Efficient Global Point Cloud Alignment using Bayesian Nonparametric Mixtures

Julian Straub*   Trevor Campbell*   Jonathan P. How   John W. Fisher III
Massachusetts Institute of Technology

Abstract

Point cloud alignment is a common problem in computer vision and robotics, with applications ranging from 3D object recognition to reconstruction. We propose a novel approach to the alignment problem that utilizes Bayesian nonparametrics to describe the point cloud and surface normal densities, and branch and bound (BB) optimization to recover the relative transformation. BB uses a novel, refinable, near-uniform tessellation of rotation space using 4D tetrahedra, leading to more efficient optimization compared to the common axis-angle tessellation. We provide objective function bounds for pruning given the proposed tessellation, and prove that BB converges to the optimum of the cost function along with providing its computational complexity. Finally, we empirically demonstrate the efficiency of the proposed approach as well as its robustness to real-world conditions such as missing data and partial overlap.

1. Introduction

Point cloud alignment is a fundamental problem for many applications in robotics [35, 23] and computer vision [45, 40, 54]. Finding the global transformation is generally hard: point-to-point correspondences typically do not exist, the point clouds might only have partial overlap, and the underlying objects themselves are often nonconvex, leading to a potentially large number of alignment local minima. As such, popular local optimization techniques suffice only in circumstances with small true relative transformations and large overlap, such as in dense 3D incremental mapping [23, 40, 54]. Solving the alignment problem for large unknown relative transformations and small point cloud overlap calls for a global approach. Example applications are the loop-closure problem in SLAM [8] and the model-based detection of objects in 3D scenes [29].

Motivated by the observation that surface normal distributions are translation invariant [25] and straightforward to compute [39, 46], we develop a two-stage branch and bound (BB) [31, 32] optimization algorithm for point cloud alignment. We model the surface normal distribution of
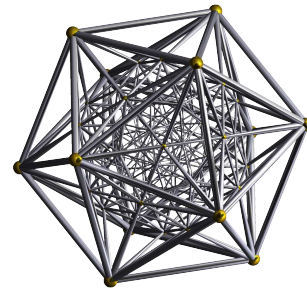

Figure 1: A 3D projection of the 600-cell [57]—a 4D object tessellating the space of rotations for the proposed branch and bound approach to point cloud alignment.

each point cloud as a Dirichlet process (DP) [18, 50] von-Mises-Fisher (vMF) [20] mixture [47] (DP-vMF-MM). To find the optimal rotation, we minimize the L^2 distance between the distributions over the space of 3D rotations. We develop a novel refinable tessellation consisting of 4D tetrahedra (see Fig. 1) which more uniformly approximates rotation space and is more efficient than the common axis-angle tessellation [33, 22] during BB optimization. Given the optimal rotation and modeling the two point distributions as DP Gaussian mixtures [3, 11] (DP-GMM), we obtain the optimal translation similarly via BB over the space of 3D translations. The use of mixture models circumvents discretization artifacts, while still permitting efficient optimization. In addition to algorithmic developments, we provide corresponding theoretical bounds on the convergence of both BB stages, linking the quality of the derived rotation and translation estimates to the depth of the search tree and thus the computation time of the algorithm. Experiments on real data corroborate the theory, and demonstrate the accuracy and efficiency of BB as well as its robustness to real-world conditions, such as partial overlap, high noise, and large relative transformations.

2. Related Work

Local Methods   There exists a variety of approaches for local point cloud alignment [10, 45]. Iterative closest point (ICP) [6], the most common of these, alternates between as-

---

*The first two authors contributed equally to this work.

sociating the points in both clouds and updating the relative transformation estimate under those associations. There are many variants of ICP [43] differing in their choice of cost function, how correspondences are established, and how the objective is optimized at each iteration. An alternative developed by Magnusson et al. [35] relies on the normal distribution transform (NDT) [7], which represents the density of the scans as a structured GMM. This approach has been shown to be more robust than ICP in certain cases [36]. Approaches that use correlation of kernel density estimates (KDE) for alignment [51] or GMMs [28] use a similar representation as the proposed approach. KDE-based methods scale poorly with the number of points. In contrast, we use mixture models inferred by nonparametric clustering algorithms (DP-means [30] and DP-vMF-means [47]). This allows adaptive compression of the data, enabling the processing of large noisy point clouds (see Sec. 6 for experiments with more than 300k points). Straub et al. propose two local rotational alignment algorithms [47, 48] that, similarly to the proposed approach, utilize surface normal distributions modeled as vMF mixtures. Common to local methods is the assumption of an initialization close to the true transformation and significant overlap between the two point clouds. If either of these assumptions are violated, local methods become unreliable as they tend to get stuck in suboptimal local minima [43, 45, 36].

Global Methods   Global point cloud alignment algorithms make no prior assumptions about the relative transformation or amount of overlap. For those reasons global algorithms, such as the proposed one, are often used to initialize local methods. 3D-surface-feature-based algorithms [44, 21, 29, 1] involve extracting local features, obtaining matches between features in the two point clouds, and finally estimating the relative pose using RANSAC [19] or other robust estimators [26]. These algorithms can be vulnerable to large fractions of incorrect feature matches and repetitive scene elements, though recent work is beginning to address the former [2, 56]. A second class of approaches, including the proposed approach, rely on statistical properties of the two point clouds. Makadia et al. [37] separate rotational and translational alignment. Rotation is obtained by maximizing the convolution of the peaks of the extended Gaussian images (EGI) [25] of the two surface normal sets. This search is performed using the spherical Fourier Transform [17]. After rotational alignment, the translation is found similarly via the fast Fourier Transform. For the alignment of 2D scans, Weiss et al. [53] and Bosse et al. [8] follow a similar convolution-based approach. Early work by Li, Hartley and Kahl [33, 22] on BB for point cloud alignment used the axis-angle (AA) representation of rotations. A drawback of this approach is that a uniform AA tessellation does not lead to a uniform tessellation in rotation space (see Sec. 4.1). As we show in

Sec. 6, this leads to less efficient BB search. Parra Bustos et al. [41] propose improved bounds for rotational alignment by reasoning carefully about the geometry of the AA tessellation. GoICP [55] nests BB over translations inside BB over rotations and utilizes ICP internally to improve the BB bounds. GOGMA [9] uses a similar approach, but replaces the objective with a convolution of GMMs. Both GoICP and GOGMA involve BB over the joint 6-dimensional rotation and translation space; since the complexity of BB is exponential in the dimension, these methods are relatively computationally expensive (see results Fig. 10).

3. The Point Cloud Alignment Problem

Our approach to point cloud alignment relies on the fact that surface normal distributions are invariant to translation [25] and easily computed [39, 46], allowing us to isolate the effects of rotation. Thus we decompose the task of finding the relative transformation into first finding the rotation using only the surface normal distribution, and then obtaining the translation given the optimal rotation.

Let a noisy sampling of a surface S be described by the joint point and surface normal density $p(x, n)$, where $x \in \mathbb{R}^3$ and $n \in \mathbb{S}^2$. A sensor observes two independent samples from this model: one from $p_1(x, n) = p(R^\star x + t^\star, R^\star n)$, and one from $p_2(x, n) = p(x, n)$ differing in an unknown rotation $R^\star \in \mathrm{SO}(3)$ and translation $t^\star \in \mathbb{R}^3$. Given these samples, we model the marginal point densities $\hat{p}_1(x)$, $\hat{p}_2(x)$ using the posterior of a Dirichlet process Gaussian mixture (DP-GMM) [3], and model the marginal surface normal densities $\hat{p}_1(n)$, $\hat{p}_2(n)$ using the posterior of a Dirichlet process von Mises-Fisher mixture (DP-vMF-MM) [5, 47]. Note that the formulation using DP mixture models admits arbitrarily accurate estimates of a large class of noisy surface densities (Theorem 2.2 in [15]). Given the density estimates, we formulate the problem of finding the relative transformation as

$$\hat{q} = \arg\max_{q \in \mathbb{S}^3} \int_{\mathbb{S}^2} \hat{p}_1(n)\hat{p}_2(q \circ n)\mathrm{d}n$$
$$\hat{t} = \arg\max_{t \in \mathbb{R}^3} \int_{\mathbb{R}^3} \hat{p}_1(x)\hat{p}_2(\hat{q} \circ x + t)\mathrm{d}x, \tag{1}$$

where we represent rotations using unit quaternions in $\mathbb{S}^3$, the 4D sphere [24], and where $q \circ n$ denotes the rotation of a surface normal n by a unit quaternion q. Eq. (1) minimizes the L_2 metric via maximization of the convolution, which has been shown to be robust in practice [28]. This is a common approach for Gaussian MMs [51, 28, 9] but to our knowledge has not been explored for vMF-MMs, nor for Bayesian nonparametric DP mixtures. In fact, the use of DP mixtures is critical, as it allows the automatic selection of a parsimonious, but accurate, representation of the point cloud data. This improves upon both kernel den-

sity estimates [51], which are highly flexible but make optimizing Eq. (1) intractable for large RGB-D datasets, and fixed-sized GMMs [28, 9], which require heuristic model selection and may not be rich enough to capture complex scene geometry. While exact posterior predictive DP-MM densities cannot be computed tractably, excellent estimation algorithms are available, which we use in this work [30, 47].

Both optimization problems in Eq. (1) are nonconcave maximizations. Considering the geometry of the problem, we expect many local maxima, rendering typical gradient-based methods ineffective. This motivates the use of a global approach. We develop a two-step BB procedure [31, 32] that first searches over $\mathbb{S}^3$ for the optimal rotation $\hat{q}$, and then over $\mathbb{R}^3$ for the optimal translation $\hat{t}$. As BB may return multiple optimal rotations (e.g. if the scene has rotational symmetry) we estimate the optimal translation under each of those rotations, and return the joint transformation with the highest translational cost lower bound. Note that while $\hat{q}, \hat{t}$ is not necessarily the optimal transformation under rotation and translation *jointly*, the decoupling of rotation and translation we propose reduces the computational complexity of BB significantly. This is because the complexity scales exponentially in the search space dimension; optimizing over two 3D spaces ($\mathbb{R}^3$ and $\mathbb{S}^3$) separately is significantly less costly than over the joint 6D space.

BB requires three major components: (1) a tessellation method for covering the optimization domain with subsets (see Sec. 4.1 and 5.1); (2) a branch/refinement procedure for subdividing any subset into smaller subsets (see Sec. 4.1 and 5.1); and (3) upper and lower bounds of the maximum objective on each subset to be used for pruning (see Sec. 4.2 and 5.2). BB proceeds by bounding the optimal objective in each subset, pruning those which cannot contain the maximum, subdividing the best subset to refine the bounds, and iterating. Note that in this work we select the node with the highest upper bound for subdivision. More nuanced strategies have been developed and could also be utilized [27, 32].

4. vMF Mixture Rotational Alignment

We model the distributions of surface normals n as von-Mises-Fisher [20] mixture models (vMF-MM) with means $\{\mu_{ik}\}_{k=1}^{K_i}$, concentrations $\{\tau_{ik}\}_{k=1}^{K_i}$, and positive weights $\{\pi_{ik}\}_{k=1}^{K_i}$, $\sum_{k=1}^{K_i} \pi_{ik} = 1$, for $i \in \{1, 2\}$, with density

$$\hat{p}_i(n) = \sum_{k=1}^{K_i} \pi_{ik} C_{ik} e^{\tau_{ik}\mu_{ik}^T n} \quad C_{ik} \triangleq \frac{\tau_{ik}}{4\pi \sinh(\tau_{ik})}. \quad (2)$$

While there are many techniques for inferring vMF-MMs [4, 16, 47], we use a nonparametric method [47] that infers an appropriate K_i automatically. The rotational alignment problem from Eq. (1) with this model becomes

$$\max_{q \in \mathbb{S}^3} \sum_{k, k'} \frac{D_{kk'}}{2\pi} \int_{\mathbb{S}^2} e^{(\tau_{1k}\mu_{1k} + \tau_{2k'} q \circ \mu_{2k'})^T n} \, dn$$

$$D_{kk'} \triangleq (2\pi)\pi_{1k}\pi_{2k'}C_{1k}C_{2k'}. \quad (3)$$
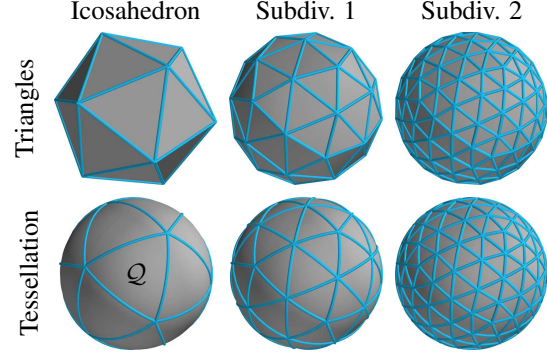
(2a) Tessellation of $\mathbb{S}^2$ via iterated triangle subdivision. The tessellation of $\mathbb{S}^3$ follows the same principles, but with 4D tetrahedra instead of 3D triangles. Note the uniformity of the tessellation.
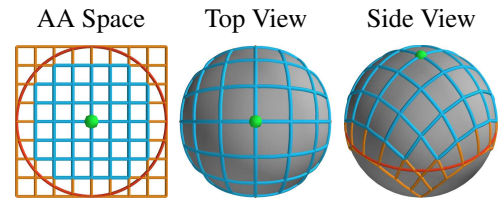
(2b) Tessellation of $\mathbb{S}^2$ via uniform tessellation in the axis-angle (AA) space. The axis-angle tessellation of $\mathbb{S}^3$ follows the same principle and incurs similar distortion. Note that orange tiles contain surface area on the lower half-sphere, so parts of the rotation space are covered twice, making BB inefficient.

We obtain the following objective function by noting that the integral is the normalization constant of a vMF density with concentration $z_{kk'}(q) \triangleq \|\tau_{1k}\mu_{1k} + \tau_{2k'} q \circ \mu_{2k'}\|$:

$$\max_{q \in \mathbb{S}^3} \sum_{k, k'} D_{kk'} f(z_{kk'}(q))$$

$$\text{where } f(z) \triangleq 2\sinh(z)z^{-1} = (e^z - e^{-z}) z^{-1}. \quad (4)$$

4.1. Cover and Refinement of the Rotation Space $\mathbb{S}^3$

In this section, we develop a novel tessellation scheme for the space of rotations, and show how to refine it in a way that guarantees convergence of BB for rotational alignment. We follow a similar approach to the geodesic grid tessellation of a sphere in 3D (i.e. $\mathbb{S}^2$): as depicted in Fig. 2a, starting from an icosahedron, each of the 20 triangular faces is subdivided into four triangles of equal size. Then the newly created triangle corners are normalized to unit length, projecting them onto the unit sphere.

In four dimensions we instead start with the analogue of the icosahedron, the *600-cell* [13] (shown in Fig. 1), an object composed of 600 4D tetrahedra. We first generate its 120 vertices with the following algorithm [13, pp. 402–403]. Let $\phi = \frac{1}{2}(1 + \sqrt{5})$. Then the (unnormalized) 120 vertices of the 600-cell in 4D are

- even permutations of $\left[\pm\phi, \pm 1, \pm\phi^{-1}, 0\right]^T$ (96 vertices),
- all permutations of $\left[\pm 2, 0, 0, 0\right]^T$ (8 vertices), and
- all permutations of $\left[\pm 1, \pm 1, \pm 1, \pm 1\right]^T$ (16 vertices).

We then scale the 120 vertices to each have unit norm, representing a <u>3D quaternion rotation</u>. Next, noting that the angle between any two connected tetrahedra vertices is $36°$, we iterate over all $\binom{120}{4}$ possible choices of 4 vertices, and only select those 600 tetrahedra for which all pairwise angles are $36°$. This collection of tetrahedra, which are "flat" in 4D analogous to triangles in 3D, comprises a 4D object which approximates the 4D sphere, $\mathbb{S}^3$. Then, since the set of all quaternion rotations may be represented by any hemisphere of $\mathbb{S}^3$ (q and $-q$ describe the same rotation), we define the "north" vector to be $[0, 0, 0, 1]^T \in \mathbb{S}^3$, and only keep those tetrahedra for which at least one vertex has angle $< 90°$ to the north vector. This results in 330 tetrahedra that approximate the 4D upper hemisphere in $\mathbb{S}^3$, i.e. the space of quaternion rotations. Note that this construction procedure is the same for *any* optimization on $\mathbb{S}^3$, so it can be performed once and the result may be stored for efficiency.

One major advantage of the proposed $\mathbb{S}^3$ tessellation is that it is exactly uniform at the 0th level and approximately uniform for deeper subdivision levels (Fig. 2a shows the analogous near-uniformity for $\mathbb{S}^2$). <u>This generally tightens bounds employed by BB, leading to more efficient optimization.</u> Another advantage is that this tessellation is a near-exact covering of the upper hemisphere of $\mathbb{S}^3$. Only 7% of rotation space is covered twice, meaning that BB wastes little time with duplicate searching. The widely employed AA-tessellation scheme [33, 22, 41, 55], in contrast, uniformly tessellates a cube enclosing the axis-angle space, <u>a 3D sphere with radius π, and maps that tessellation onto the rotation space.</u> There are two major issues with the AA approach. <u>First, it covers 46% of rotation space twice [33, 22] (see Fig. 2b). Second, it does not lead to uniform tessellation in rotation space.</u> The reason for this is that the <u>Euclidean metric in AA space is a poor approximation of the distance on the rotation manifold</u> [33]. Fig. 2b shows the AA tessellation analog for $\mathbb{S}^2$, highlighting its significant non-uniformity. We empirically find that the $\mathbb{S}^3$ tessellation leads to more efficient BB optimization than the AA tessellation (see results in Figs. 6 and 7).

We now discuss two properties of the proposed tessellation required by BB: 1) that it is a *cover* for the upper hemisphere of $\mathbb{S}^3$, guaranteeing that BB will search the whole space of rotations; and 2) that it is *refinable*, so BB can search promising subsets in increasingly more detail.

Cover Let the four vertices of a single tetrahedron from our approximation of $\mathbb{S}^3$ be denoted $q_j \in \mathbb{S}^3$, $j \in \{1, \ldots, 4\}$. Then, stacking them horizontally into a matrix $Q \in \mathbb{R}^{4 \times 4}$, the projection $\mathcal{Q}$ of the tetrahedron onto $\mathbb{S}^3$ is:

$$\mathcal{Q} = \left\{ q \in \mathbb{R}^4 : \|q\| = 1, \ q = Q\alpha, \ \alpha \in \mathbb{R}_+^4 \right\}. \quad (5)$$
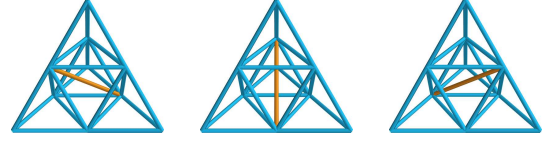
(3a) The three subdivision patterns of a tetrahedron displayed in 3D. The internal orange edge is chosen to minimize distortion.
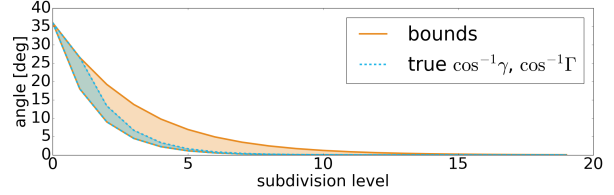
(3b) The bounds in Eq. (8) compared to the true min & max angles between tetrahedron vertices for increasing refinement level.

In other words, $\mathcal{Q}$ is the set of unit quaternions found by extending the (flat in 4D) tetrahedron to the unit sphere using rays from the origin. For $\mathbb{S}^2$, this is displayed in the second row of Fig. 2a. The proposed set of 330 projected tetrahedra $\mathcal{Q}$ forms a cover of the upper hemisphere of $\mathbb{S}^3$.

Refinement Next, we require a method of subdividing any $\mathcal{Q}$ in the cover. Similar to the triangle subdivision method for refining the tessellation of $\mathbb{S}^2$, each 4D tetrahedron can be subdivided into eight smaller tetrahedra [34] as depicted in Fig. 3a. The resulting six new vertices for the subdivided tetrahedra are scaled to unit length. As we have the freedom to choose one of three internal edges for subdivision, we choose the internal edge with the minimum angle between its unit-norm vertices. In other words, denoting ξ_k for $k \in \{1, 2, 3\}$ to be the three internal dot products,

$$k^\star = \operatorname*{arg\,max}_{k \in \{1,2,3\}} \ \xi_k. \quad (6)$$

This process forms the eight new subdivided cover elements $\mathcal{Q}$. For example, if q_i, $i \in \{1, \ldots, 4\}$ are the vertices of $\mathcal{Q}$, then one of the subdivisions (corresponding to one of the "corner" subtetrahedra in Fig. 3a) of $\mathcal{Q}$ would have vertices

$$q_1, \quad \frac{q_1 + q_2}{\|q_1 + q_2\|}, \quad \frac{q_1 + q_3}{\|q_1 + q_3\|}, \quad \text{and} \quad \frac{q_1 + q_4}{\|q_1 + q_4\|}. \quad (7)$$

Selecting the internal edge via Eq. (6) is critical to our BB convergence guarantee in Sec. 4.4. If Eq. (6) is not used, the individual subsets $\mathcal{Q}$ can become highly skewed due to repeated distortion from the unit-norm projection of the vertices, and refining $\mathcal{Q}$ does not necessarily correspond to shrinking the angular range of rotations it captures. Since we use Eq. (6), however, Lemma 1 guarantees that subdividing $\mathcal{Q}$ shrinks its set of rotations appropriately:

Lemma 1 *Let γ_N be the min dot product between vertices of any one $\mathcal{Q}$ at refinement level N. Then*
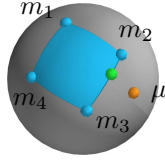
$$\frac{2\gamma_{N-1}}{1+\gamma_{N-1}} \leq \gamma_N, \qquad \text{where} \qquad \gamma_0 \triangleq \cos 36°. \quad (8)$$
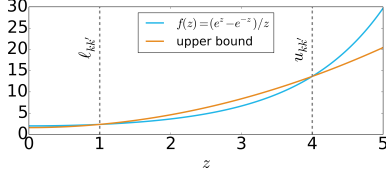
(4a) The function $f(z)$ and its quadratic upper bound, valid for $z \in [\ell_{kk'}, u_{kk'}]$ (here, $\ell_{kk'} = 1$ and $u_{kk'} = 4$).

(4b) Closest point (green) from a point μ (orange, Eq. 16).

This result (proof in the supplement) shows that the tetrahedra shrink and allow BB to improve its bounds during subdivision. Figure 3b demonstrates the tightness of this bound, showing that $\cos^{-1} \gamma_N$ converges to 0 as $N \to \infty$. We conjecture that the *max* dot product Γ_N satisfies a similar recursion, $\Gamma_N \leq \sqrt{(1 + \Gamma_{N-1})/2}$, although this is not required for our convergence analysis. Fig. 3b shows empirically that this matches the true max dot product, but we leave the proof as an open problem.

4.2. vMF Mixture Model Bounds

BB requires both upper and lower bounds on the maximum of the objective function within each projected tetrahedron $\mathcal{Q}$, i.e. we need L and U such that

$$L \leq \max_{q \in \mathcal{Q}} \sum_{k,k'} D_{kk'} f(z_{kk'}(q)) \leq U. \qquad (9)$$

For the lower bound L, one can evaluate the objective at *any* point in $\mathcal{Q}$ (e.g. its center). For the upper bound U, we use a quadratic upper bound on $f(z)$ (see Fig. 4a and the supplement for details), noting that $\ell_{kk'} \leq z_{kk'}(q) \leq u_{kk'}$ for all $q \in \mathcal{Q}$, where

$$\ell_{kk'} \triangleq \min_{q \in \mathcal{Q}} z_{kk'}(q) \quad \text{and} \quad u_{kk'} \triangleq \max_{q \in \mathcal{Q}} z_{kk'}(q), \qquad (10)$$

whose computation is discussed in Sec. 4.3. This results in the upper bound U where

$$U = \max_{q \in \mathcal{Q}} q^T A q + B$$
$$A \triangleq \sum_{k,k'} 2 D_{kk'} \tau_{1k} \tau_{2k'} g_{kk'} \Xi_{kk'}$$
$$B \triangleq \sum_{k,k'} D_{kk'} \left((\tau_{1k}^2 + \tau_{2k'}^2) g_{kk'} + h_{kk'} \right) \qquad (11)$$
$$g_{kk'} \triangleq \frac{f(u_{kk'}) - f(\ell_{kk'})}{u_{kk'}^2 - \ell_{kk'}^2}$$
$$h_{kk'} \triangleq \frac{u_{kk'}^2 f(\ell_{kk'}) - \ell_{kk'}^2 f(u_{kk'})}{u_{kk'}^2 - \ell_{kk'}^2},$$

and $\Xi_{kk'} \in \mathbb{R}^{4 \times 4}$ is defined as the matrix for which $\mu_{1k}^T (q \circ \mu_{2k'}) = q^T \Xi_{kk'} q$ for any quaternion q (see the supplement for details). Writing $q = Q\alpha$ as a linear combination of vertices of $\mathcal{Q}$ as in Eq. (5),

$$U = \max_{\alpha \in \mathbb{R}^4} \alpha^T Q^T A Q \alpha + B$$
$$\text{s.t.} \quad \alpha^T Q^T Q \alpha = 1, \ \alpha \geq 0. \qquad (12)$$

Since $\alpha \in \mathbb{R}^4$, and we have the constraint $\alpha \geq 0$, we can search over all $\sum_{i=1}^4 \binom{4}{i} = 15$ possible combinations of components of α being zero or nonzero. Thus we solve the optimization for $U_{\mathcal{I}}$ given each possible subset $\mathcal{I} \subseteq \{1, 2, 3, 4\}$ of nonzero components of α, and set

$$U = B + \max_{\mathcal{I} \subseteq \{1,2,3,4\}} U_{\mathcal{I}}. \qquad (13)$$

For $U_{\mathcal{I}}$, we use a Lagrange multiplier for the equality constraint in Eq. (12) and set the derivative to 0, yielding a small generalized eigenvalue problem of dimension $|\mathcal{I}| \leq 4$,

$$U_{\mathcal{I}} = \max \{ \lambda : \exists v \geq 0, \ (Q^T A Q)_{\mathcal{I}} \, v = \lambda (Q^T Q)_{\mathcal{I}} \, v \}, \qquad (14)$$

where v is a $|\mathcal{I}|$-dimensional vector, and subscript $\mathcal{I}$ denotes the submatrix with rows and columns selected from $\mathcal{I}$. The condition that all elements of v are nonnegative in Eq. (14) enforces that $\alpha \geq 0$ and thus α corresponds to a solution q that lies in $\mathcal{Q}$. Note that if v is an eigenvector, so is $-v$. If no v satisfies $v \geq 0$, then we define $U_{\mathcal{I}} = -\infty$.

4.3. Computing $\ell_{kk'}$ and $u_{kk'}$

To find the upper bound U in Eq. (12), we require the constants $\ell_{kk'}$ and $u_{kk'}$ for each pair of mixture components k, k'. Given their definitions in Eq. (10), we have

$$u_{kk'} = \sqrt{\tau_{1k}^2 + \tau_{2k'}^2 + 2\tau_{1k}\tau_{2k'} \max_{q \in \mathcal{Q}} \mu_{1k}^T (q \circ \mu_{2k'})},$$
$$\ell_{kk'} = \sqrt{\tau_{1k}^2 + \tau_{2k'}^2 - 2\tau_{1k}\tau_{2k'} \max_{q \in \mathcal{Q}} (-\mu_{1k})^T (q \circ \mu_{2k'})}. \qquad (15)$$

Since the inner optimization objective only depends on the rotation of $\mu_{2k'}$ by q, we can reformulate the optimization as being over the set of 3D vectors $v \in \mathbb{S}^2$ such that $v = q \circ \mu_{2k'}$ for some $q \in \mathcal{Q}$. Thus, finding $u_{kk'}$ and $\ell_{kk'}$ is equivalent to finding the closest and furthest unit vectors in 3D to μ_{1k} over the set of such vectors v, shown in Fig. 4b. To solve this problem, let the vertices of $\mathcal{Q}$ be q_i, $i \in \{1, \ldots, 4\}$, and define the matrix $M \triangleq [m_1, \ldots, m_4] \in \mathbb{R}^{3 \times 4}$ where $m_i \triangleq q_i \circ \mu_{2k'}$. The inner optimization in Eq. (15) can be written as (for $u_{kk'}$ set $\mu = \mu_{1k}$; for $\ell_{kk'}$ set $\mu = -\mu_{1k}$)

$$J = \max_{\alpha \in \mathbb{R}^4} \mu^T M \alpha$$
$$\text{s.t.} \quad \alpha^T M^T M \alpha = 1 \quad \alpha \geq 0. \qquad (16)$$

Showing that Eq. (16) is equivalent to solving the inner optimizations of Eq. (15) is quite technical and is deferred to the supplement. Again we search over all $\sum_{i=1}^3 \binom{4}{i} = 14$ possible combinations of components of α being zero or nonzero (we do not check the $i = 4$ case since in this case the matrix $M_{\mathcal{I}}$ below is rank-deficient). We thus solve the optimization for $J_{\mathcal{I}}$ given each subset $\mathcal{I} \subseteq \{1, \ldots, 4\}$, $|\mathcal{I}| \leq 3$ of nonzero components, and set

$$J = \max_{\mathcal{I} \subseteq \{1,2,3,4\} \text{ s.t. } |\mathcal{I}| \leq 3} J_{\mathcal{I}}. \qquad (17)$$

To solve for $J_\mathcal{I}$, we use a Lagrange multiplier for the equality constraint, and set derivatives to 0 to find that

$$J_\mathcal{I} = \sigma \sqrt{\mu^T M_\mathcal{I} \left(M_\mathcal{I}^T M_\mathcal{I}\right)^{-1} M_\mathcal{I}^T \mu} \qquad (18)$$

where

$$\sigma = \begin{cases} 1 & \left(M_\mathcal{I}^T M_\mathcal{I}\right)^{-1} M_\mathcal{I}^T \mu \geq 0 \\ -1 & \left(M_\mathcal{I}^T M_\mathcal{I}\right)^{-1} M_\mathcal{I}^T \mu \leq 0 \\ -\infty & \text{else}, \end{cases} \qquad (19)$$

and $M_\mathcal{I}$ is the matrix constructed from the set of columns in M corresponding to $\mathcal{I}$. Note that σ is also defined to be $\sigma = -\infty$ if $M_\mathcal{I}^T M_\mathcal{I}$ is not invertible. After solving for the value of J via Eq. (17), we substitute it back into Eq. (15) to obtain $u_{kk'}$ or $\ell_{kk'}$ as desired.

4.4. Convergence Properties

We have now developed all the components necessary to optimize Eq. (4) via BB on $\mathbb{S}^3$. Theorem 1 (proof in the supplement) provides a bound on the worst-case search tree depth N to guarantee BB terminates with rotational precision of ϵ degrees, along with the overall computational complexity. Note that the complexity of BB is exponential in N, but since N is logarithmic in ϵ^{-2} (by Theorem 1, Eq. (20) and $\cos x \simeq 1 - x^2$ for $x \ll 1$), the complexity of BB is polynomial in ϵ^{-1}. Recall from Sec. 4.1 that γ_0 for the 600-cell is $\gamma_0 \triangleq \cos 36°$.

Theorem 1 *Suppose γ_0 is the initial maximum angle between vertices in the tetrahedra tessellation of $\mathbb{S}^3$, and let*

$$N \triangleq \max \left\{0, \left\lceil \log_2 \frac{\gamma_0^{-1}-1}{\cos(\epsilon/2)^{-1}-1} \right\rceil \right\}. \qquad (20)$$

Then at most N refinements are required to achieve an angular tolerance of ϵ on $\mathbb{S}^2$, and BB has complexity $O(\epsilon^{-6})$.

5. Gaussian Mixture Translational Alignment

In this section, we reuse notation for simplicity and to highlight parallels between the translational and rotational alignment problems. We model the density of points in the two point clouds as Gaussian mixture models (GMMs) with means $\{\mu_{ik}\}_{k=1}^{K_i}$, covariances $\{\Sigma_{ik}\}_{k=1}^{K_i}$, and weights $\{\pi_{ik}\}_{k=1}^{K_i}$, $\sum_{k=1}^{K_i} \pi_{ik} = 1$, for $i \in \{1,2\}$, with density

$$\hat{p}_i(x) = \sum_{k=1}^{K_i} \pi_{ik} \mathcal{N}(x; \mu_{ik}, \Sigma_{ik}). \qquad (21)$$

GMMs can be inferred in a variety of ways [30, 11]. Let $R^\star \in \mathrm{SO}(3)$ be the optimal rotation corresponding to $q^\star$ recovered using BB over $\mathbb{S}^3$. Then defining

$$\begin{aligned} m_{kk'} &\triangleq R^\star \mu_{2k'} - \mu_{1k}, \\ S_{kk'} &\triangleq \Sigma_{1k} + R^\star \Sigma_{2k'} R^{\star T}, \\ z_{kk'}(t) &\triangleq -\tfrac{1}{2}\left(t - m_{kk'}\right)^T S_{kk'}^{-1}\left(t - m_{kk'}\right), \end{aligned} \qquad (22)$$
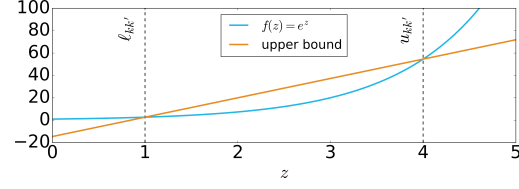

Figure 5: The function $f(z)$ and its linear upper bound, valid for $z \in [\ell_{kk'}, u_{kk'}]$ (here, $\ell_{kk'} = 1$ and $u_{kk'} = 4$).

the translational optimization in Eq. (1) becomes:

$$\begin{aligned} &\max_{t \in \mathbb{R}^3} \sum_{k,k'} D_{kk'} f(z_{kk'}(t)) \\ &\text{where } f(z) \triangleq e^z, \ D_{kk'} \triangleq \frac{\pi_{1k}\pi_{2k'}}{\sqrt{(2\pi)^3 |S_{kk'}|}}. \end{aligned} \qquad (23)$$

This is again a nonconcave maximization, motivating the use of a global approach. Thus, we develop a second BB procedure on $\mathbb{R}^3$ to find the optimal translation.

5.1. Cover and Refinement of $\mathbb{R}^3$

We tessellate the space of translations, $\mathbb{R}^3$ with rectangular cells. The initial tessellation is obtained by enclosing both point clouds with a single rectangular bounding box with diagonal length γ_0. For the refinement step, we choose to subdivide the cell into eight equal-sized rectangular cells. Thus, the minimum γ_N diagonal of the rectangular cells at refinement level N possesses a straightforward shrinkage property similar to Eq. (8),

$$\frac{\gamma_{N-1}}{2} = \gamma_N. \qquad (24)$$

5.2. Gaussian Mixture Model Bounds

As in the rotational problem, the translational BB algorithm requires lower and upper bounds on the objective function in Eq. (23):

$$L \leq \max_{t \in \mathcal{Q}} \sum_{k,k'} D_{kk'} f(z_{kk'}(t)) \leq U. \qquad (25)$$

For the lower bound L, one can evaluate the objective at *any* $t \in \mathcal{Q}$ (e.g. its center). For the upper bound U, we use a linear bound on $f(z)$ (see Fig. 5 and the supplement), noting that $\ell_{kk'} \leq z_{kk'}(t) \leq u_{kk'}$ for all $q \in \mathcal{Q}$, where

$$\ell_{kk'} \triangleq \min_{t \in \mathcal{Q}} z_{kk'}(t) \quad \text{and} \quad u_{kk'} \triangleq \max_{t \in \mathcal{Q}} z_{kk'}(t), \qquad (26)$$

whose computation is discussed in Section 5.3. This results in the upper bound U, where

$$\begin{aligned} U &\triangleq \max_{t \in \mathcal{Q}} t^T A t + B^T t + C \\ A &\triangleq -\tfrac{1}{2}\sum_{k,k'} D_{kk'} g_{kk'} S_{kk'}^{-1} \\ B &\triangleq \sum_{k,k'} D_{kk'} g_{kk'} S_{kk'}^{-1} m_{kk'} \\ C &\triangleq \sum_{k,k'} D_{kk'} \left(h_{kk'} - \tfrac{1}{2} g_{kk'} m_{kk'}^T S_{kk'}^{-1} m_{kk'}\right) \\ g_{kk'} &\triangleq \frac{f(u_{kk'}) - f(\ell_{kk'})}{u_{kk'} - \ell_{kk'}} \\ h_{kk'} &\triangleq \frac{u_{kk'} f(\ell_{kk'}) - \ell_{kk'} f(u_{kk'})}{u_{kk'} - \ell_{kk'}}. \end{aligned} \qquad (27)$$

This is a concave quadratic maximization over a rectangular cell $\mathcal{Q}$. Thus, we obtain U as the maximum over all local optima in the interior, faces, edges, and vertices of $\mathcal{Q}$.

5.3. Computing $\ell_{kk'}$ and $u_{kk'}$

Using the form of $z_{kk'}(t)$ in Eq. (22), we have that

$$\ell_{kk'}/u_{kk'} = \min_{t \in \mathcal{Q}} / \max_{t \in \mathcal{Q}} \quad t^T A t + B^T t + C \qquad (28)$$
$$A \triangleq -\tfrac{1}{2} S_{kk'}^{-1}, \; B \triangleq -2Am_{kk'}, \; C \triangleq -\tfrac{1}{2} m_{kk'}^T B.$$

Because of the concavity of the objective, $u_{kk'}$ can be obtained with the exact same algorithm as used to solve Eq. (27). $\ell_{kk'}$ can be obtained by checking the vertices of $\mathcal{Q}$, as the minimum of a concave function over a rectangular cell must occur at one of its vertices.

5.4. Convergence Properties

We now have all the components necessary to optimize Eq. (23) via BB on $\mathbb{R}^3$. As in the rotational alignment case, we provide a characterization (Theorem 2, proof in the supplement) of the maximum refinement depth N required for a desired translational precision ϵ, along with the complexity of the algorithm. Note that while the complexity of BB is exponential in N, N is logarithmic in ϵ^{-1} (Theorem 2), so BB has polynomial complexity in ϵ^{-1}.

Theorem 2 *Suppose γ_0 is the initial diagonal length of the translation cell in $\mathbb{R}^3$, and let*

$$N \triangleq \max\left\{0, \left\lceil \log_2 \tfrac{\gamma_0}{\epsilon} \right\rceil\right\}. \qquad (29)$$

Then at most N refinements are required to achieve a translational tolerance of ϵ, and BB has complexity $O(\epsilon^{-3})$.

6. Results and Evaluation

We evaluate BB (both with and without final local refinement [12]) on four datasets [14, 52, 42] compared to three global methods: an FT-based method [37], GoICP [55] (20% trimming), and GOGMA [9]. To generate the vMF-MMs and GMMs for BB, we cluster the data with DP-vMF-means [47] and DP-means [30], and fit maximum likelihood MMs to the clustered data. To account for nonuniform point densities due to the sensing process, we weight each point's contribution to the MMs by its surface area, estimated by the disc of radius equal to the fifth nearest neighbor distance. We use kNN+PCA [38, 58, 59] to extract surface normals. To improve the robustness of BB, it is run three times on each problem with scale values $\lambda_n \in \{45°, 65°, 80°\}$ in DP-vMF-means (included in the timing results). The scale λ_x for DP-means is manually selected to yield around 50 mixture components, a good trade-off between accuracy and speed. Using Theorems 1 and 2, we terminate rotational BB at $N = 11$ and translational BB at $N = 10$ for
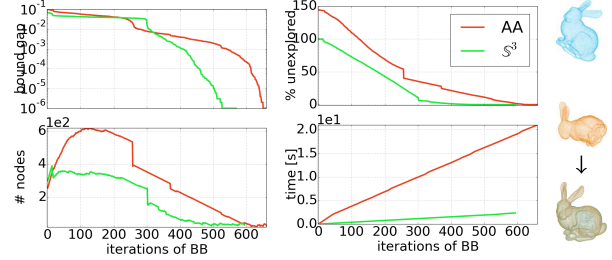

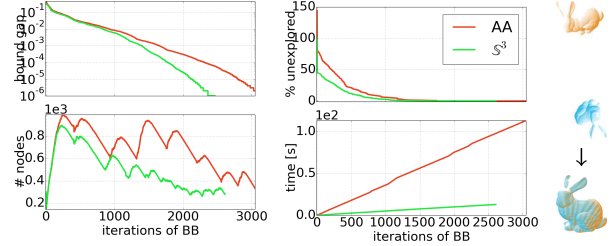Figure 6: BB alignment of the full Stanford Bunny.

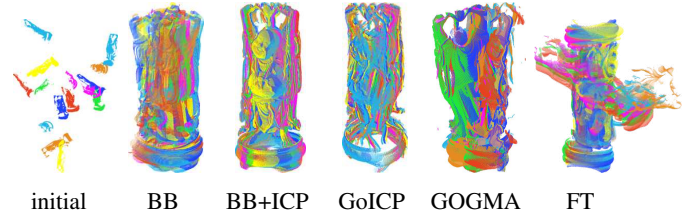Figure 7: Alignment of partial scans of the Stanford Bunny.

initial   BB   BB+ICP   GoICP   GOGMA   FT
Figure 8: Alignment of partial scans of Happy Buddha.

a rotational accuracy of $1°$ and a translational accuracy of $\frac{\gamma_0}{1024}$, where γ_0 is defined in Eq. (24). All timing results include algorithm-specific preprocessing of the data. We used a 3GHz core i7 CPU and a GeForce GTX 780 GPU. While clustering via DP-means and DP-vMF-means uses the GPU, we only use parallel CPU threads for the eight BB bound evaluations after each branch step.

Stanford Bunny [52] Independent of the tessellation strategy, BB perfectly aligns the Stanford Bunny with a randomly transformed version of itself, as shown in Fig. 6. The results of aligning two partial scans of the Stanford Bunny with relative viewpoint difference $45°$ are shown in Fig. 7. BB's initial alignment is close enough to allow ICP to converge to a perfect alignment. The proposed approach leads to a faster reduction in the bound gap, faster exploration, and a smaller number of active nodes, while reducing the computation time per iteration by an order of magnitude vs the AA tessellation. This shows conclusively that the proposed tessellation leads to more efficient BB optimization. Note that the AA tessellation starts at 146% unexplored space because it covers the rotation space more than once as discussed in Sec. 4.1. In both cases BB finds the optimal translation within 200 iterations.

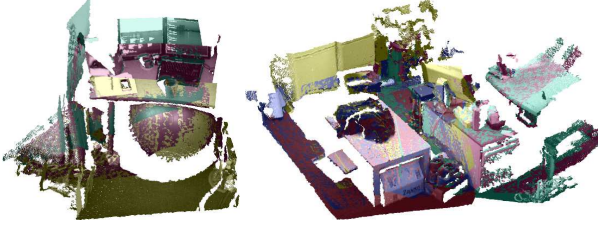**Happy Buddha [14]** This dataset consists of 15 scans

Figure 9: Correct alignment of five noisy, incomplete, and partially overlapping RGB-D point clouds of cluttered indoor scenes using BB+ICP. Colors indicate different scans.

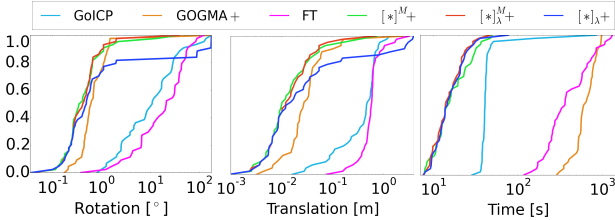Figure 10: Apartment dataset [42] aligned using BB+ICP

Figure 11: Apartment dataset [42] cumulative density functions of rotational error, translational error, and runtime.

taken at $24°$ rotational increments about the vertical axis of a statue. This dataset is challenging, as the scans contain few overlapping points, and the surface normal distributions are anisotropic. We perform pairwise alignment of consecutive scans, and render the aligned scans together in one coordinate system (Fig. 8). The only successful alignment is produced by BB+ICP. This shows the advantage of using surface normals for rotational alignment. Other methods using points (GoICP) or GMMs (GOGMA) have difficulty dealing with ambiguities due to the "flatness" of the scans.

Office Scan Figure 9 demonstrates that BB+ICP finds accurate registrations on noisy, incomplete, cluttered and irregular point clouds as long as good surface normal estimates are available. This demonstrates the potential use of BB+ICP for loop closure detection.

Apartment Dataset [42] This dataset consists of 44 LiDAR scans with an average overlap of 84%. Figure 10 shows the BB+ICP aligned scans of the dataset. Table 1 compares the accuracy and inlier percentages defined by (C)oarse (2m; $10°$), (M)edium (1m; $5°$) and (F)ine (0.5m; $2.5°$) thresholds for all algorithms. For GoICP, we used 100 scan points and an accuracy threshold of 0.01. We used the scale $\lambda_x = 1.3$m for both GOGMA and BB.

Man-made environments such as this dataset exhibit

Method	$[*]_\lambda$	$[*]_{\lambda+}$	$[*]^M$	$[*]^M_+$	$[*]^M_\lambda$	$[*]^M_{\lambda+}$	[9]	[9]+	[55]	[37]
Rot [°]	28.6	26.9	5.52	1.61	3.77	**1.36**	7.14	5.14	24.2	30.0
Tran [m]	0.48	0.43	0.12	0.04	0.08	**0.03**	0.22	0.09	0.46	0.65
Inl % C	79.6	81.8	90.9	95.5	93.2	**97.7**	97.5	97.5	47.7	29.5
Inl % M	75.0	81.8	79.6	95.5	86.4	**97.7**	85.0	97.5	34.1	18.2
Inl % F	54.6	81.8	36.4	95.5	61.4	**97.7**	47.5	97.5	13.6	2.27
Time [s]	**32.6**	50.0	38.4	57.3	140	156	405	675	62.0	470

Table 1: Apartment [42] results using BB [∗], GOGMA [9], GoICP [55], and FT [37]. We denote search over rotational scale via $_\lambda$, search over MW ambiguities with ^{M} and local refinement with $+$. We report rotational (Rot), translational (Tran), timing, and inlier (Inl) percentages for (C)oarse, (M)edium and (F)ine alignment (as defined in the text).

"Manhattan World" (MW) symmetry in their surface normal distributions [46, 49]. We thus transform the rotation obtained via rotational BB by all 24 MW rotations, and search over all using translational BB. Note that doing this is straightforward in the proposed decoupled BB approach, as opposed to a joint approach, e.g. GoICP and GOGMA.

Table 1 and Fig. 11 show that BB with searching over both scale and MW rotations leads to the best accuracy among all algorithms, with a 3x speedup over the 2nd best method, GOGMA (which uses a GPU). From the inlier percentages it is clear that FT and GoICP do not perform well. The CDFs in Fig. 11 show that accounting for MW symmetry (red, green) is important; ignoring it (blue) causes scans to be flipped by $90°/180°$, affecting the mean error strongly. Our method's runtime is spent 80% on preprocessing (scale estimation, mixture fitting) and 20% on the BB search.

Additional Evaluation See Sec. 3 of the supplement for an evaluation of BB's robustness to noise and outliers, as well as its performance in aligning unstructured point clouds of a natural outdoor scene.

7. Conclusion

We introduced a BB approach to global point cloud alignment with convergence guarantees, based on a Bayesian nonparametric point cloud representation and a novel tessellation of rotation space. The method decouples translation and rotation via the use of surface normals, making it more efficient than previous joint approaches. Experiments demonstrate the robustness of the method to noisy real world data, partial overlap, and angular viewpoint differences. We expect that the proposed tessellation of $\mathbb{S}^3$ will be useful in other rotational BB optimization algorithms. All code is available at http://people.csail.mit.edu/jstraub/.

Acknowledgments

References

[1] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM TOG*, volume 27, page 85, 2008. 2

[2] A. Albarelli, E. Rodolà, and A. Torsello. Fast and accurate surface alignment through an isometry-enforcing game. *Pattern Recognition*, 48(7):2209–2226, 2015. 2

[3] C. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 1152–1174, 1974. 1, 2

[4] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Sra, and G. Ridgeway. Clustering on the unit hypersphere using von Mises-Fisher distributions. *JMLR*, 6(9), 2005. 3

[5] M. Bangert, P. Hennig, and U. Oelfke. Using an infinite von Mises-Fisher mixture model to cluster treatment beam directions in external radiation therapy. In *ICMLA*, 2010. 2

[6] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *TPAMI*, 14(2):239–256, 1992. 1

[7] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *IROS*, 2003. 2

[8] M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *IJRR*, 27(6):667–691, 2008. 1, 2

[9] D. Campbell and L. Petersson. Gogma: Globally-optimal gaussian mixture alignment. In *CVPR*, June 2016. 2, 3, 7, 8

[10] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001. 1

[11] J. Chang and J. W. Fisher III. Parallel sampling of DP mixture models using sub-clusters splits. In *NIPS*, 2013. 1, 6

[12] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *ICRA*, 1991. 7

[13] H. S. M. Coxeter. *Regular polytopes*. Courier Corporation, 1973. 3, 4

[14] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996. 7, 8

[15] L. Devroye. *A Course in Density Estimation*. Birkhauser Boston Inc., 1987. 2

[16] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175, 2001. 3

[17] J. R. Driscoll and D. M. Healy. Computing Fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15(2):202–250, 1994. 2

[18] T. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 209–230, 1973. 1

[19] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[20] N. I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, 1995. 1, 3

[21] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Symposium on Geometry Processing*, volume 2, page 5, 2005. 2

[22] R. I. Hartley and F. Kahl. Global optimization through rotation space search. *IJCV*, 82(1):64–79, 2009. 1, 2, 4

[23] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *IJRR*, 31(5):647–663, 2012. 1

[24] B. K. Horn. Some notes on unit quaternions and rotation. 2001. 2

[25] B. K. P. Horn. Extended Gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686, 1984. 1, 2

[26] P. J. Huber. *Robust statistics*. Springer, 1981. 2

[27] T. Ibaraki. Theoretical comparisons of search strategies in branch-and-bound algorithms. *IJCIS*, 5(4):315–344, 1976. 3

[28] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *PAMI*, 33(8):1633–1645, 2011. 2, 3

[29] A. E. Johnson and M. Hebert. Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16(9):635–651, 1998. 1, 2

[30] B. Kulis and M. I. Jordan. Revisiting k-means: New algorithms via Bayesian nonparametrics. In *ICML*, 2012. 2, 3, 6, 7

[31] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, 497–520, 1960. 1, 3

[32] E. L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719, 1966. 1, 3

[33] H. Li and R. Hartley. The 3D-3D registration problem revisited. In *ICCV*, 2007. 1, 2, 4

[34] A. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision. *AMS Math. Comp.*, 65(215):1183–1200, 1996. 4

[35] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007. 1, 2

[36] M. Magnusson, A. Nüchter, C. Lörken, A. J. Lilienthal, and J. Hertzberg. Evaluation of 3D registration reliability and speed-a comparison of ICP and NDT. In *ICRA*, 2009. 2

[37] A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3D point clouds. In *CVPR*, 2006. 2, 7, 8

[38] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Proceedings of the nineteenth annual symposium on Computational geometry*, 2003. 7

[39] N. J. Mitra, A. Nguyen, and L. Guibas. Estimating surface normals in noisy point cloud data. *IJCGA*, 14:261–276, 2004. 1, 2

[40] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 1

[41] A. J. Parra Bustos, T.-J. Chin, and D. Suter. Fast rotation search with stereographic projections for 3D registration. In *CVPR*, 2014. 2, 4

[42] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart. Challenging data sets for point cloud registration algorithms. *IJRR*, 31(14):1705–1711, 2012. 7, 8

[43] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling*, 2001. 2

[44] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *ICRA*, 2009. 2

[45] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007. 1, 2

[46] J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher III. Real-time Manhattan world rotation estimation in 3D. In *IROS*, 2015. 1, 2, 8

[47] J. Straub, T. Campbell, J. P. How, and J. W. Fisher III. Small-variance nonparametric clustering on the hypersphere. In *CVPR*, 2015. 1, 2, 3, 7

[48] J. Straub, O. Freifeld, G. Rosman, J. J. Leonard, and J. W. Fisher. The Manhattan frame model–Manhattan World inference in the space of surface normals. *TPAMI*, 2017. 2

[49] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher III. A Mixture of Manhattan Frames: Beyond the Manhattan World. In *CVPR*, 2014. 8

[50] Y. W. Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer, New York, 2010. 1

[51] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *ECCV*, 2004. 2, 3

[52] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH*, 1994. 7

[53] G. Weiss, C. Wetzler, and E. Von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *IROS*, 1994. 2

[54] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. Leonard, and J. McDonald. Real-time large scale dense RGB-D SLAM with volumetric fusion. *IJRR*, 2014. 1

[55] J. Yang, H. Li, and Y. Jia. Go-ICP: Solving 3D registration efficiently and globally optimally. In *ICCV*, 2013. 2, 4, 7, 8

[56] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *ECCV*, 2016. 2

[57] R. Webb. Stella software. http://www.software3d.com/Stella.php and https://en.wikipedia.org/wiki/600-cell. 1

[58] Meshlab. http://meshlab.sourceforge.net/. Accessed: 2016-11-15. 7

[59] Point cloud library. http://pointclouds.org/. Accessed: 2016-11-15. 7