

# 2D-3D Point Set Registration Based on Global Rotation Search

Yinlong Liu, Yuan Dong<sup>ID</sup>, Zhijian Song, and Manning Wang

**Abstract**—Simultaneously determining the relative pose and correspondence between a set of 3D points and its 2D projection is a fundamental problem in computer vision, and the problem becomes more difficult when the point sets are contaminated by noise and outliers. Traditionally, this problem is solved by local optimization methods, which usually start from an initial guess of the pose and alternately optimize the pose and the correspondence. In this paper, we formulate the problem as optimizing the pose of the 3D points in the SE(3) space to make its 2D projection best align with the 2D point set, which is measured by the cardinality of the inlier set on the 2D projection plane. We propose four geometric bounds for the position of the projection of a 3D point on the 2D projection plane and solve the 2D-3D point set registration problem by combining a global optimal rotation search and a grid search of translation. Compared with existing global optimization approaches, the proposed method utilizes a different problem formulation and more efficiently searches the translation space, which improves the registration speed. Experiments with synthetic and real data showed that the proposed approach significantly outperformed the state-of-the-art local and global methods.

**Index Terms**—Rigid registration, 2D-3D registration, pose estimation, correspondence determination, global rotation search.

## I. INTRODUCTION

ESTIMATING the relative pose between a 3D object and its 2D projection is a fundamental problem in computer vision [1]–[4] and medical imaging [5]–[8]. Figure 1 shows that this problem can be formulated as a 2D-3D point set registration problem if the projection parameters are known and fixed.

Manuscript received December 12, 2017; revised May 7, 2018, July 31, 2018, and September 22, 2018; accepted October 17, 2018. Date of publication December 17, 2018; date of current version March 6, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 81471758 and Grant 81701795, in part by the National Key Research and Development Program of China under Grant 2017YFC0110700, and in part by the Program of Shanghai Academic/Technology Research Leaders under Grant 16XD1424900. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Guoliang Fan. (*Yinlong Liu and Yuan Dong contributed equally to this work.*) (*Corresponding authors:* Zhijian Song; Manning Wang.)

Y. Liu was with the Digital Medical Research Center, School of Basic Medical Science, Fudan University, Shanghai 200032, China. He is now with the Department of Informatics, Technical University of Munich, 80333 Munich, Germany (e-mail: yinlongliu15@fudan.edu.cn).

Y. Dong, Z. Song, and M. Wang are with the Digital Medical Research Center, School of Basic Medical Science, Fudan University, Shanghai 200032, China, and also with the Shanghai Key Laboratory of Medical Image Computing and Computer Assisted Intervention, Shanghai 200032, China (e-mail: yuandong15@fudan.edu.cn; zjsong@fudan.edu.cn; mnwang@fudan.edu.cn).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. The material includes three parts: (1) Proof of some lemmas and theories; (2) Illustration of non-elliptical projection of an uncertainty ball; (3) Difference of two different formulations of 2D-3D point set registration problem. Size is 0.767 MB.

Digital Object Identifier 10.1109/TIP.2018.2887207

Many successful solutions have been proposed for known correspondences between the 2D and the 3D points to be registered. If the correspondences are perfect, then the problem becomes the well-studied classic PnP problem. Many methods are available to accurately estimate the pose from noisy data [9]–[11]. If outliers exist in the correspondences, then the problem becomes a robust estimation problem, which is also a popular topic in the computer vision field. Many approaches have also been proposed to alleviate the influence of outliers, such as the classic (RANSAC) method [12] and its variants [35], robust estimators [13]–[15] or outlier removal techniques [16], [17]. Nevertheless, all these methods require some correspondences; however, establishing correspondence between 2D and 3D points is not a trivial task. Although feature extracting and matching between a pair of 3D point clouds [36] or between a pair of 2D images [31] are fairly mature techniques, matching 2D and 3D features is difficult because of the loss of 3D geometry in a 2D projection, perspective distortion, occlusions, etc. [8]. For example, features on 3D models usually explore the 3D geometry of the object, which is lost in its 2D projection, whereas features on 2D images usually utilize image textures, which are either not available in the 3D model or are severely distorted in the 2D projection.

When no correspondence is available, 2D-3D point set registration becomes the more challenging *Simultaneous Pose and Correspondence Determination* (SPCD) [4], [18] problem. SPCD solves the pose and the correspondence problem simultaneously without additional information except for the points' location. The difficulty of SPCD comes from the fact that the two subproblems are mutually interlocked, which means solving one is the precondition for solving the other [18].

In this paper, we introduce a new algorithm to solve the 2D-3D point set registration problem without point correspondences, and it is based on a novel global optimal rotation search algorithm using the branch and bound (BnB) optimization scheme. Based on the rotation search, we propose a synchronized grid-search method to efficiently find the optimal translation.

## II. LITERATURE REVIEW

The most popular methods for solving the SPCD problem are EM-type algorithms [19], [20] which start from an initial pose and solve the two interlocked sub-problems alternately until convergence to a local optimum. Essentially, these algorithms represent a type of local optimization method, and a good pose prior is needed for them to converge to the global optimum. ICP is a typical EM-type algorithm [20], and the

original ICP algorithm only allows for binary correspondence, which makes the objective function have many local minima and the convergence basin is small. Many efforts have been made to achieve a large basin of convergence by allowing some type of soft correspondence [4], [21], and one of the best approaches is the SoftPOSIT technique [4], in which multiple weighted correspondences are used. More recently, researchers have modeled the point set with Gaussian mixture models (GMM) and transformed the point set registration problem into probability distribution alignment [6], [22]. When the variance of the Gaussian kernels is increased, the objective function will become smoother and there will be fewer local optima, which facilitates reaching the global optimum from an approximate pose prior. Despite these efforts, the results of these methods heavily depend on a good initial pose, which is difficult to find in the SE(3) space.

To relieve the dependence on a good pose prior, a direct approach is to run a local method multiple times at different positions in the parameter space and choose the best result. However, brute-force multi-start running is computationally impractical, especially when the speed of the local method itself decreases rapidly as the number of points increases. Another line of research is to utilize heuristic strategies to search the parameter space in a more intelligent manner, such as differential evolution algorithms [23] and genetic algorithms [24]. Although these methods can improve the convergence rate, the convergence rate is still not high, and these methods tend to be slow.

A convex programming framework for 2D-3D point set registration was proposed to obtain global optimality without a pose prior in [5]. Concretely, this framework formulates the 2D-3D registration as a mixed-integer nonlinear program problem and relaxes it to a convex semidefinite program that can be solved globally by the interior-point method. However, only the rotation of the 3D points is recovered, and no solution is provided to address the translation. In addition, this method only obtains exact recovery guarantees when there is no noise in the points' position, which is an unrealistic assumption, especially when the points are only samples on corresponding lines or curves.

Recently, there has been a surge of interest in using the BnB optimization framework to solve the 3D-3D point set registration problem globally [18], [25]–[28], which guarantees global optimality by searching the entire SE(3) space. The key in using the BnB framework to solve an optimization problem is to find upper and/or lower bounds of the objective function, and almost every new algorithm involves the derivation of a new bound. This framework was also used to globally register 2D and 3D points or lines in [2], where the BnB was applied to minimize the sum of the angular distance between 2D and 3D points. This method used a trimmed angular distance as the objective function, which requires the user to specify the inlier fraction. Campbell *et al.* [1] pointed out that the global optimum of a trimmed objective function may not occur at the true pose, particularly when an incorrect trimming fraction is used. Moreover, the inlier fraction can rarely be known exactly, and choosing a proper one is not intuitive in certain applications. More recently, another BnB-based global optimal

method was proposed to solve the SPCD problem [1], and it maximizes the cardinality of the inlier set and avoids the problems associated with trimming.

Both [1] and [2] formulate the 2D-3D point set registration as a camera pose estimation problem, in which the 3D points are fixed and the optimal camera orientation (in SO(3)) and camera center (in  $\mathbb{R}^3$ ) are searched in SE(3) so that the image of the 3D points captured by the camera matches the 2D point set. One disadvantage of this formulation is that if we want to cover the whole relative angle between the 3D points and the camera, we need to move the camera center all around the 3D points, and the range of the translation that needs to be searched is very large. In this paper, we utilize another formulation of the 2D-3D point set registration problem, in which the 2D points and the projection coordinate system are fixed, and we search the rotation and the translation of the 3D points in the projection coordinate system to generate a 2D projection that best aligns with the 2D point set. The difference between our formulation and that in [1] and [2] is further discussed in the Supplement C. The cardinality of the inlier set on the 2D projection plane is used as the objective function, which is robust to outliers. We maximize the objective function using the BnB framework with newly proposed bounds, and a global optimal rotation search was efficiently achieved in SO(3). Furthermore, based on the fast global optimal rotation search and the new problem formulation, we provide an efficient algorithm for searching the SE(3) space.

*Contribution:* The main contribution of this paper can be summarized as follows.

1) *New bounds for a global rotation search in 2D-3D registration:* We formulate 2D-3D point set registration as the maximization of the cardinality of the inlier set on the 2D projection plane. Based on this formulation, we first derive four upper bounds of the objective function and propose an efficient global optimal rotation search algorithm in the parameter space of SO(3) using the BnB framework.

2) *An efficient algorithm to search in SE(3):* Compared with [1] and [2], our problem formulation does not consider the interaction between rotation and translation, thus gaining efficiency at the expense of guaranteed global optimality. We show that under our new problem formulation, the proposed rotation search algorithm can cover a fairly large region in the translation space, and based on this finding, we developed a multi-start grid-search algorithm to search the translation space. We develop a synchronizing strategy to improve the efficiency of the grid search in the translation space.

### III. METHOD

#### A. Problem Formulation

Let a set of 3D points  $X = \{\mathbf{x}_i\}_{i=1}^M$  represent a 3D model, where  $\mathbf{x}_i \in \mathbb{R}^3$  is defined in the 3D model coordinate system and a set of 2D points  $Y = \{\mathbf{y}_j\}_{j=1}^N$ , where  $\mathbf{y}_i \in \mathbb{R}^2$  represents the projection of the 3D model on a 2D projection plane.  $M$  may be different from  $N$  because of the occlusion of the 3D model or the outliers on the projection plane. The projection operator, which projects a 3D point from the projection coordinate system onto the 2D plane, is assumed to

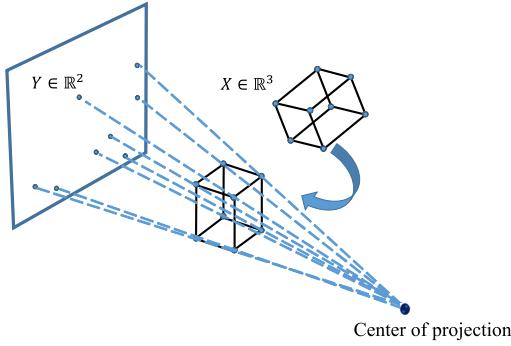


Fig. 1. Illustration of the 2D-3D point set registration problem. Assuming the projection parameters are known and fixed, the problem is to find a rotation and a translation of the 3D point set  $X$  to ensure that its projection on the projection plane matches the 2D point set  $Y$ .

be known and fixed, and we denote it as  $P(\cdot)$ . The 2D-3D point set registration problem is to find a transformation in  $\text{SE}(3)$  to transform  $X$  from the model coordinate system to the projection coordinate system so that its 2D projection aligns with  $Y$ . Without loss of generality, we make the model coordinate system the same as the projection coordinate system, and then the problem becomes rotating and translating  $X$  (in the projection coordinate system) to a proper pose so that its 2D projection aligns with  $Y$ , as illustrated in Fig. 1.

In this paper, we use the cardinality of the inlier set on the 2D projection plane as the criterion of alignment; therefore, we maximize the following function over  $\mathbf{R} \in \text{SO}(3)$  and  $\mathbf{t} \in \mathbb{R}^3$ .

$$E(\mathbf{R}, \mathbf{t}) = \sum_j \max_i \lfloor \|P(\mathbf{R}(\mathbf{x}_i - \mathbf{x}_o) + \mathbf{x}_o + \mathbf{t}) - \mathbf{y}_j\| \leq \epsilon \rfloor \quad (1)$$

where  $\mathbf{x}_o \in \mathbb{R}^3$  is the rotation center,  $\|\cdot\|$  is the Euclidean norm, and  $\lfloor \cdot \rfloor$  is a binary function, which is 1 if the inner condition is true and 0 otherwise. In (1),  $\epsilon$  is the inlier threshold, and a pair of points  $\mathbf{x}_i$  and  $\mathbf{y}_j$  are considered an inlier pair only when their distance on the 2D projection plane is equal to or less than  $\epsilon$ . Function (1) is robust to outliers since it does not count the points out of a distance threshold. More critically, this objective function avoids the problem associated with trimming, which is influenced by the inlier fraction.

### B. Rotation Search With Branch and Bound

The BnB framework has been widely used in searching  $\text{SE}(3)$  for both 3D-3D and 2D-3D point set registration. Typically, searching the rotation parameters in  $\text{SO}(3)$  is more difficult than searching the translation parameters in  $\mathbb{R}^3$  due to the special structure of  $\text{SO}(3)$  [25]. Therefore, most of the existing methods solve the rotation search problem first, which is by no means a trivial problem, and then extend it to search in  $\text{SE}(3)$ . In this paper, we also follow this technological process to search the rotation space first.

When only the pure rotation search problem is considered, (1) can be simplified as follows:

$$Q(\mathbf{R}) = \sum_j \max_i \lfloor \|P(\mathbf{R}(\mathbf{x}_i - \mathbf{x}_o) + \mathbf{x}_o) - \mathbf{y}_j\| \leq \epsilon \rfloor \quad (2)$$

---

**Algorithm 1** Global Optimal Pose Optimization in  $\text{SO}(3)$ 


---

**Input:** 3D point set  $X$ , 2D point set  $Y$ , and threshold  $\epsilon$ .  
**Output:** Optimal rotation  $\mathbf{R}^*$  with quality  $Q^*$ .

```

1: Initialize priority queue  $q$ ,  $\mathbb{B} \leftarrow$  cube of size  $2\pi$ ,
    $Q^* \leftarrow 0$ ,  $\mathbf{R}^* \leftarrow \mathbf{I}$ .
2: Insert  $\mathbb{B}$  with priority  $\overline{Q}(\mathbb{B})$  into  $q$ .
3: While  $q$  is not empty do:
4:   Obtain highest priority cube  $\mathbb{B}$  from  $q$ .
5:   If  $\overline{Q}(\mathbb{B}) = Q^*$  Then
6:     Terminate.
7:   End If
8:    $\mathbf{R}_c \leftarrow$  Center rotation of  $\mathbb{B}$ .
9:   If  $Q(\mathbf{R}_c) > Q^*$  Then
10:    Update  $Q^* \leftarrow Q(\mathbf{R}_c)$ ,  $\mathbf{R}^* \leftarrow \mathbf{R}_c$ .
11:   End If
12:   Subdivide  $\mathbb{B}$  into eight cubes  $\{\mathbb{B}_d\}_{d=1}^8$ .
13:   For each  $\mathbb{B}_d$ ,
14:     If  $\overline{Q}(\mathbb{B}_d) \geq Q^*$ , Then
15:       Insert  $\mathbb{B}_d$  with priority  $\overline{Q}(\mathbb{B}_d)$  into  $q$ .
16:     Else
17:       Discard  $\mathbb{B}_d$ .
18:     End If
19:   End For
20: End while
```

---

To express the rotation space more compactly, we employ the axis-angle representation for rotation parameterization as in [29]. In the axis-angle representation, a rotation is presented as a vector  $\mathbf{r} \in \mathbb{R}^3$  whose direction and norm specify the axis and angle of the rotation, respectively. We use  $3 \times 3$  matrix  $\mathbf{R}_r$  to denote the rotation matrix corresponding to vector  $\mathbf{r} \in \text{SO}(3)$ . These two representations are associated with each other by *Rodrigues's formula* [18], [29].

With the angle-axis representation, the entire 3D rotation space is represented as a solid  $\pi$ -ball, which could be enclosed within a cube of side  $2\pi$ . Therefore, the rotation search domain lies within a bounded cube  $[-\pi, \pi]^3$ .

We summarize the typical BnB-based algorithm for finding the globally optimal pure rotation registration in **Algorithm 1**. The algorithm recursively explores branches of a tree, which represent subsets of the parameter space. Before evaluating a candidate solution within a branch, the upper bound of this branch is checked against the current best solution, and the branch is discarded if it cannot produce a better solution than the best one found so far.

In **Algorithm 1**,  $\overline{Q}(\mathbb{B})$  is an upper bound of  $Q(\mathbf{R})$  over branch  $\mathbb{B}$ , which satisfies:

$$\overline{Q}(\mathbb{B}) \geq \max_{\mathbf{R} \in \mathbb{B}} Q(\mathbf{R}) \quad (3)$$

The key of **Algorithm 1** is to find an efficient upper bound function  $\overline{Q}(\mathbb{B})$ , where efficiency not only means tightness but also the ability to be evaluated quickly.

In the following subsections, we give four upper bounds on the basis of our formulation of 2D-3D point set registration. We first give the geometry bound of a 3D point's position under rotation in a 3D space and then provide the geometry

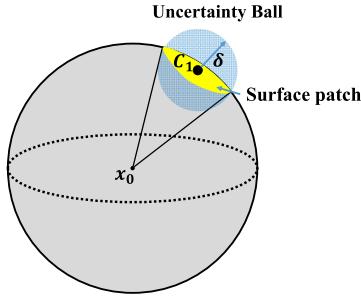


Fig. 2. Geometry bound of a 3D point's position under pure rotation. Given a rotation cube  $\mathbb{B}$ , after an arbitrary rotation  $\mathbf{u} \in \mathbb{B}$  is applied, the position of a point after rotation is confined in a ball, which is called the uncertainty ball. Furthermore, the rotated point lies on a sphere; therefore, it is confined on the yellow surface patch of the sphere contained in the uncertainty ball.  $C_1$  and  $\delta$  are the center and the radius of the uncertainty ball, respectively.

bound of its projection on a 2D projection plane. Finally, the upper bound of (2) is given.

**Bound 1:** First, we revisit the geometry bound of a 3D point under rotation given in [25] by applying the results established in [29], which are also used in [1], [2], and [26] to establish bounds for rotation search. The geometry bound is summarized in **Lemma 1**.

**Lemma 1:** Given a 3D point  $\mathbf{x} \in \mathbb{R}^3$ , the rotation center  $\mathbf{x}_o$ , and a rotation cube domain  $\mathbb{B}$ , let  $\mathbf{p}$  and  $\mathbf{q}$  be the points at two opposite corners of  $\mathbb{B}$ , then the center of the cube  $\mathbb{B}$  is  $\mathbf{c} = 0.5 * (\mathbf{p} + \mathbf{q})$ . For any rotation  $\mathbf{u} \in \mathbb{B}$ , we have

$$\|\mathbf{R}_c(\mathbf{x} - \mathbf{x}_o) - \mathbf{R}_u(\mathbf{x} - \mathbf{x}_o)\| \leq \delta \quad (4)$$

where

$$\delta = \sqrt{2\|\mathbf{x} - \mathbf{x}_o\|^2 \left(1 - \cos\left(\frac{\|\mathbf{p} - \mathbf{q}\|}{2}\right)\right)} \quad (5)$$

Figure 2 shows the geometric explanation for (4). Intuitively, when applied to a rotation in cube  $\mathbb{B}$ , the 3D point  $\mathbf{x}$  rotating around  $\mathbf{x}_o$  must lie within the ball of radius  $\delta$  centered at  $C_1 = \mathbf{R}_c(\mathbf{x} - \mathbf{x}_o) + \mathbf{x}_o$ . This ball is called the uncertainty ball.

Furthermore, [25] point out that the uncertainty ball is rather conservative, and the authors prove that the yellow surface patch shown in Fig. 2 is a tighter bound and provide the following lemma, which is also used in [27].

**Lemma 2:** Given a 3D point  $\mathbf{x} \in \mathbb{R}^3$ ,  $\forall \mathbf{R} \in \mathbb{B}$ , let  $\mathbf{x}_o$  be rotation center; then:

$$\mathbf{R}(\mathbf{x} - \mathbf{x}_o) + \mathbf{x}_o \in \text{Patch} \subseteq \text{Sphere} \quad (6)$$

where  $\text{Sphere}$  represents the uncertainty ball, and  $\text{Patch}$  represents the yellow uncertainty patch in Fig. 2.

Given the geometry bound of a 3D point under the rotation established in **Lemma 1** and **Lemma 2**, we can easily obtain the geometry bound of its projection on the projection plane by projecting its 3D geometry bound onto the projection plane, which leads to **Lemma 3**.

**Lemma 3:** Given a 3D point  $\mathbf{x} \in \mathbb{R}^3$ , for any Cube  $\mathbb{B}$  in the rotation space,  $\forall \mathbf{R} \in \mathbb{B}$ , let  $\mathbf{x}_o$  be the rotation center; then:

$$P(\mathbf{R}(\mathbf{x} - \mathbf{x}_o) + \mathbf{x}_o) \in P(\text{Patch}) \subseteq P(\text{Sphere}) \quad (7)$$

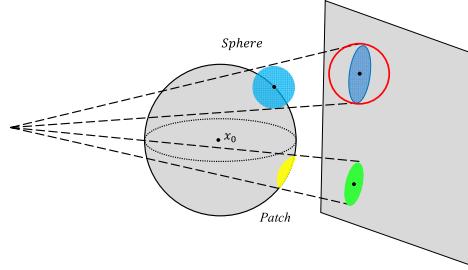


Fig. 3. Projection of uncertainty ball and uncertainty patch. The projection of an uncertainty ball onto the 2D projection plane is an elliptical region, which we call the uncertainty ellipse.

where we abuse the use of  $P(\cdot)$  to represent the set formed by the projection of every point in the point set “.”. **Lemma 3** holds because point  $\mathbf{R}(\mathbf{x} - \mathbf{x}_o) + \mathbf{x}_o$  is an element of the set  $\text{Patch}$  and  $\text{Patch}$  is a subset of  $\text{Sphere}$ . Geometrically, **Lemma 3** can be understood by referring to Fig. 3, which means that the rotated point is projected in a confined region. Then, we obtain the following two bounding functions for the pure rotation objective function (2):

$$\overline{Q}_{\text{patch}}(\mathbb{B}) = \sum_j \max_i \lfloor \|P(\text{Patch}_i) - \mathbf{y}_j\| \leq \epsilon \rfloor \quad (8)$$

$$\overline{Q}_{\text{sphere}}(\mathbb{B}) = \sum_j \max_i \lfloor \|P(\text{Sphere}_i) - \mathbf{y}_j\| \leq \epsilon \rfloor \quad (9)$$

where  $\text{Patch}_i$  and  $\text{Sphere}_i$  are the uncertainty patch and uncertainty ball of the  $i$ -th 3D point in **Lemma 2** and **Lemma 1**, respectively.

**Lemma 4:** For a cube  $\mathbb{B}$  in the rotation space,  $\forall \mathbf{R} \in \mathbb{B}$ ,

$$\max_{\mathbf{R} \in \mathbb{B}} Q(\mathbf{R}) \leq \overline{Q}_{\text{patch}}(\mathbb{B}) \leq \overline{Q}_{\text{sphere}}(\mathbb{B}) \quad (10)$$

Proofs of this lemma and the following theories can be found in the Supplement A.

It is worth noting that using a tighter bound function in the BnB algorithm can be counterproductive if evaluating the bound itself takes too much time. If we use  $\overline{Q}_{\text{patch}}$  as the upper bound in **Algorithm 1**, then calculating  $P(\text{Patch}_i)$  becomes complex because the projection of a surface patch varies with different relative positions between the projection center and the surface patch, which we will discuss in detail when deriving **Bound 4**. Moreover, the irregularly shaped projected region increases the complexity of evaluating  $\|P(\text{Patch}_i) - \mathbf{y}_j\|$ . Therefore, we used the solid uncertainty ball as the geometry bound in the 3D space, whose projection is an elliptical region [30], to derive our first bound for the objective function (2). The scenario where the projection of an uncertainty ball is not an ellipse is discussed in the Supplement B.

Although the elliptical projection region can be calculated efficiently,  $\overline{Q}_{\text{sphere}}$  is still not a good choice of bound function for the BnB algorithm because evaluating this function repeatedly is too time consuming, even though the function  $\|P(\text{Sphere}_i) - \mathbf{y}_j\| = \min_{\mathbf{y} \in P(\cdot)} \|\mathbf{y} - \mathbf{y}_j\|$  is convex. Here, we propose a more efficient geometry bound on the projection plane, which is the first one of our four bounds discussed in this paper. The elliptical geometry bound on the projection plane is extended to a circular region, which is shown as the

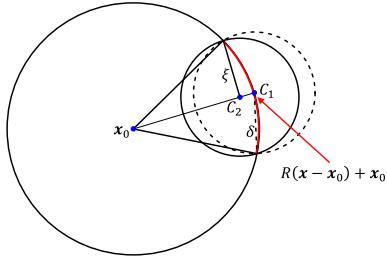


Fig. 4. Illustration of a smaller uncertainty ball in 3D space. The dashed circle represents the uncertainty ball used in **Lemma 1**, and the small solid circle represents a smaller uncertainty ball used in **Theorem 2**.

red circle in Fig. 3, and we call it the uncertainty circle. The uncertainty circle is constructed with the same center as the uncertainty ellipse, and the radius is equal to half of the major axis of the ellipse.

Therefore, we have

$$P(Sphere) \subseteq Circle_{max} \quad (11)$$

where  $Circle_{max}$  denotes the uncertainty circle region that we construct on the projection plane for a projected point as just described. Based on the uncertainty circle, we have the following function, which can be evaluated very efficiently:

$$\overline{Q}_{circle}(\mathbb{B}) = \sum_j \max_i \left[ \|c_1^i - y_j\| \leq \epsilon + r_i \right] \quad (12)$$

where  $c_1^i$  and  $r_i$  are the center and radius of the uncertainty circle, respectively. In (12), evaluating the bound becomes determining the existence of neighbors within a specified distance, which can be implemented quickly. Therefore, we have the following theorem for the first bound proposed in this paper.

**Theorem 1:** For any cube  $\mathbb{B}$  in the rotation space,  $\forall \mathbf{R} \in \mathbb{B}$ ,

$$\max_{\mathbf{R} \in \mathbb{B}} Q(\mathbf{R}) \leq \overline{Q}_{circle}(\mathbb{B}) \quad (13)$$

Thus, the first upper bound  $\overline{Q}_{circle}(\mathbb{B})$  for (2) is obtained, and the pure rotation 2D–3D point set registration problem can be globally solved by using **Algorithm 1**.

**Bound 2:** From the derivation of  $\overline{Q}_{circle}$ , we can see that for an uncertainty ball in the 3D space of a 3D point, we can obtain an uncertainty circle on the 2D projection plane to construct an upper bound. A smaller uncertainty ball will result in a smaller uncertainty circle on the projection plane and a tighter bound for the objective function. Actually, the rotated 3D point can be constrained in a smaller ball as shown in Fig. 4, and the new bound is given in **Theorem 2**.

Given a 3D point  $x \in \mathbb{R}^3$ , a rotation center  $x_0$ , and a rotation cube  $\mathbb{B}$ , let  $p$  and  $q$  be the two points at opposite corners of  $\mathbb{B}$ ; then, the center of  $\mathbb{B}$  is  $c = 0.5 * (p + q)$ . We have an uncertainty ball centered at  $C_1 = \mathbf{R}_c(x - x_0) + x_0$  from **Lemma 1**. We create a tighter ball in **Theorem 2**, and the center  $C_2$  and radius  $\xi$  are:

$$C_2 = x_0 + \mathbf{R}_c(x - x_0) \cos \left( \min \left( \frac{\pi}{2}, \alpha \right) \right) \quad (14)$$

$$\xi = \|x - x_0\| \sin \left( \min \left( \frac{\pi}{2}, \alpha \right) \right) \quad (15)$$

where  $\alpha = 0.5 * \|p - q\|$ .

**Theorem 2:** Given a 3D point  $x \in \mathbb{R}^3$ , a rotation center  $x_0$ , and a rotation cube  $\mathbb{B}$ , for any rotation  $\mathbf{u} \in \mathbb{B}$ , the rotated point is  $x_u = \mathbf{R}_u(x - x_0) + x_0$ ; then, we have

$$\|x_u - C_2\| \leq \xi \quad (16)$$

Geometrically, this theorem states that the rotated point is constrained in an uncertainty ball centered at  $C_2$  with a radius  $\xi$ . This ball is the smallest ball enclosing the uncertainty patch, and this fact seems obvious in Fig. 4.

The new uncertainty ball is smaller than the one in **Lemma 1** because  $\xi < \delta$ . From this smaller ball, we can obtain a smaller uncertainty circle, and we denote the new bound using this circle as follows:

$$\overline{Q}_{circle}(\mathbb{B}) = \sum_j \max_i \left[ \|c_2^i - y_j\| \leq \epsilon + \xi_i \right] \quad (17)$$

where  $c_2^i$  and  $\xi_i$  are the center and radius of the uncertainty circle on the projection plane for point  $x_i, r$ , respectively.

**Theorem 3:** For any cube  $\mathbb{B}$  in the rotation space,  $\forall \mathbf{R} \in \mathbb{B}$ ,

$$\max_{\mathbf{R} \in \mathbb{B}} Q(\mathbf{R}) \leq \overline{Q}_{circle}(\mathbb{B}) \quad (18)$$

**Bound 3:** The two uncertainty balls used in **Bound 1** and **Bound 2** are partially overlapped as illustrated in Fig. 4; therefore, an obvious tighter bound function is the smaller one of the two bounding functions.

$$\hat{Q}(\mathbb{B}) = \min \left( \overline{Q}_{circle}(\mathbb{B}), \overline{Q}_{patch}(\mathbb{B}) \right) \quad (19)$$

**Theorem 4:** For any cube  $\mathbb{B}$ ,  $\forall \mathbf{R} \in \mathbb{B}$ ,

$$\max_{\mathbf{R} \in \mathbb{B}} Q(\mathbf{R}) \leq \hat{Q}(\mathbb{B}) \quad (20)$$

**Bound 4:** Let us reconsider the projection of the uncertainty patch that produces the tightest upper bound in this paper.

$$\overline{Q}_{patch}(\mathbb{B}) = \sum_j \max_i \left[ \|P(Patch_i) - y_j\| \leq \epsilon \right] \quad (21)$$

where  $Patch_i$  is the surface patch generated from  $\mathbf{R}(x_i - x_0) + x_0, \mathbf{R} \in \mathbb{B}$ . When the projection center is on the sphere, the problem becomes a stereographic projection problem studied in [25]. However, the projection center is arbitrary in our problem; therefore, we first need to prove that  $\|P(Patch_i) - y_j\|$  can be evaluated precisely.

Obviously,  $\|P(Patch_i) - y_j\| = \min_{y \in P(Patch_i)} \|y - y_j\|$  can be exactly evaluated if  $P(Patch_i)$  is a convex set. From **Lemma 2**,  $Patch_i$  can be geometrically expressed as a spherical cap, which is not a convex set. However, we can extend the spherical cap to a solid one, which is a convex set and has the same projection region as the nonconvex spherical cap. According to [32], projection preserves the convexity. Therefore, the projected region of the solid spherical cap is convex, which proves that  $P(Patch_i)$  is a convex set and  $\min_{y \in P(Patch_i)} \|y - y_j\|$  can be evaluated exactly. Therefore,  $\overline{Q}_{patch}(\mathbb{B})$  can work as an upper bound. Although this bound is the tightest bound of the four bounds proposed in this paper, the evaluation of  $\overline{Q}_{patch}(\mathbb{B})$  needs to calculate  $\|P(Patch_i) - y_j\|$  for every pair of  $(x_i, y_j)$ , which must call a convex optimization routine, and the total burden of calculation is enormous.

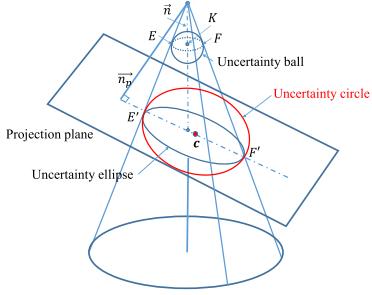


Fig. 5. Uncertainty circle of a point's projection on the projection plane.  $E$ ,  $F$  are two tangent points, and the corresponding projection points are  $E'$ ,  $F'$ . These tangent points are the end points of the long axis of the projection ellipse. All the tangent points form a circle in the 3D projection coordinate denoted as a tangent circle (dashed line), and the center point and radius of this circle are  $K$  and  $r$ , respectively.

*1) Calculating the Uncertainty Circle on the Projection Plane:* To evaluate the first three bounds, we need to find the uncertainty circle that encloses the elliptical projection region of a 3D sphere. Projecting a sphere is a basic problem that has been studied extensively [33]. For example, an algorithm for establishing the analytical equation of the projected ellipse is given in [30]. We do not need the exact ellipse equation but rather a rapid method of determining the center and the radius of the circle. Therefore, we only need to know the two end points of the long axis of the projection ellipse, which is also the diameter of the circle. As illustrated in Fig. 5, the center and the radius of the circle can be determined by the following four steps.

Step (1): Calculate the norm  $\vec{n}$ , the radius  $r$  and center  $K$  of the tangent circle, and the unit normal vector  $\vec{n}_p$  of the projection plane.

Step (2): Calculate  $\vec{EF} = \vec{n} \times (\vec{n} \times \vec{n}_p)$  and the tangent points  $E$  and  $F$ :  $E = K + r \frac{\vec{EF}}{\|\vec{EF}\|}$ ,  $F = K - r \frac{\vec{EF}}{\|\vec{EF}\|}$ .

Step (3): Calculate  $E'$ ,  $F'$  by the projection function, where  $E'F'$  is a diameter of the uncertainty circle on the projection plane.

Step (4): Calculate the center and radius of the uncertainty circle with the diameter  $E'F'$ .

### C. Efficient Searching in SE(3)

After solving the more difficult rotation search problem, extending the solution to search the SE(3) space is usually straightforward, such as by enlarging the radius of the uncertainty ball by the size of translation and searching the rotation space and the translation space together [25], [26] or in a nested way [1], [2]. When calculating the first three bounds in this paper, we can also enlarge the 3D uncertainty ball before projecting the bounds onto the projection plane to obtain the bounds for the parameter space in SE(3). However, we propose a more efficient method of searching the SE(3) space based on our new problem formulation and our insights into the rotation search algorithm.

*1) Robustness of Rotation Search Against Translation Perturbation:* Figure 6 illustrates the robustness of the objection function (2) against the translation perturbation of a 3D point.

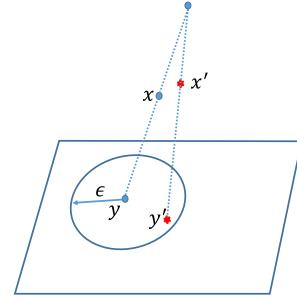


Fig. 6. Robustness of the objective function against translation perturbation.  $x$  is a 3D point, and  $y$  is its projection. In 3D space,  $x$  is translated to  $x'$  and  $y'$  is the projection of  $x'$ .  $\epsilon$  is the inlier threshold.

In the 3D projection coordinate system, a 3D point  $x$  is translated to a new position  $x'$  by a translation vector in  $\mathbb{R}^3$ , and its projection point is translated to a new position  $y'$  on the projection plane. Although the new projection point  $y'$  deviates from the original 2D point  $y$ , which is the projection of  $x$ , our objective function does not depend on the absolute distance between the 2D point  $y$  and the new projected point  $y'$ , and its value remains unchanged only if the distance between  $y$  and  $y'$  is still less than the inlier threshold  $\epsilon$ . Therefore, when we translate the 3D points by a short distance from their true position, the objective function based on the inlier set on the 2D projection plane remains unchanged. Therefore, if  $R^* \in SO(3)$  and  $t^* \in \mathbb{R}^3$  are the true rotation and translation, respectively, then there is a region around  $t^*$  in  $\mathbb{R}^3$  within which the global optimal rotation search by **Algorithm 1** returns the same value. We say that the rotation search algorithm covers a region in the 3D translation space, and in the covering region, we need only run the rotation search at one point. We show this phenomenon visually in the Experiment section.

*2) Multi-Start Grid Search and Fast Searching Schema:* Since the rotation search algorithm can cover a region in the translation space, we need not perform the rotation search at every point in  $\mathbb{R}^3$ , which is impossible. Instead, we only need to search the translation space block by block, and we can obtain the global optimal result in SE(3) if we keep the searching blocks smaller than the covering region of the rotation search algorithm. In practice, we divide the translation space into a series of blocks and run a rotation search at the center of each block. Equation (22) shows the multi-start search in SE(3) mathematically:

$$\max E(\mathbf{R}, \mathbf{t}) = \max_{h=1 \dots k} (\max_{\mathbf{R} \in SO(3)} E(\mathbf{R}, \mathbf{t}_h)) \quad (22)$$

where  $\mathbf{t}_h$  is the center of the  $h$ -th searching block, and  $E(\mathbf{R}, \mathbf{t}_h)$  is  $Q(\mathbf{R})$  after translating all the 3D points with  $\mathbf{t}_h$ . The size of the covering region depends on the value of the inlier threshold  $\epsilon$  in (2), the projection parameters, and the position of the 3D points. A specific value of the size of the covering region can be calculated from these parameters. In theory, the size of the covering region may be infinitely small, but in practice, we can choose a fairly large block size to achieve high efficiency. Although guaranteed global optimality is lost by this search scheme, the searching

speed is significantly improved, especially when we adopt a synchronized searching scheme.

In our formulation of the 2D–3D point set registration problem, when the 3D and 2D point sets are given, the possible position of the 3D point sets is highly confined so that we do not need to search a large region in the translation space. For example, we can easily find the smallest circle on the 2D projection plane that encloses all the 2D points. In most cases, searching the translation in the cone formed by connecting this circle and the projection center is safe. Therefore, in practice, we do not need to search too many blocks.

Furthermore, when we divide the searching region in the translation space into different blocks and run one global rotation search in each block, in the block that covers the global optimal solution, the rotation search converges very fast, whereas in blocks far away from the global optimal solution, the rotation search converges much more slowly. We will show this phenomenon in the Experiment section. The reason for this disparity is that in the blocks far from the global optimal solution, the upper bound and the current best value of the objective function touch at a very small value and the rotation space needs to be divided into very small branches to decrease the upper bound to that value. As a result, if we naively run the rotation search in each block one by one, most of the runtime will be spent on meaningless rotation search in marginal blocks. Here, we propose a synchronized searching schema in which we run all the rotation searches in a synchronized way and keep the largest one of the current best value of each rotation search. A rotation search is terminated and the corresponding block is discarded when its upper bound decreases to a value smaller than this universal best value of the objective function. In this way, the useless searching at distant blocks is largely avoided. Experiments show that the overall speed of this synchronized searching schema is almost greater than a magnitude faster than the naive one-by-one searching scheme. The algorithm of this synchronized searching of the SE(3) space is shown in **Algorithm 2**.

#### D. Local Refinement

Objective function based on the cardinality of inlier set makes the corresponding points fall within an inlier threshold, but it cannot drive them closer. Therefore, many algorithms using this objective function utilize a local refinement to further improve the solution. In this paper, we use a local refinement method based on the Gaussian mixture model (GMMreg algorithm) [21]. GMMreg aligns two Gaussian mixtures constructed from the two point sets to be registered. This method belongs to soft assignment registration, and it has a large basin of convergence on the 2D–3D point set registration [6]. For details on the GMMreg algorithm, refer to [6] and [21]. The objective function of GMMreg is different from that of POSO, which is our global optimal pose optimization algorithm in SO(3), or POSE, which is our pose optimization algorithm in SE(3); therefore, local refinement does not guarantee optimality in terms of cardinality of the inlier set. However, GMMreg guarantees finding a better solution than the one obtained by POSO or POSE in terms of GMMreg’s objective function.

---

**Algorithm 2** Pose Optimization in SE(3)

---

**Input:** 3D point set  $X$ , 2D point set  $Y$ , threshold  $\epsilon$  and translational cube  $\mathbb{V}$ .  
**Output:** Optimal rotation  $\mathbf{R}^*$  and  $\mathbf{t}^*$  with quality  $E^*$ .

- 1: Subdivide  $\mathbb{V}$  into cubes  $\{\mathbb{V}_h\}_{h=1}^k$  and the center of each cube  $\{\mathbf{t}_h\}_{h=1}^k$ ;  $\{E_h\}_{h=1}^k \leftarrow 0$  and global  $E^* \leftarrow 0$ ; global upper bound  $\hat{E}$ ;  $\{\mathbb{B}_h\}_{h=1}^k \leftarrow$  cube of size  $2\pi$  and  $\{\mathbf{R}_h\}_{h=1}^k \leftarrow \mathbf{I}$ .
- 2: Insert  $\{\mathbb{B}_h\}_{h=1}^k$  with priority  $\{\bar{E}(\mathbb{B}_h, \mathbf{t}_h)\}_{h=1}^k$  into  $\{q_h\}_{h=1}^k$ .
- 3: **While**  $\{q_h\}_{h=1}^k$  is not empty **do**:
- 4:   **For each** non-empty  $q_h$ :
- 5:     Obtain highest priority cube  $\mathbb{B}_h$  from  $q_h$ .
- 6:      $\mathbf{R}_c^h \leftarrow$  Center rotation of  $\mathbb{B}_h$
- 7:     **If**  $E(\mathbf{R}_c^h, \mathbf{t}_h) > E_h^*$  **Then**
- 8:       Update  $E_h^* \leftarrow E(\mathbf{R}_c^h, \mathbf{t}_h)$ ,  $\mathbf{R}_h^* \leftarrow \mathbf{R}_c^h$
- 9:     **End If**
- 10:    **If**  $\bar{E}(\mathbb{B}_h, \mathbf{t}_h) = E_h^*$  **Then**
- 11:       $q_h \leftarrow \phi$
- 12:     **End If**
- 13:   **End For**
- 14:    $\hat{E} \leftarrow \max\left(\{\bar{E}(\mathbb{B}_h, \mathbf{t}_h)\}_{h=1}^k\right)$
- 15:    $E^* \leftarrow \max(E^*, \{E_h^*\}_{h=1}^k)$
- 16:   **If**  $\hat{E} = E^*$  **Then**
- 17:      Terminate
- 18:   **End If**
- 19:   **For each** non-empty  $q_h$ :
- 20:     Subdivide highest priority cube  $\mathbb{B}_h$  into eight cubes  $\{\mathbb{B}_h^d\}_{d=1}^8$
- 21:     **For each**  $\mathbb{B}_h^d$
- 22:       **If**  $\bar{E}(\mathbb{B}_h^d, \mathbf{t}_h) \geq E^*$ , **Then**
- 23:          Insert  $\mathbb{B}_h^d$  with priority  $\bar{E}(\mathbb{B}_h^d, \mathbf{t}_h)$  into  $q_h$
- 24:       **Else**
- 25:          Discard  $\mathbb{B}_h^d$
- 26:       **End If**
- 27:     **End For**
- 28:   **End For**
- 29: **End While**

---

## IV. EXPERIMENT

We evaluate the performance of the proposed algorithms and compare them to state-of-the-art algorithms using both synthetic data and real data. Our global optimal pose optimization algorithm in SO(3) is denoted as POSO, and our pose optimization algorithm in SE(3) is denoted as POSE. A local refinement is performed after POSO and POSE when it is needed, and we will not emphasize this point for simplification.

For the POSO algorithm, we used synthetic data to study its properties and compare its performance against state-of-the-art local and global methods. Performance comparison was made between POSO and three local methods, SoftPOSIT [4], GMMreg and simulated annealing and one global method, fast rotation search [25]. The SoftPOSIT and fast rotation

search codes are made available by the authors. GMMreg is an extension of the 3D-3D point set registration algorithm [21], whose convergence range is affected by the standard deviation  $\sigma$  of the Gaussian kernels. We implemented the GMMreg algorithm in MATLAB. Both SoftPOSIT and GMMreg are local search algorithms. Simulated annealing is a heuristic search method to solve nonconvex problems, and we applied the naive “simannealbnd” in MATLAB to optimize the GMMreg’s objective function in our experiments, and this implementation is denoted as SA.

For the POSE algorithm, we first showed the covering range of POSO in the translation space and the time efficiency of the proposed synchronized multi-start searching scheme in POSE. Then, we compared the performance of POSE against state-of-the-art methods, including four local methods, SoftPOSIT, GMMreg, RANSAC and SA and one global method, GOPAC [1] using both synthetic and real data. GOPAC is a global optimal 2D-3D point set registration method, and its code is made available by the author.

We have implemented the POSO and POSE algorithms in MATLAB 2016B, and the code can be found online<sup>1</sup>. All experiments were performed on a desktop computer (Intel(R) Xeon(R) CPU E5-1620 v3 @ 3.50 GHz). GOPAC was run on the Ubuntu 16.04 platform.

#### A. Pose Optimization in $SO(3)$

In this section, we study the performance of the POSO algorithm in registering 2D and 3D point sets in the parameter space of  $SO(3)$ , which means that the 3D points are only rotated to match the 2D points. We first show the convergence of POSO and its efficiency by using the four different bounds proposed in Section III.B. Then, we studied its efficiency with different inlier thresholds used on the 2D projection plane and showed its global optimality. Finally, we showed its robustness to noise and outliers and compared its efficiency against both local and global methods.

*1) Convergence and Efficiency by Using Different Bounds:* To illustrate the convergence of the POSO algorithm with all four bounds, we used POSO to register a pair of 2D and 3D synthetic point sets. In the projection coordinate system, the projection center was chosen at the origin, and the 2D projection plane was placed at point  $(0, 0, 150)$ . In a cube centered at point  $(0, 0, 50)$  with an edge size of 10, we generated 20 uniformly distributed random 3D points and projected 17 of them onto the projection plane. The 3D point set was rotated by  $(0, \pi/3, 0)$  and registered back to the 2D point set.

POSO was run four times by using the four different upper bounds separately. Fig. 7 (a) plots the evolution of the upper bound and the current best value of the objective function, which is the lower bound of the global maximum. We can see that all four runs converge at the correct number of inliers, and fewer iterations were needed to converge for tighter bounds.

Although Fig. 7 (a) indicates that bound 4 is the tightest one of the four bounds, a tight bound is not necessarily the most efficient bound in a BnB optimization framework. The bound

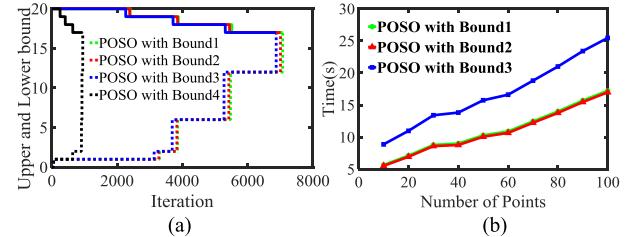


Fig. 7. Convergence and runtime of POSO using four different bounds. (a) Evolution of the upper (solid line) and current best value of the objective function (dashed line). (b) Runtime with respect to the number of points.

must be evaluated many times; therefore, the efficiency of evaluating the bound function also plays a large role in the overall efficiency. In the following experiment, we examined the overall efficiency of POSO with each bound. The same projection setting used in the previous experiment was applied. Different numbers of uniformly distributed 3D points were randomly generated in the 3D cube, and all of them were projected onto the projection plane to obtain the 2D point sets. For each value of the 3D point numbers, registration was repeated 100 times with random rotations of the 3D points, and the average runtime against the point number is plotted in Fig. 7 (b). We do not plot the runtime of POSO with bound 4 because it is too slow. In our experimental environment, more than 580 s is required to register two point sets (10-to-10) by using bound 4. Although bound 3 and 4 are tighter, using bound 1 or 2 is much faster because bound 1 and 2 can be rapidly evaluated in each iteration. Although the evaluating efficiencies of bound 1 and 2 are equivalent, bound 2 is tighter than bound 1; therefore, it is the fastest in all four bounds. In the following experiments, bound 2 is used in both the POSO and the POSE algorithms.

*2) Runtime of POSO With Different Inlier Thresholds and Its Deterministic Global Optimality:* In this section, we first investigate the efficiency of POSO and the influence of the inlier threshold  $\epsilon$  on its runtime. The projection center was at the origin, and the projection plane was placed at point  $(0, 0, 1000)$ . Uniformly distributed random 3D points were generated in a cube of edge size 100 centered at  $(0, 0, 600)$ , and they were projected onto the projection plane to generate the 2D point set. We chose four values for  $\epsilon$ : 0.1, 1, 5 and 10. For each  $\epsilon$  and each number of points in the point sets to be registered, 20 registrations were run, and the average convergence time was recorded. For each registration, the 3D points were first rotated by a random rotation in  $SO(3)$  and registered back to their corresponding 2D point set. The average runtimes with respective to point number are plotted in Fig. 8 (a). A smaller inlier threshold  $\epsilon$  corresponds to a faster runtime of the algorithm; moreover, and the POSO scales well when the inlier threshold is not very large.

Then we show the deterministic global optimality of the POSO algorithm. In this experiment, we rotated the 3D points in the range of  $[-180, 180]$  degrees with an interval of two degrees and registered them back to the 2D point sets obtained from projecting the 3D point set from its original position. For every rotation angle, we generated 100 rotations along

<sup>1</sup> Available online: <http://www.fudanmiccai.org/col.jsp?id=112>

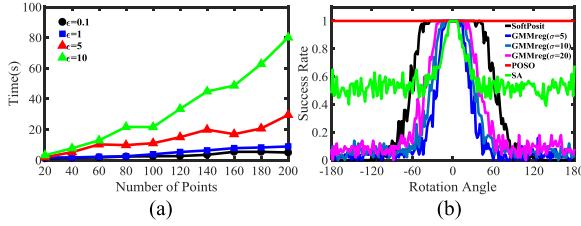


Fig. 8. Runtime and global optimality of POSO. (a) Runtime of POSO with respect to the point number under different inlier thresholds used on the 2D projection plane. (b) Success rate of POSO, SoftPOSIT, SA and GMMreg with three different standard deviations.

random rotation axis and applied each them to the 3D points. The projection plane was placed at point  $(0, 0, 1000)$ , and in each registration, 20 uniformly distributed random 3D points were generated in a cube of edge size 100 centered at point  $(0, 0, 600)$ . We used an inlier threshold  $\epsilon = 1$ . We were able to calculate the target registration error (TRE) on the projection plane for each point since we knew the true correspondence. A registration with a root mean square TRE less than 1 was regarded as successful, and the success rate with respect to rotation angle is plotted in Fig. 8 (b). We also ran SoftPOSIT, SA and GMMreg with three different standard deviations of their Gaussian kernel using the same data and plotted the success rate in the same figure.

Fig. 8 (b) shows that POSO achieves a 100% success rate for all degrees of rotation and the other methods can achieve a 100% success rate only within a small rotation angle. The success rate of SoftPOSIT is 100% in a range less than  $[-30^\circ, 30^\circ]$  and drops rapidly when the rotation angle exceeds this range. The success rate drops to near zero when the rotation angle increases to  $100^\circ$ . The success rates of the three GMMreg algorithms with different standard deviations of their Gaussian kernels are slightly lower than that of SoftPOSIT, and the success rate increases slightly when we increase the standard deviation. For SA, the success rates are approximately 50% for large rotation angles, and a 100% success rate can only be achieved in a small angle range. This result clearly shows that the heuristic search method can improve the convergence of the local search algorithm; however, global optimality still cannot be guaranteed.

3) *Robustness to Noise and Outliers:* In this section, we compare POSO against SoftPOSIT, SA and GMMreg in more challenging scenarios, including noise on the points' position and outliers in 2D or 3D point sets. We divided the rotation angle into three ranges: small rotation  $[-30^\circ, 30^\circ]$ , medium rotation  $[-90^\circ, -30^\circ]$  and  $[30^\circ, 90^\circ]$  and large rotation  $[-180^\circ, -90^\circ]$  and  $[90^\circ, 180^\circ]$ . For each rotation range, a random rotation angle was generated within the corresponding range. The same projection parameters as in the previous section were used in this experiment. Thirty uniformly distributed random 3D points were generated in the projection coordinate system, and they were projected to the projection plane to generate the original 2D point set. For the noise experiment, different levels of Gaussian noise were added to the coordinates of the 2D points. In the 2D/3D outlier experiments, different numbers of uniformly distributed random outliers

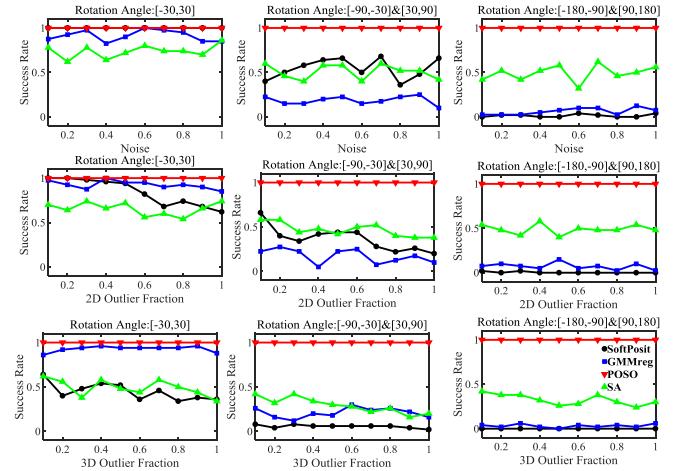


Fig. 9. Success rate of registering point sets with noise and outliers. First row: noise experiment. Second row: 2D outlier experiment. Third row: 3D outlier experiment. The outlier fraction is the ratio between the outlier points added and the inlier points.

were added on the projection plane or to the 3D point set, and Gaussian noise with a standard deviation of 1 was added to the inlier 2D points. In all these experiments, a registration is regarded as successful when the root mean square TRE is less than four times the standard deviation of the noise. In this experiment, we use an inlier threshold  $\epsilon$  of 5 in our algorithm to address possible large noise of the point location. The standard deviation of the GMMreg algorithm was set as 10, and the initial annealing parameter and maximum iterations of SoftPOSIT were set to 0.05 and 100, respectively. For each parameter setting, 50 registrations were performed, and the average success rate was recorded. The results are plotted in Fig. 9.

POSO achieved a 100% success rate in all scenarios. GMMreg, SA and SoftPOSIT obtained high, but not 100%, success rates for small rotation ranges. In the middle rotation range, the success rates of these algorithms were smaller than in the small rotation range. In the large rotation range, GMMreg and SoftPOSIT failed in most cases, whereas SA still had a similar success rate to that of the middle rotation range.

4) *Efficiency Comparison Against State-of-the-Art Local and Global Methods:* We first compare the efficiency of POSO and three local methods in registering 2D and 3D point sets under the condition of noise and outliers. The three local methods are SoftPosit, GMMReg and SA, and the same parameter setting was used as in the previous section. The 2D and 3D points with noise and outliers were generated in the same way as in the previous section and the same projection parameters were utilized. For each noise level and outlier fraction, we generated 50 pairs of 2D and 3D point sets and registered them with different methods. To achieve a fair comparison, we run each local method multiple times for a pair of 2D and 3D point sets so that its total runtime was equal or longer than the runtime of POSO, and the best result was used to determine if the registration was successful. The success rates are illustrated in Fig. 10. We can see that in most cases, POSO achieves a higher success rate than the local methods in equal or less time.

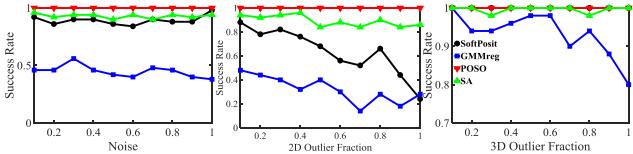


Fig. 10. Success rate of POSO and three local methods. The outlier fraction is the ratio between the outlier points added and the inlier points. For each pair of 2D and 3D point sets, multiple times are run for each local method so that its total runtime is equal or longer than the runtime of POSO.

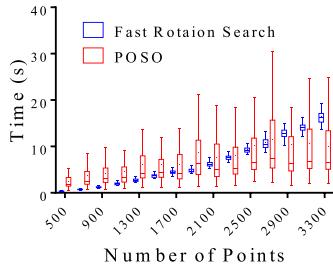


Fig. 11. Running of POSO and Fast Rotation Search with respect to point number.

We then compared the efficiency of POSO and another global optimal rotation search algorithm [25], which we denote as Fast Rotation Search. Fast Rotation Search is a 3D-3D point set registration algorithm, and it cannot be directly used to perform 2D-3D point set registration. Therefore, we performed this comparison as follows. We set the projection center at the origin and the 2D projection plane passed through point  $(0, 0, -10)$ . Random numbers of 3D points were generated in a cube with an edge size of 1 centered at  $(0, 0, -5)$ . The 3D points were first projected onto the projection plane and then rotated by a random rotation. For POSO, the rotated 3D points were directly registered to the projected 2D points. For Fast Rotation Search, the rotated 3D points and the projected 2D points were first projected back to the unit sphere around the projection center to form two 3D point sets on the sphere, and then the two 3D point sets were registered to each other. For different point numbers, we run each registration 20 times, and the statistics of the runtimes are plotted in Fig. 11. The inlier threshold of POSO was set at 0.01, and the inlier threshold of Fast Rotation Search was set at 0.01 in this experiment. We can see that Fast Rotation Search is faster when the point number is small, and as the point number becomes larger, POSO outperforms Fast Rotation Search.

### B. Pose Optimization in SE(3)

In this section, we examine the performance of the proposed POSE algorithm in registering 2D and 3D point sets in the parameter space of  $\text{SE}(3)$ , which means that the 3D points are both rotated and translated to match the 2D points. We first show the robustness of the POSO algorithm against the perturbation of translation, which is the basis of our block search method. The runtime of POSE with respect to the point number and the effectiveness of the synchronized iteration scheme were then analyzed. Finally, we evaluated the performance of POSE and compared it against three

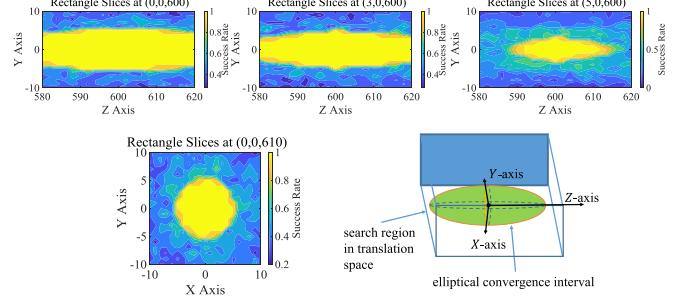


Fig. 12. Success rate of the POSO algorithm followed by a 6DOF GMMreg refinement at different perturbation positions. First row: success rate on three rectangular areas perpendicular to the X-axis. Second row: success rate on rectangular area perpendicular to the Z-axis (left) and illustration of the ellipse convergence region (right).

local methods, SoftPOSIT, SA and GMMreg; and one global method, GOPAC, using both synthetic and real data.

**1) Robustness of POSO Against Translation Perturbation:** We ran POSO at different positions in the projection coordinate system to see if the true 6D transformation parameters could be successfully restored by the global optimal rotation search in  $\text{SO}(3)$  followed by a local refinement in  $\text{SE}(3)$ . As shown in Fig. 12, the 3D points were perturbed within four rectangular slices in the projection coordinate system, three of which are perpendicular to the X-axis and one is perpendicular to the Z-axis. The first three rectangles are centered at  $(0, 0, 600)$ ,  $(3, 0, 600)$  and  $(5, 0, 600)$ , and the lengths in the Y and Z directions are 20 and 40, respectively. The last rectangle is centered at  $(0, 0, 610)$ , and the lengths in the X and Y directions are both 20. We put sampling points on a regular grid in these rectangles, with a Y spacing of 1 and a Z spacing of 2 in the first three rectangles and with a spacing of 1 in both the X and the Y direction in the last rectangle. We randomly generated 20 3D points in a cube of  $[-50, 50]^3$  centered at  $(0, 0, 600)$  and projected them onto the projection plane with the same parameters as in the previous section to generate the 2D point set. Then, the center of the cube containing the 3D points was placed at each sampling point to generate the perturbed 3D point sets, and each perturbed 3D point set was further rotated randomly and then registered back to the 2D point set by POSO, followed by a 6D local refinement by using GMMreg with a standard deviation of 10. The registrations with a root mean square TRE on 2D plane less than 3 were regarded as successful registrations. At each sampling point, 40 registrations were performed, and the success rates are shown in Fig. 12 with color coding.

Fig. 12 shows that around the true position of the 3D point set corresponding to the 2D point set, i.e., point  $(0, 0, 600)$ , a large region exists in which the success rate is 100%. This region is similar to an ellipsoid, and its long axis is parallel to the Z axis. This shape provides guidance on choosing the block size in the POSE algorithm to guarantee that the POSO algorithm is run in the region with a 100% success rate. For example, we should use small sizes in the X and Y directions, whereas larger block sizes can be used in the Z direction.

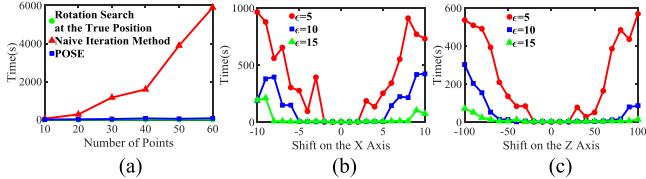


Fig. 13. (a) Runtime of POSE versus the number of the points. (b) Runtime of POSO at different distances from the true position along X axis. (c) Runtime of POSO at different distances from the true position along Z axis.

### 2) Effectiveness of the Synchronized Iteration Scheme:

In this section, we study the runtime of POSE with respect to the number of points to be registered and show the efficiency of our synchronized iteration scheme. We used the same projection parameters as in the previous section, and the 2D and 3D points had one-to-one correspondence. There is no noise or outliers in these points. The average runtime of 10 registrations by POSE with respect to point number is illustrated in Fig. 13 (a). In Fig. 13 (a), we also plot the average runtime of a single rotation search by POSO at the true translation position and the runtime of 6DOF registration using the naive iteration method, in which the rotation search by POSO runs one by one at the center of each block. The runtime of POSE is approximately the runtime of the rotation search at the true position multiplied by the number of searching blocks in the translation space. Fig. 13 (b) and (c) show the runtime of a single POSO at different positions in the translation space, and we can see that the runtime of the rotation search is the shortest within a range around the true translation position. In POSE, all rotation searches run in a synchronized way, so the largest current best value of all the rotation searches can be used to terminate other rotation searches. On the other hand, if we run all the rotation search one by one, time will be wasted on blocks that are far from the true translation position. Fig. 13 (a) shows the significant runtime improvement of this synchronized searching scheme against the naive one-by-one searching scheme.

**3) Performance Comparison Against State-of-the-Art Local and Global Methods:** In this section, we first compare the efficiency of POSE and four local methods in 6DOF 2D–3D point set registration using synthetic data with noise and outliers. The four local methods include SoftPOSIT, SA, GMMreg and RANSAC. In the RANSAC approach, a transformation is calculated by utilizing the P3P solver provided in MATLAB [45] with four randomly selected 2D–3D correspondences, and in a number of random tries the transformation with the greatest number of inliers was kept. The clean point sets were generated in the same way as in the noise and outlier experiment for POSO, and the only difference is that a random translation in the range of  $[-5, 5]^3$  was used to the 3D points before registration. In the 2D and 3D outlier experiments, Gaussian noise with a standard deviation of 1 was added to the coordinates of the inlier points. POSE was run by using nine searching blocks obtained by dividing the searching region  $[-5, 5]^3$  into three parts in both the X and Y directions. For a fair comparison, multiple runs of SoftPOSIT, SA and GMMreg with random initialization were used so that the overall runtimes for these three algorithms

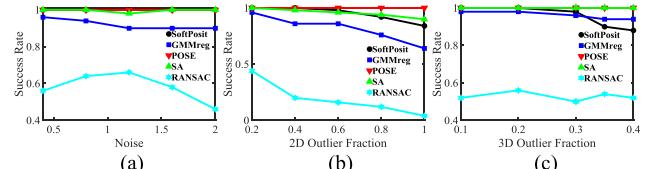


Fig. 14. Success rate for 6DOF 2D–3D registration with noise and outliers. Outlier-free point sets consist of 30 points. (a) Registration with noise on 2D point position. (b) Registration with 2D outlier points. A 0.2 outlier fraction means that  $0.2 \times 30 = 6$  randomly distributed points were added to the 2D point sets. (c) Registration with 3D outliers. A 0.2 outlier fraction means the  $0.2 \times 30 = 6$  of the 3D points were not projected on to the 2D projection plane so that the 2D point set has 6 points less than the 3D point set.

were all approximately the same as that of POSE. For the 2D outlier experiments, SoftPOSIT, SA and GMMreg ran 300, 5 and 100 times, respectively. For the 3D outlier experiments, SoftPOSIT, SA and GMMreg ran 600, 10 and 200 times, respectively. The RANSAC approach ran 500,000 iterations, which cost more time than the other methods. The inlier threshold  $\epsilon$  was set to 10 in POSE. The standard deviation of the GMMreg algorithm for comparison was set to 10. For every experimental configuration, 50 registrations with random rotation and translation in SE(3) were performed and the results are shown in Fig. 14. POSE achieved a 100% success rate in all cases, whereas the other four local cannot.

Then we compare the performance of POSE and GOPAC [1], which is a global optimal 2D–3D point set registration method. We compared the registration accuracy and runtimes of the two algorithms under different distances between the projection center and the 3D points. The projection center is at the origin, and the 2D projection plane passes through the point  $(0, 0, 40)$ . A set of 30 uniformly distributed random 3D points were generated in a cube with an edge size of 10. The center of the cube containing these 3D points was placed at different positions of the Z axis. To compare POSE and GOPAC fairly, we generated the translation and rotation of the 3D points according to the specific problem formulation of POSE and GOPAC. For POSE, we perturbed the 3D point set with a random translation in the range of  $[-0.5, 0.5]$  of each coordinate of each point, and then we rotated the points around its centroid randomly. POSE was used to restore the random rotation and translation. For GOPAC, we first disturbed the projection center with a random translation in the range of  $[-0.5, 0.5]$  of each coordinate, and then we rotated the 3D points around the new projection center randomly. GOPAC was used to restore the projection center and the rotation angle. The size of the translation and the rotation space that the two algorithms searched was the same. At each position of the cube containing the 3D points at the Z-axis, we ran each registration algorithm 50 times and recorded the runtime and the residual rotation and translation error. The inlier threshold of GOPAC was set as 1 degree, and the inlier threshold of POSE was set as 1. The average residual rotation and translation errors with respect to the position of the cube containing the 3D points are shown in Fig. 15 (a) and (b), respectively. The average runtime with respect to the position of the cube is shown in Fig. 15 (c). We found that GOPAC lost its ability to register the

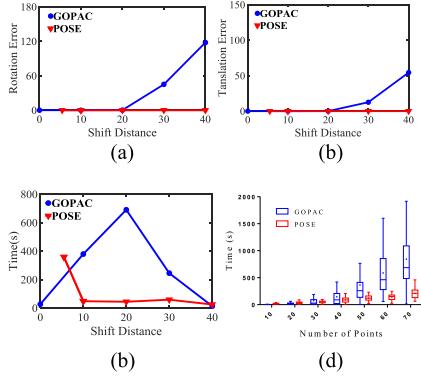


Fig. 15. Average residual rotation (a) and translation (b) error of POSE and GOPAC with respect to different distances between the projection center and the 3D points. The horizontal axis represents the position of the cube containing the original 3D points at the Z axis. (c) Median runtimes of POSE and GOPAC with respect to different distances between the projection center and the 3D points. (d) Runtimes of POSE and GOPAC with respect to point number.

3D points to the 2D points at shift distances greater than 20. However, POSE was able to register all 3D and 2D point sets. GOPAC was faster than POSE when the 3D points were very close to the projection center, whereas POSE was much faster when the 3D points were farther away.

Next, we compared the runtime of POSE and GOPAC with respect to different point numbers. The experimental setup was the same as that in the previous experiment, and the time comparison was performed when both POSE and GOPAC can successfully register the two point sets. The results in Fig. 15(d) show that when the point number was less than 40, GOPAC was slightly faster than POSE. However, when the point number was larger than 40, POSE was significantly faster than GOPAC.

**4) Application in Registering 2D and 3D Vessels:** 2D-3D point set registration is widely used in image-guided interventions [8], [37], including interventions on the head [7], [38], heart [5], [6], liver [39], [40], etc. In these applications, 3D vessels are extracted from 3D-DSA, CT or MRI images preoperatively, and then they are registered to intraoperative 2D vessels extracted from X-ray or 2D-DSA images to establish spatial correspondence between the preoperative and intraoperative images. In this section, we use POSE in a real application of 2D-3D registration of cerebral angiograms [7]. First, we briefly explain the background. Minimally invasive endovascular image-guided interventions (EIGIs) have been developed for the treatment of many types of cerebrovascular diseases, and they involve the navigation of a catheter through the vasculature followed by the application of the treatment at the site of the abnormality using live 2D projection images for guidance. The 3D images acquired prior to the EIGI are used to quantify the vascular abnormality and plan the intervention. If fused with the information of live 2D images, these 3D images can also facilitate navigation and treatment. For this purpose, 2D-3D registration is required in EIGI. The clinical dataset is publicly available from [7].

The data consists of 3D-DSA and 2D-DSA images of ten patients. Highly accurate “gold standard” registrations were obtained by aligning fiducial markers. Therefore, the projection parameters and the true position of the 3D and 2D images are known. We extracted vessel points from the 3D DSA images by the following process: first, we segmented the data into vessels and background by thresholding; second, we executed the morphological operation “open” on the segmented vessels, grouped the remaining points into connected regions and kept only the largest one; then, we utilized the multistencils fast marching method (MSFM) to extract the centerline of the vessels as the 3D point set [41]–[43]; finally, grid down-sampling was used to reduce the number of the 3D points to a manageable size. In several cases, we manually edited the centerlines to remove branches that corresponded to thin vessels. The vessel points on the 2D-DSA images were extracted in the following way: the 2D images were preprocessed by Gaussian filtering and contrast-limited adaptive histogram equalization; then, a Hessian-based Frangi vesselness filter was used to enhance the contrast between the vessels and background [44]; finally, the vessels were segmented by thresholding, with the largest connected region retained and the centerline points extracted via MSFM [41]–[43]. The number of the extracted 3D points ranged from 63 to 147, and the number of the extracted 2D points ranged from 292 to 907 for the 10 cases. The concrete number of 2D and 3D points for each case is illustrated in Table 1. These points were used in the following 2D-3D registration experiment.

Following the method of van de Kraats *et al.* [7], rigid-body translations and rotations were randomly sampled in the range of  $[-20, 20]$  millimeters in each direction and  $[-10^\circ, 10^\circ]$  along a randomly generated axis, respectively, to generate the initial displacements of the pre-EIGI 3D points from the “gold standard” position. This random displacement represents the uncertainty of the position of the patient relative to the 2D-DSA imaging device during the operation. We ran 40 repeated experiments for each case, and the results are listed in Table 1.

Figure 16 shows three illustrative registration results. The average rotation error and the average translation error of RANSAC, SoftPOSIT5 and GMMReg are very large, and this means that these three methods failed in most of the registrations. The average rotation error and the average translation error of SA, POSE without refinement and POSE are all small, which means that these three methods succeeded in all the registrations, but the proposed methods are much faster than SA. For each registration, the 2D TRE means the average distance between the projection of each 3D point to its nearest neighboring 2D point. Very small average 2D TRE was achieved by POSE for each case, which means the 2D inlier set was maximized. Please note that a small 2D TRE itself does not mean that the registration is accurate, and on the contrary the registration may be totally wrong, as illustrated in the last column of Fig. 16. In this experiment, we divided the translation searching range into 500 blocks. By comparing the result of GMMReg and POSE followed by GMMReg refinement, we can see that by adding several seconds spent on

TABLE I

2D-3D DSA REGISTRATION RESULT FOR 10 CASES. FOR EACH CASE, THE NUMBER OF POINTS IN 2D AND 3D DATA SET ARE LISTED TOGETHER WITH THE AVERAGE ROTATION ERROR, AVERAGE TRANSLATION ERROR, AVERAGE RUNTIME AND AVERAGE 2D TRE IN 40 RANDOM REGISTRATIONS FOR EACH METHOD

	Case	1	2	3	4	5	6	7	8	9	10
Rotation Error( $^{\circ}$ )	SA	2.57	3.80	<b>6.57</b>	3.23	<b>2.59</b>	6.02	2.21	9.79	3.77	2.62
	RANSAC	147.79	122.04	128.40	129.82	137.83	143.50	148.43	139.03	135.61	101.01
	SoftPOSIT5	73.54	45.13	133.15	110.98	120.68	141.45	128.84	132.80	155.38	14.25
	GMMreg	88.73	81.48	86.99	97.14	85.79	91.39	80.82	90.86	94.21	67.70
	POSE <sup>-</sup>	7.73	6.25	5.18	5.26	6.54	4.49	7.03	6.92	5.91	5.39
Translation Error(mm)	SA	<b>1.55</b>	<b>3.32</b>	7.43	<b>2.53</b>	2.98	<b>3.48</b>	<b>2.15</b>	<b>3.62</b>	<b>2.99</b>	<b>1.92</b>
	RANSAC	15.56	4.98	<b>4.42</b>	20.80	<b>6.95</b>	23.76	3.92	19.67	19.07	<b>10.41</b>
	SoftPOSIT5	1.04E+4	8.64E+3	6.16E+3	1.28E+4	8.14E+3	9.23E+3	8.16E+3	8.40E+3	1.89E+4	2.33E+4
	GMMreg	3.79E+2	1.19E+3	6.45E+8	4.75E+6	8.73E+2	1.59E+3	5.00E+2	1.84E+3	2.01E+8	1.45E+2
	POSE <sup>-</sup>	26.19	20.02	29.04	32.53	26.52	27.14	19.32	31.18	24.43	25.73
Number of Points	POSE	17.14	12.75	13.69	<b>13.75</b>	10.29	<b>10.96</b>	12.90	12.12	<b>11.90</b>	10.64
	3D	104	80	84	74	147	87	128	79	63	114
	2D	815	534	907	533	880	586	706	892	292	632
	SA	604.75	345.86	724.61	276.81	1196.50	435.84	668.21	425.55	123.54	392.68
	RANSAC	282.26	242.70	224.56	223.42	238.89	210.07	217.17	237.26	224.62	259.82
Time(s)	SoftPOSIT5	54.35	26.31	45.41	25.94	99.56	34.29	64.41	44.56	12.26	51.34
	GMMreg	32.21	16.83	22.79	14.84	53.57	21.03	42.23	24.65	6.87	34.26
	POSE <sup>-</sup>	<b>7.57</b>	<b>2.16</b>	<b>1.65</b>	<b>1.53</b>	<b>13.82</b>	<b>2.38</b>	<b>19.16</b>	<b>2.31</b>	<b>1.65</b>	<b>3.99</b>
	POSE	37.87	18.73	27.36	13.82	67.40	19.52	54.73	27.23	8.15	26.85
	SA	1.51	1.56	0.78	1.17	2.42	1.53	2.45	4.99	1.34	1.32
2D TRE	RANSAC	<b>0.80</b>	<b>0.60</b>	<b>0.88</b>	<b>0.47</b>	<b>1.07</b>	<b>0.81</b>	<b>0.97</b>	<b>0.72</b>	<b>0.36</b>	<b>0.61</b>
	SoftPOSIT5	2.96	2.35	2.14	1.46	4.72	3.62	2.74	2.05	0.72	1.77
	GMMreg	4.36	5.09	6.14	6.31	6.75	6.18	4.42	6.41	6.17	7.69
	POSE	1.22	1.01	0.81	0.91	2.44	0.78	2.45	1.06	0.88	1.02
	Gold Standard	1.57	1.09	0.97	1.15	2.59	1.04	2.44	1.34	1.14	1.07

POSE<sup>-</sup> denotes POSE without local refinement.

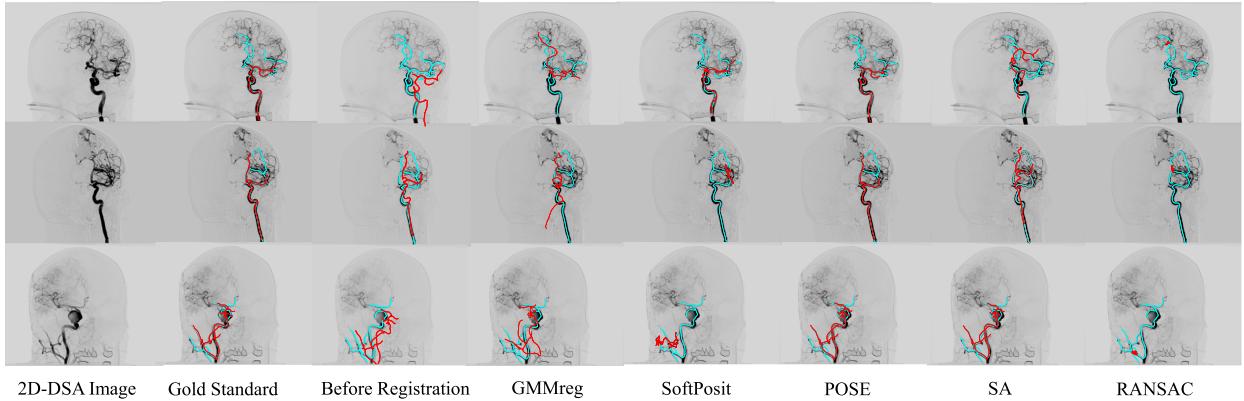


Fig. 16. Illustrative registration results of three cases. The original 2D DSA images are shown as in the first column, and the light blue are the 2D vessel point sets in the other columns. The red curves are the 3D vessel points projected by the gold standard transformation or registrations by each algorithm, respectively.

the synchronized multi-start rotation search, possible failures by using GMMReg alone are avoided. GOPAC failed to register the 3D and 2D points in this application.

## V. DISCUSSION

In this paper, we formulate the 2D-3D point set registration problem as finding the proper rotation and translation of

the 3D point set so that its projection can match the 2D point set, and we provide a global optimization approach for searching the rotation using the BnB framework. We found that the rotation search algorithm in SO(3) space can cover a certain region, and this functionality is used to produce an efficient algorithm to search the SE(3) space. We compared the proposed algorithms against state-of-the-art local and

global methods by using both synthetic and real data, and the proposed algorithms significantly outperformed the existing methods.

The global optimum is theoretically guaranteed in the rotation search in  $\text{SO}(3)$  space when using the proposed POSO algorithm. The geometry bound in  $\text{SO}(3)$  can be easily extended to  $\text{SE}(3)$  by considering the influence of translation on the size of the uncertainty ball in the 3D projection coordinate system; thus, theoretically, a guaranteed global optimum can be obtained by searching the translation and rotation jointly or in a nested way as shown in many previous studies. However, this method of searching the translation space is not efficient. In this paper, we showed that the rotation search algorithm can cover a region in the translation space, and the translation space can be searched block by block to find the optimal solution in  $\text{SE}(3)$ . In our experiment for the proposed  $\text{SE}(3)$  search algorithm, global optimality was obtained in all cases.

The efficiency of the proposed algorithm comes from two factors. First, we propose a circular geometry bound for the 3D points' positions on the projection plane so that the distance to 2D points can be calculated very quickly. Therefore, the pure rotation search algorithm, POSO, is fairly fast. Second, we proposed an efficient method of searching the translation space in the POSE algorithm. Instead of searching the  $\text{SE}(3)$  space by searching the rotation space and the translation space jointly or in a nested way, we search the translation space block by block, and the translation search is further accelerated by searching these blocks in a synchronized way. This synchronized searching scheme is based on the fact that at the position near the true position of the 3D points, the rotation search runs much faster than at positions far from the true position. In our synchronized searching scheme, the results of the faster rotation search at the true position can help terminate the slower rotation searches at positions far from the true position, thus avoiding wasteful runtime. The results in Fig. 12 (a) show that the synchronized searching scheme is one or two magnitudes faster than the naive one-by-one block search.

Further acceleration of our algorithm is possible. The current POSO algorithm uses the KD tree to search for inliers on the 2D projection plane, and other data structures, such as the matching list in [25], can be utilized to further improve the search speed. In addition, the whole POSO algorithm needs simple calculations and can be implemented on a GPU.

## VI. CONCLUSIONS

In this paper, we propose a novel method of efficiently performing 2D-3D point set registration. We apply the BnB method to obtain the global and deterministic optimality in  $\text{SO}(3)$  and then propose a synchronized searching schema in the translation space to achieve efficient 6DOF registration in the  $\text{SE}(3)$  space. The experiments showed that the proposed algorithm significantly outperformed state-of-the-art local and global methods.

## ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their valuable comments in revising this manuscript.

## REFERENCES

- [1] D. Campbell, L. Petersson, L. Kneip, and H. Li, "Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1–10.
- [2] M. Brown, D. Windridge, and J.-Y. Guillemaut, "Globally optimal 2D-3D registration from points or lines without correspondences," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2111–2119.
- [3] H. Zhou, T. Zhang, and W. Lu, "Vision-based pose estimation from points with unknown correspondences," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3468–3477, Aug. 2014.
- [4] P. David, D. Dementhon, R. Duraiswami, and H. Samet, "SoftPOSIT: Simultaneous pose and correspondence determination," *Int. J. Comput. Vis.*, vol. 59, no. 3, pp. 259–284, Sep. 2004.
- [5] Y. Khoo and A. Kapoor, "Non-iterative rigid 2D/3D point-set registration using semidefinite programming," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 2956–2970, Jul. 2016.
- [6] N. Baka, C. T. Metz, C. J. Schultz, R.-J. van Geuns, W. J. Niessen, and T. van Walsum, "Oriented Gaussian mixture models for nonrigid 2D/3D coronary artery registration," *IEEE Trans. Med. Imag.*, vol. 33, no. 5, pp. 1023–1034, May 2014.
- [7] U. Mitrović, Z. Špiclin, B. Likar, and F. Pernuš, "3D-2D registration of cerebral angiograms: A method and evaluation on clinical images," *IEEE Trans. Med. Imag.*, vol. 32, no. 8, pp. 1550–1563, Aug. 2013.
- [8] P. Markelj, D. Tomažević, B. Likar, and F. Pernuš, "A review of 3D/2D registration methods for image-guided interventions," *Med. Image Anal.*, vol. 16, no. 3, pp. 642–661, Apr. 2012.
- [9] L. Kneip, H. Li, and Y. Seo, "UPnP: An optimal  $O(n)$  solution to the absolute pose problem with universal applicability," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 127–142.
- [10] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (DLS) method for PnP," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 383–390.
- [11] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate  $O(n)$  solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009.
- [12] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
- [13] O. Enqvist, E. Ask, F. Kahl, and K. Åström, "Tractable algorithms for robust model estimation," *Int. J. Comput. Vis.*, vol. 112, no. 1, pp. 115–129, Mar. 2015.
- [14] E. Ask, O. Enqvist, and F. Kahl, "Optimal geometric fitting under the truncated L2-norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Portland, OR, USA, Jun. 2013, pp. 1722–1729.
- [15] O. Enqvist, E. Ask, F. Kahl, and K. Åström, "Robust fitting for multiple view geometry," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Florence, Italy, 2012, pp. 738–751.
- [16] T.-J. Chin, Y. H. Kee, A. Eriksson, and F. Neumann, "Guaranteed outlier removal with mixed integer linear programs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 5858–5866.
- [17] C. Olsson, A. Eriksson, and R. Hartley, "Outlier removal using duality," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1450–1457.
- [18] H. Li and R. Hartley, "The 3D-3D registration problem revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.
- [19] T. Benseghir, G. Malandain, and R. Vaillant, "Iterative closest curve: A framework for curvilinear structure registration application to 2D/3D coronary arteries registration," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, vol. 16, 2013, pp. 179–186.
- [20] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 2002.
- [21] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.
- [22] F. Moreno-Noguer, V. Lepetit, and P. Fua, "Pose priors for simultaneously solving alignment and correspondence," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 405–418.
- [23] J. Xia, X. Xu, and J. Xiong, "Simultaneous pose and correspondence determination using differential evolution," in *Proc. Int. Conf. Nat. Comput. (ICNC)*, May 2012, pp. 703–707.
- [24] H. Yang, F. Wang, Z. Li, and H. Dong, "Simultaneous pose and correspondence estimation based on genetic algorithm," *Int. J. Distrib. Sens. Netw.*, vol. 11, no. 11, p. 828241, Jan. 2015.

- [25] A. P. Bustos, T. J. Chin, A. Eriksson, H. Li, and D. Suter, “Fast rotation search with stereographic projections for 3D registration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2227–2240, Nov. 2016.
- [26] J. Yang, H. Li, D. Campbell, and Y. Jia, “Go-ICP: A globally optimal solution to 3D ICP point-set registration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, Nov. 2016.
- [27] D. Campbell and L. Petersson, “GOGMA: Globally-optimal Gaussian mixture alignment,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5685–5694.
- [28] J.-C. Bazin, Y. Seo, and M. Pollefeys, “Globally optimal consensus set maximization through rotation search,” in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2012, pp. 539–551.
- [29] R. I. Hartley and F. Kahl, “Global optimization through rotation space search,” *Int. J. Comput. Vis.*, vol. 82, no. 1, pp. 64–79, Apr. 2009.
- [30] R. Clackdoyle and C. Mennessier, “Centers and centroids of the cone-beam projection of a ball,” *Phys. Med. Biol.*, vol. 56, no. 23, pp. 7371–7391, Dec. 2011.
- [31] H. Zhou, Y. Yuan, and C. Shi, “Object tracking using SIFT features and mean shift,” *Comput. Vis. Image Understand.*, vol. 113, no. 3, pp. 345–352, Mar. 2009.
- [32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [33] D. Cox, J. Little, and D. O’Shea, *Projective Algebraic Geometry*. Berlin, Germany: Springer, 1997, pp. 349–428.
- [34] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, “New algorithms for 2D and 3D point matching: Pose estimation and correspondence,” *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.
- [35] S. Choi, T. Kim, and W. Yu, “Performance evaluation of RANSAC family,” in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2009, pp. 1–12.
- [36] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, “A comprehensive performance evaluation of 3D local feature descriptors,” *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, Jun. 2016.
- [37] S. Matl, R. Brosig, M. Baust, N. Navab, and S. Demirci, “Vascular image registration techniques: A living review,” *Med. Image Anal.*, vol. 35, pp. 1–17, Jan. 2017.
- [38] U. Mitrović, F. Pernuš, B. Likar, and Ž. Špiclin, “Simultaneous 3D–2D image registration and C-arm calibration: Application to endovascular image-guided interventions,” *Med. Phys.*, vol. 42, no. 11, pp. 6433–6447, Nov. 2015.
- [39] J. Kim, J. Lee, J. W. Chung, and Y.-G. Shin, “Locally adaptive 2D–3D registration using vascular structure model for liver catheterization,” *Comput. Biol. Med.*, vol. 70, pp. 119–130, Mar. 2016.
- [40] M. Groher, T. Jakobs, N. Paday, and N. Navab, “Planning and intraoperative visualization of liver catheterizations: New CTA protocol and 2D–3D registration method,” *Acad. Radiol.*, vol. 14, no. 11, pp. 1324–1339, Nov. 2007.
- [41] J. A. Bærentzen, “On the implementation of fast marching methods for 3D lattices,” Dept. Inform. Math. Model., Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep. IMM-REP-2001-13, 2001.
- [42] M. S. Hassouna and A. A. Farag, “Multistencils fast marching methods: A highly accurate solution to the Eikonal equation on Cartesian domains,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1563–1574, Sep. 2007.
- [43] R. Van Uitert and I. Bitter, “Subvoxel precise skeletons of volumetric data based on fast marching methods,” *Med. Phys.*, vol. 34, no. 2, pp. 627–638, Feb. 2007.
- [44] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 1998, pp. 130–137.
- [45] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, “Complete solution classification for the perspective-three-point problem,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003.



**Yinlong Liu** received the M.S. degree from the Digital Medical Research Center, School of Basic Medical Science, Fudan University, in 2018. He is currently pursuing the Ph.D. degree with the Department of Informatics, Technical University of Munich, Munich, Germany. His research interests include image guided surgery and computer vision, especially on optimization methods in geometric vision.



**Yuan Dong** received the B.E. degree from Chongqing Medical University, Chongqing, China. He is currently pursuing the Ph.D. degree in biomedical engineering with the School of Basic Medical Science, Fudan University, Shanghai, China. His research interest includes registration, image guided surgery, and computer vision.



**Zhijian Song** received the B.S. degree from the Shandong University of Technology, Shandong, China, in 1982, the M.S. degree from the Jiangsu University of Technology, Jiangsu, China, in 1991, and the Ph.D. degree in biomedical engineering from Xi’an Jiaotong University, Xi’an, China, in 1994. He is currently a Distinguished Professor of biomedical engineering with the School of Basic Medical Science, Fudan University, Shanghai, China, where he is also the Director of the Digital Medical Research Center and the Shanghai Key Laboratory of Medical Imaging Computing and Computer Assisted Intervention. His research interests include medical image processing, image-guided intervention, and the application of virtual and augmented reality technologies in medicine.



**Manning Wang** received the B.S. and M.S. degrees in power electronics from Shanghai Jiao Tong University, Shanghai, China, in 1999 and 2002, respectively, and the Ph.D. degree in biomedical engineering from Fudan University, Shanghai, China, in 2011. He is currently a Professor of biomedical engineering with the School of Basic Medical Science, Fudan University. His research interests include computer vision, medical image processing, and computer-assisted surgery.