL OF COMPUTER VISION, vol. 82, pp. 64-79, 2009.

15. A. P. Bustos, T. Chin, A. Eriksson, H. Li, and D. Suter, "Fast Rotation Search with Stereographic Projections for 3D Registration," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 38, pp. 2227-2240, 2016.

16. D. Campbell, L. Petersson, L. Kneip, and H. Li, "Globally-Optimal Inlier Set Maximisation for Simultaneous Camera Pose and Feature Correspondence," in IEEE International Conference on Computer Vision, 2017, pp. 1-10.

17. D. Campbell and L. Petersson, "GOGMA: Globally-Optimal Gaussian Mixture Alignment," in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5685-5694.

18. W. Lian, L. Zhang and M. Yang, "An Efficient Globally Optimal Algorithm for Asymmetric Point Matching.," IEEE transactions on pattern analysis and machine intelligence, 2016.
19. M. A. FISCHLER and R. C. BOLLES, "RANDOM SAMPLE CONSENSUS - A PARADIGM FOR MODEL-FITTING WITH APPLICATIONS TO IMAGE-ANALYSIS AND AUTOMATED CARTOGRAPHY," COMMUNICATIONS OF THE ACM, vol. 24, pp. 381-395, 1981.