This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

GOGMA: Globally-Optimal Gaussian Mixture Alignment

Dylan Campbell and Lars Petersson
Australian National University National ICT Australia (NICTA)*
{dylan.campbell,lars.petersson}@anu.edu.au

Abstract

Gaussian mixture alignment is a family of approaches that are frequently used for robustly solving the point-set registration problem. However, since they use local optimisation, they are susceptible to local minima and can only guarantee local optimality. Consequently, their accuracy is strongly dependent on the quality of the initialisation. This paper presents the first globally-optimal solution to the 3D rigid Gaussian mixture alignment problem under the L_2 distance between mixtures. The algorithm, named GOGMA, employs a branch-and-bound approach to search the space of 3D rigid motions $SE(3)$, guaranteeing global optimality regardless of the initialisation. The geometry of $SE(3)$ was used to find novel upper and lower bounds for the objective function and local optimisation was integrated into the scheme to accelerate convergence without voiding the optimality guarantee. The evaluation empirically supported the optimality proof and showed that the method performed much more robustly on two challenging datasets than an existing globally-optimal registration solution.

1. Introduction

Gaussian Mixture Alignment (GMA), the problem of finding the transformation that best aligns one Gaussian mixture with another, has been investigated extensively in computer vision and robotics. It has a natural application to point-set registration, which endeavours to solve the same problem as GMA for discrete point-sets in $\mathbb{R}^n$. Indeed, the Iterative Closest Point (ICP) algorithm [4, 56] and several other local registration algorithms [11, 10, 47, 33] can be interpreted as special cases of GMA [25]. Applications include merging multiple partial scans into a complete model [5, 24]; using registration results as fitness scores for object recognition [26, 3]; registering a view into a global coordinate system for sensor localisation [34, 37]; and finding relative poses between sensors [52, 20].

The dominant solution for 2D and 3D rigid registration is the ICP algorithm [4, 56] and variants, due to its conceptual simplicity, ease of use and good performance. However, ICP is limited by its assumption that closest point pairs should correspond, which fails when the point-sets are not coarsely aligned, or the moving 'model' point-set is not a proper subset of the static 'scene' point-set. The former means that, without a good initialisation, ICP is very susceptible to local minima, producing erroneous estimates without a reliable means of detecting failure. The latter regularly occurs, since differing sensor viewpoints and dynamic objects lead to occlusion and partial-overlap.

Gaussian mixture alignment [10, 47, 25, 8] was introduced to address these shortcomings. By aligning point-sets without establishing explicit point correspondences, GMA is less sensitive to missing correspondences from partial overlap or occlusion and mitigate the problem of local minima by widening the basin of convergence. Robust objective functions can also be applied, such as the L_2 distance between mixtures [25, 8]. However, GMA still requires a good initialisation and cannot guarantee global optimality.

Go-ICP [54, 53] was the first globally-optimal algorithm for the 3D rigid registration problem defined by ICP. Specifically, it used a branch-and-bound approach to find the global minimum of the ICP error metric, the L_2 norm of closest-point residuals. Despite solving the problem of local minima, Go-ICP inherits the non-robust ICP cost function that is susceptible to occlusion and partial overlap. Yang *et al*. [54] proposed a trimming strategy to handle outlier correspondences. However, this increased the runtime and required an additional trimming parameter to be set.

This work is the first to propose a globally-optimal solution to the 3D Gaussian mixture alignment problem under Euclidean (rigid) transformations. It inherits the robust L_2 density distance objective function of L_2 GMA while avoiding the problem of local minima. The method, named GOGMA, employs the branch-and-bound algorithm to guarantee global optimality regardless of initialisation, using a parametrisation of $SE(3)$ space that facilitates branching. The pivotal contribution is the derivation of the objective function bounds using the geometry of $SE(3)$. In addition, local GMA optimisation is applied whenever the algorithm finds a better transformation, to accelerate convergence without voiding the optimality guarantee.

*NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

2. Related Work

The large quantity of work published on ICP, its variants and other registration techniques precludes a comprehensive list. The reader is directed to the surveys on ICP variants [40, 37] and recent 3D point-set and mesh registration techniques [46] for additional background. To improve the robustness of ICP to occlusion and partial overlap, approaches included trimming [9] and outlier rejection [56, 22]. To enlarge ICP's basin of convergence, approaches included LM-ICP [19], which used the Levenberg-Marquardt algorithm [32] and a distance transform to optimise the ICP error, without establishing explicit point correspondences. The family of Gaussian mixture alignment approaches also sought to improve robustness to poor initialisations, noise and outliers. Notable GMA-related algorithms for rigid and non-rigid registration include Robust Point Matching [12], Coherent Point Drift [33], Kernel Correlation [47] and GMMReg [25]. GMMReg [25] defines an equally-weighted Gaussian at every point in the set with identical and isotropic covariances and minimises the robust L_2 distance between densities. The Normal Distributions Transform (NDT) algorithm [30, 45] defines Gaussians for every cell in a grid and estimates full data-driven covariances. SVR [8] uses an SVM to construct a Gaussian mixture with non-uniform weights that adapts to the structure of the point-set and is robust to occlusion and partial overlap. While more robust than ICP, these methods all employ local optimisation, which is dependent on the initial pose.

There are many heuristic or stochastic methods for global alignment that are not guaranteed to converge. One class utilises stochastic optimisation techniques, such as particle filtering [42], genetic algorithms [44, 39] and simulated annealing [5, 36]. Another class is feature-based alignment, which exploits the transformation invariance of a local descriptor to build sparse feature correspondences, such as fast point feature histograms [41]. The transformation can be found from the correspondences using random sampling [41], greedy algorithms [26], Hough transforms [50] or branch-and-bound [21, 2]. SUPER 4PCS [31] is a recent example of a method that uses random sampling without features. It is a four-points congruent sets method that exploits a clever data structure to achieve linear-time performance, extending the original 4PCS algorithm [1].

In contrast, globally-optimal techniques avoid local minima by searching the entire transformation space, often using the branch-and-bound paradigm. Existing 3D methods [29, 35, 6, 54] are often very slow or make restrictive assumptions about the point-sets, correspondences or transformations. For example, Li and Hartley [29] minimised a Lipschitzized L_2 error function using branch-and-bound, but assumed that the point-sets were the same size and the transformation was pure rotation. Olsson *et al.* [35] found optimal solutions to point-to-point/line/plane registration using branch-and-bound and bilinear relaxation of rotation quaternions, but assumed correspondences were known. Recently, Bustos *et al.* [6] achieved efficient runtimes using stereographic projection techniques for optimal 3D alignment, but assumed that translation was known. Finally, Yang *et al.* [54, 53] proposed the Go-ICP algorithm, which finds the optimal solution to the closest point L_2 error between point-sets and is accelerated by using local ICP as a sub-routine. However, it is sensitive to occlusion and partial overlap, due to its non-robust cost function. The proposed trimming strategy goes some way to alleviating this, but increases the runtime, requires an estimate of the overlap percentage and may lead to ambiguity in the solution.

The rest of the paper is organised as follows: we outline Gaussian mixture alignment in Section 3, we develop a parametrisation of the domain of 3D motions, a branching strategy and a derivation of the bounds in Section 4, we propose an algorithm for globally-optimal GMA in Section 5 and we evaluate the its performance in Section 6.

3. Gaussian Mixture Alignment

The alignment of Gaussian Mixture Models (GMMs) to solve the point-set registration task is a well-studied problem [10, 47, 30, 25, 8]. GMMs can be constructed from point-set data using Kernel Density Estimation (KDE) [25, 17, 13, 51], Expectation Maximisation (EM) [15, 16] or a Support Vector Machine (SVM) [8]. Once the point-sets are in GMM form, the registration problem can be posed as minimising a discrepancy measure between GMMs. We use the L_2 distance formulation of Jian and Vemuri [25], which can be expressed in closed-form and efficiently implemented. The L_2E estimator minimises the L_2 distance between densities and is inherently robust to outliers [43].

Let $\mathcal{G}_\mathcal{X} = \{\mathbf{x}_i, \sigma^2_{i\mathcal{X}}, \phi^\mathcal{X}_i\}_{i=1}^m$ and $\mathcal{G}_\mathcal{Y} = \{\mathbf{y}_j, \sigma^2_{j\mathcal{Y}}, \phi^\mathcal{Y}_j\}_{j=1}^n$ be GMMs constructed from point-sets $\mathcal{X}$ and $\mathcal{Y}$, with means $\mathbf{x}_i$ and $\mathbf{y}_j$, variances $\sigma^2_{i\mathcal{X}}$ and $\sigma^2_{j\mathcal{Y}}$, mixture weights $\phi^\mathcal{X}_i$ and $\phi^\mathcal{Y}_j$ and number of components m and n respectively. Also let $T(\mathcal{G}, \mathbf{R}, \mathbf{t})$ be the function that rigidly transforms $\mathcal{G}$ with rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. The L_2 distance D between transformed $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$ is given by

$$D(\mathbf{R}, \mathbf{t}) = \int_{\mathbb{R}^3} \left(p\left(\mathbf{p}|T(\mathcal{G}_\mathcal{X}, \mathbf{R}, \mathbf{t})\right) - p\left(\mathbf{p}|\mathcal{G}_\mathcal{Y}\right)\right)^2 \mathrm{d}\mathbf{p} \quad (1)$$

where $p\left(\mathbf{p}|\mathcal{G}\right)$ is the probability of observing a point $\mathbf{p}$ given a mixture model $\mathcal{G} = \{\boldsymbol{\mu}_i, \sigma^2_i, \phi_i\}_{i=1}^\ell$, that is

$$p\left(\mathbf{p}|\mathcal{G}\right) = \sum_{i=1}^{\ell} \phi_i \mathcal{N}\left(\mathbf{p}\middle|\boldsymbol{\mu}_i, \sigma^2_i\right) \quad (2)$$

where $\mathcal{N}\left(\mathbf{p}\middle|\boldsymbol{\mu}, \sigma^2\right)$ is the probability of the Gaussian at $\mathbf{p}$. Expanding (1), $\left[p\left(\mathbf{p}|T(\mathcal{G}_\mathcal{X}, \mathbf{R}, \mathbf{t})\right)\right]^2$ is invariant under rigid transformations and $\left[p\left(\mathbf{p}|\mathcal{G}_\mathcal{Y}\right)\right]^2$ is independent of $(\mathbf{R}, \mathbf{t})$.

The integral of the $-2p\left(\mathbf{p}|T(\mathcal{G}_{\mathcal{X}},\mathbf{R},\mathbf{t})\right)p\left(\mathbf{p}|\mathcal{G}_{\mathcal{Y}}\right)$ term has a closed form, derived by applying the identity

$$\int_{\mathbb{R}^3}\mathcal{N}\left(\mathbf{p}|\boldsymbol{\mu}_1,\sigma_1^2\right)\mathcal{N}\left(\mathbf{p}|\boldsymbol{\mu}_2,\sigma_2^2\right)\,\mathrm{d}\mathbf{p} = \mathcal{N}\left(\mathbf{0}|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2,\sigma_1^2+\sigma_2^2\right). \quad (3)$$

Therefore, by substituting in (2) and (3), the objective function for Gaussian Mixture Alignment (GMA) is given by

$$\begin{aligned} f\left(\mathbf{R},\mathbf{t}\right) &= -\int_{\mathbb{R}^3} p\left(\mathbf{p}|T(\mathcal{G}_{\mathcal{X}},\mathbf{R},\mathbf{t})\right)p\left(\mathbf{p}|\mathcal{G}_{\mathcal{Y}}\right)\,\mathrm{d}\mathbf{p} \\ &= -\sum_{i=1}^{m}\sum_{j=1}^{n}\phi_i^{\mathcal{X}}\phi_j^{\mathcal{Y}}\mathcal{N}\left(\mathbf{0}\middle|\mathbf{R}\mathbf{x}_i+\mathbf{t}-\mathbf{y}_j,\sigma_{i\mathcal{X}}^2+\sigma_{j\mathcal{Y}}^2\right) \\ &= -\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\phi_i^{\mathcal{X}}\phi_j^{\mathcal{Y}}}{Z}\exp\left[-\frac{\left[e_{ij}\left(\mathbf{R},\mathbf{t}\right)\right]^2}{2\left[\sigma_{i\mathcal{X}}^2+\sigma_{j\mathcal{Y}}^2\right]}\right] \quad (4) \end{aligned}$$

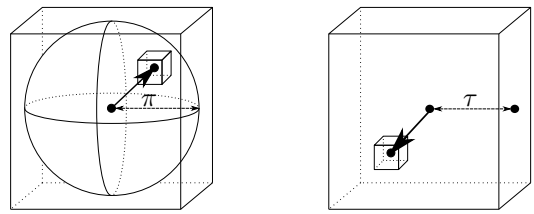
where Z is the normalisation factor and $e_{ij}\left(\mathbf{R},\mathbf{t}\right)$ is the pairwise residual error. For local GMA, we use the quasi-Newton L-BFGS-B algorithm [7] to minimise (4).

4. Branch-and-Bound

Branch-and-Bound (BB) [27] is a global optimisation technique that is frequently used to solve non-convex and NP-hard problems [28]. To apply the technique to 3D GMA, a suitable means of parametrising and branching (partitioning) the domain of 3D rigid transformations $SE(3)$ must be found, as well as an efficient way to find the upper and lower bounds of each branch. The latter is critical for the time and memory efficiency of the algorithm, since tight bounds that are quick to evaluate reduce the search space by allowing suboptimal branches to be pruned.

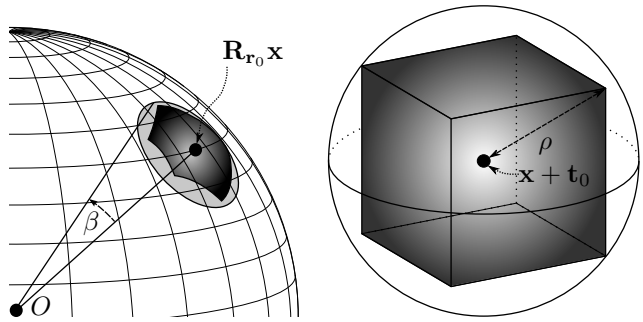
4.1. Parametrising and Branching the Domain

For the globally-optimal alignment problem, the objective function (4) must be minimised over the domain of 3D motions, that is, the group $SE(3)=SO(3)\times\mathbb{R}^3$. One minimal parametrisation of $SO(3)$ is the angle-axis representation, where a rotation is parametrised as a 3-vector $\mathbf{r}$ with a rotation angle $\|\mathbf{r}\|$ and rotation axis $\mathbf{r}/\|\mathbf{r}\|$. We use the notation $\mathbf{R}_{\mathbf{r}}\in SO(3)$ to denote the rotation matrix obtained from the matrix exponential map of the skew-symmetric matrix $[\mathbf{r}]_\times$ induced by $\mathbf{r}$. The Rodrigues' rotation formula can be used to efficiently calculate this mapping. The space of all 3D rotations can therefore be represented as a solid ball of radius π in $\mathbb{R}^3$. The mapping is one-to-one on the interior of the π-ball and two-to-one on the surface. For ease of manipulation, we use the 3D cube circumscribing the π-ball as the rotation domain [29]. The translation domain is chosen as the bounded cube $[-\tau,\tau]^3$. If the GMMs were constructed from point-sets scaled to fit within $[-0.5,0.5]^3$,



(a) Rotation Domain (b) Translation Domain

Figure 1. Parametrisation of $SE(3)$. (a) The rotation space $SO(3)$ is parametrised by angle-axis 3-vectors within a solid radius-π ball. (b) The translation space $\mathbb{R}^3$ is parametrised by 3-vectors within a cube of half side-length τ. The joint domain is branched using a hyperoctree data structure, with a sub-hypercube depicted as two sub-cubes in the rotation and translation dimensions.



(a) Rotation Uncertainty Region (b) Translation Uncertainty Region

Figure 2. Uncertainty region induced by hypercube $C=C_r\times C_t$. (a) Rotation uncertainty region for C_r with centre $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}$. The optimal rotation of $\mathbf{x}$ may be anywhere within the heavily-shaded umbrella-shaped uncertainty region, which is entirely contained by the lightly-shaded spherical cap defined by $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}$ and β. (b) Translation uncertainty region for C_t with centre $\mathbf{x}+\mathbf{t}_0$. The optimal translation of $\mathbf{x}$ may be anywhere within the cube, which is entirely contained by the circumscribed sphere with radius ρ.

choosing $\tau=1$ would ensure that the domain covered every feasible translation. In practice, a smaller τ is acceptable, if a minimum detectable bounding box overlap is acceptable.

Together, these domains form a 6D hypercube, shown separately in Figure 1. During BB, the hypercube is branched using a hyperoctree data structure. The uncertainty region induced by a hypercube on a point $\mathbf{x}$ is shown for rotation and translation separately in Figure 2. The transformed point may lie anywhere in the uncertainty region, which is the Minkowski sum of a spherical patch and a cube for rotation and translation dimensions respectively.

4.2. Bounding the Branches

The success of a BB algorithm is predicated on the quality of its bounds. For rigid 3D Gaussian mixture alignment, the GMA objective function (4) within a transformation domain $C_r\times C_t$ needs to be bounded. To simplify the bounds, the GMMs are assumed to have isotropic covariances. The translation component $\mathbf{t}\in C_t$ can be bounded by a single value ρ by observing that the translation cube can be inscribed in a sphere (Figure 2b), as in [54].

Lemma 1. *(Translation sphere radius) Given a 3D point* $\mathbf{x}$ *and a translation cube* C_t *of half side-length* δ_t *centred at* $\mathbf{t}_0$*, then* $\forall \mathbf{t} \in C_t$*,*

$$\|(\mathbf{x}+\mathbf{t})-(\mathbf{x}+\mathbf{t}_0)\| \leqslant \sqrt{3}\delta_t \doteq \rho. \tag{5}$$

Proof. Inequality (5) can be derived by observing that $\|(\mathbf{x}+\mathbf{t})-(\mathbf{x}+\mathbf{t}_0)\| = \|\mathbf{t}-\mathbf{t}_0\|$ and $\max \|\mathbf{t}-\mathbf{t}_0\| = \sqrt{3}\delta_t$ (the half space diagonal) for $\mathbf{t} \in C_t$. □

To bound the rotation component $\mathbf{r} \in C_r$, Lemmas 1 and 2 from [23] are used. For reference, the relevant parts are merged into Lemma 2, as in [53]. The lemma indicates that the angle between two rotated vectors is less than or equal to the Euclidean distance between their rotations' angle-axis representations in $\mathbb{R}^3$.

Lemma 2. *For an arbitrary vector* $\mathbf{x}$ *and two rotations, represented as* $\mathbf{R}_{\mathbf{r}_1}$ *and* $\mathbf{R}_{\mathbf{r}_2}$ *in matrix form and* $\mathbf{r}_1$ *and* $\mathbf{r}_2$ *in angle-axis form,*

$$\angle(\mathbf{R}_{\mathbf{r}_1}\mathbf{x},\, \mathbf{R}_{\mathbf{r}_2}\mathbf{x}) \leqslant \|\mathbf{r}_1-\mathbf{r}_2\|. \tag{6}$$

From this, the maximum angle β between a vector $\mathbf{x}$ rotated by $\mathbf{r}_0$ and $\mathbf{x}$ rotated by $\mathbf{r} \in C_r$ can be found as follows.

Lemma 3. *(Maximum aperture angle) Given a 3D point* $\mathbf{x}$ *and a rotation cube* C_r *of half side-length* δ_r *centred at* $\mathbf{r}_0$*, then* $\forall \mathbf{r} \in C_r$*,*

$$\angle(\mathbf{R}_{\mathbf{r}}\mathbf{x}, \mathbf{R}_{\mathbf{r}_0}\mathbf{x}) \leqslant \min(\sqrt{3}\delta_r, \pi) \doteq \beta. \tag{7}$$

Proof. Inequality (7) can be derived as follows:

$$\angle(\mathbf{R}_{\mathbf{r}}\mathbf{x}, \mathbf{R}_{\mathbf{r}_0}\mathbf{x}) \leqslant \min(\|\mathbf{r}-\mathbf{r}_0\|, \pi) \tag{8}$$
$$\leqslant \min(\sqrt{3}\delta_r, \pi) \tag{9}$$

where (8) follows from Lemma 2 and (9) follows from $\max \|\mathbf{r}-\mathbf{r}_0\| = \sqrt{3}\delta_r$ (the half space diagonal of the rotation cube) for $\mathbf{r} \in C_r$. □

As a first step towards bounding the GMA objective function, bounds for the pairwise residual error $e_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t})$ need to be found. This $e_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t})$ represents the minimal L_2 distance between the Gaussian means $(\mathbf{R}_{\mathbf{r}}\mathbf{x}_i+\mathbf{t})$ and $\mathbf{y}_j$ for $(\mathbf{r}, \mathbf{t}) \in (C_r \times C_t)$. For convenience, let $\mathbf{y}'_j = (\mathbf{y}_j - \mathbf{t}_0)$.

Theorem 1. *(Bounds of the pairwise residual error) For the 3D transformation domain* $C_r \times C_t$ *centred at* $(\mathbf{r}_0, \mathbf{t}_0)$ *with translation sphere radius* ρ*, the upper bound* $\bar{e}_{ij}$ *and the lower bound* $\underline{e}_{ij}$ *of the optimal pairwise residual error* $e_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t})$ *for* $\mathbf{x}_i$ *and* $\mathbf{y}_j$ *can be chosen as*

$$\bar{e}_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t}) = e_{ij}(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) \tag{10}$$

$$\underline{e}_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t}) = \max\Big(\big|\|\mathbf{x}_i\| - \|\mathbf{y}_j - \mathbf{t}_0\|\big| - \rho,\, 0\Big). \tag{11}$$

Proof. The validity of the upper bound $\bar{e}_{ij}$ follows from the error e_{ij} at a specific point within the domain being larger than the minimal error within the domain, that is

$$e_{ij}(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) \geqslant \min_{\forall(\mathbf{r},\mathbf{t})\in(C_r\times C_t)} e_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t}). \tag{12}$$

The validity of the lower bound $\underline{e}_{ij}$ is proved as follows. Observe that $\forall(\mathbf{r}, \mathbf{t}) \in (C_r \times C_t)$,

$$e_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t}) = \|\mathbf{R}_{\mathbf{r}}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_j\| \tag{13}$$
$$= \|\mathbf{R}_{\mathbf{r}}\mathbf{x}_i - (\mathbf{y}_j - \mathbf{t}_0) - (\mathbf{t}_0 - \mathbf{t})\| \tag{14}$$
$$\geqslant \big|\|\mathbf{R}_{\mathbf{r}}\mathbf{x}_i - \mathbf{y}'_j\| - \|(\mathbf{t}_0 - \mathbf{t})\|\big| \tag{15}$$
$$\geqslant \max\big(\|\mathbf{R}_{\mathbf{r}}\mathbf{x}_i - \mathbf{y}'_j\| - \|(\mathbf{t}_0 - \mathbf{t})\|,\, 0\big) \tag{16}$$
$$\geqslant \max\big(\|\mathbf{R}_{\mathbf{r}}\mathbf{x}_i - \mathbf{y}'_j\| - \rho,\, 0\big) \tag{17}$$
$$\geqslant \max\big(\big|\|\mathbf{R}_{\mathbf{r}}\mathbf{x}_i\| - \|\mathbf{y}'_j\|\big| - \rho,\, 0\big) \tag{18}$$
$$= \max\big(\big|\|\mathbf{x}_i\| - \|\mathbf{y}_j - \mathbf{t}_0\|\big| - \rho,\, 0\big) \tag{19}$$

where (14) adds and subtracts $\mathbf{t}_0$, (15) follows from the reverse triangle inequality $\|\mathbf{x}-\mathbf{y}\| \geqslant |\|\mathbf{x}\| - \|\mathbf{y}\||$, (16) from the absolute value of a quantity being positive, (17) follows from (5), and (18) from the reverse triangle inequality. □

The geometric intuition of Theorem 1 is that all valid points $(\mathbf{R}_{\mathbf{r}}\mathbf{x}_i + \mathbf{t})$ lie within ρ of the rotation sphere centred at $\mathbf{t}_0$ with radius $\|\mathbf{x}_i\|$. However, the gap between this 'sphere distance' pairwise lower bound and the pairwise upper bound does not converge to zero as the sub-cube sizes decrease, since the lower bound is independent of the rotation sub-cube size δ_r. We can find a converging lower bound by observing that all valid points, neglecting translation uncertainty, lie on a spherical cap. Letting $\mathbf{x}_i^0 = \mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i$, the spherical cap is defined by the sphere of radius $\|\mathbf{x}_i\|$ centred at $\mathbf{t}_0$ with the constraint that $\angle\left(\mathbf{x}_i^0, \mathbf{x}\right) \leqslant \beta$ for all points $(\mathbf{x} + \mathbf{t}_0)$ on the cap. Now let $\mathbf{x}'_i = \mathbf{R}\mathbf{x}_i$ be an arbitrary point on the origin-centred spherical cap and let $\mathbf{x}^*_{ij} = \mathbf{R}\mathbf{x}_i$ be the point coplanar with $\mathbf{x}_i^0$ and $\mathbf{y}'_j$ on the edge of the spherical cap, such that $\angle\left(\mathbf{x}_i^0, \mathbf{x}^*_{ij}\right) = \beta$.

Theorem 2. *(Tight lower bound of the pairwise residual error) For the 3D transformation domain* $C_r \times C_t$ *centred at* $(\mathbf{r}_0, \mathbf{t}_0)$ *with translation sphere radius* ρ*, the lower bound* $\underline{e}_{ij}$ *of the optimal pairwise residual error* $e_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t})$ *for* $\mathbf{x}_i$ *and* $\mathbf{y}_j$ *can be chosen as*

$$\underline{e}_{ij}(\mathbf{R}_{\mathbf{r}}, \mathbf{t}) = \begin{cases} \max\Big[\big|\|\mathbf{x}_i\| - \|\mathbf{y}'_j\|\big| - \rho, 0\Big] & \text{for } \alpha \leqslant \beta \\ \max\Big[\|\mathbf{x}^*_{ij} - \mathbf{y}'_j\| - \rho, 0\Big] & \text{for } \alpha > \beta \end{cases} \tag{20}$$

where α *and* β *are shown in Figure 3 and are given by*

$$\alpha = \angle\left(\mathbf{x}_i^0, \mathbf{y}'_j\right) \tag{21}$$
$$\beta = \angle\left(\mathbf{x}_i^0, \mathbf{x}^*_{ij}\right) = \min(\sqrt{3}\delta_r, \pi). \tag{22}$$

Proof. Observe that $\forall(\mathbf{r},\mathbf{t}) \in (C_r \times C_t)$,

$$e_{ij}(\mathbf{R_r},\mathbf{t}) \geqslant \max\left(\|\mathbf{R_r}\mathbf{x}_i - \mathbf{y}'_j\| - \rho,\, 0\right) \tag{23}$$

$$\geqslant \max\left(\min\|\mathbf{R_r}\mathbf{x}_i - \mathbf{y}'_j\| - \rho,\, 0\right) \tag{24}$$

$$\geqslant \max\left(\min\|\mathbf{x}'_i - \mathbf{y}'_j\| - \rho,\, 0\right) \tag{25}$$

$$= \begin{cases} \max\left[\left|\|\mathbf{x}_i\| - \|\mathbf{y}'_j\|\right| - \rho, 0\right] & \text{for } \alpha \leqslant \beta \\ \max\left[\|\mathbf{x}^*_{ij} - \mathbf{y}'_j\| - \rho, 0\right] & \text{for } \alpha > \beta \end{cases} \tag{26}$$

where (23) is transferred from (17), (25) states that the minimum distance to a constrained point on the spherical cap is greater than or equal to the minimum distance to an unconstrained point on the cap and (26) is from Theorem 3. □

Theorem 3. *(Spherical cap distance) For the spherical cap defined by the vector $\mathbf{x}'_i + \mathbf{t}_0$ constrained by $\angle(\mathbf{x}'_i, \mathbf{x}^0_i) \leqslant \beta$, the minimum distance from a point $\mathbf{y}_j$ to the spherical cap is given by*

$$\min\|\mathbf{x}'_i + \mathbf{t}_0 - \mathbf{y}_j\| = \begin{cases} \left|\|\mathbf{x}_i\| - \|\mathbf{y}_j - \mathbf{t}_0\|\right| & \textit{for } \alpha \leqslant \beta \\ \|\mathbf{x}^*_{ij} - (\mathbf{y}_j - \mathbf{t}_0)\| & \textit{for } \alpha > \beta \end{cases} \tag{27}$$

where α and β are as per Theorem 2, shown in Figure 3.

Proof. Dropping the subscripts and translating everything by $(-\mathbf{t}_0)$, an arbitrary point $\mathbf{x}'$ on the spherical cap can be expressed as the rotation of the point $\mathbf{x}^0$ about the sphere centre towards $\mathbf{y}'$ by an angle $\gamma \in [0, \beta]$, followed by a rotation of this intermediate vector about the axis $\mathbf{x}^0$ by θ. By two applications of the Rodrigues' rotation formula,

$$\mathbf{x}' = \left[\mathrm{c}_\gamma - \frac{\mathrm{c}_\alpha\, \mathrm{s}_\gamma\, \mathrm{c}_\theta}{\mathrm{s}_\alpha}\right]\mathbf{x}^0 + \frac{\mathrm{s}_\gamma}{\mathrm{s}_\alpha}\frac{\mathbf{x}^0 \times \mathbf{y}'}{\|\mathbf{y}'\|} + \frac{\mathrm{s}_\gamma\, \mathrm{c}_\theta\, \|\mathbf{x}\|}{\mathrm{s}_\alpha\, \|\mathbf{y}'\|}\mathbf{y}' \tag{28}$$

where $\mathrm{s}_A = \sin A$ and $\mathrm{c}_A = \cos A$. By substitution, the squared distance between the point $\mathbf{y}'$ and an arbitrary point on the spherical cap is given by

$$\|\mathbf{x}' - \mathbf{y}'\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}'\|^2 - 2\left[\mathrm{c}_\alpha \mathrm{c}_\gamma + \mathrm{s}_\alpha \mathrm{s}_\gamma \mathrm{c}_\theta\right]\|\mathbf{x}\|\|\mathbf{y}'\| \tag{29}$$

and is minimised when $\theta = 0$. Therefore,

$$\min\|\mathbf{x}' - \mathbf{y}'\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}'\|^2 - 2\cos[\alpha - \gamma]\|\mathbf{x}\|\|\mathbf{y}'\|. \tag{30}$$

When $\alpha \leqslant \beta$ (Case 1), (30) is minimised when $\gamma = \alpha$:

$$\min\|\mathbf{x}' - \mathbf{y}'\|^2 = \left(\|\mathbf{x}\| - \|\mathbf{y}'\|\right)^2 \tag{31}$$

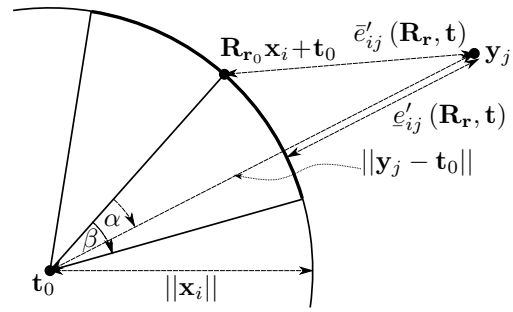
$$\therefore \min\|\mathbf{x}' - \mathbf{y}'\| = \left|\|\mathbf{x}\| - \|\mathbf{y}'\|\right| \text{ for } \alpha \leqslant \beta. \tag{32}$$

When $\alpha > \beta$ (Case 2), (30) is minimised when $\gamma = \beta$:

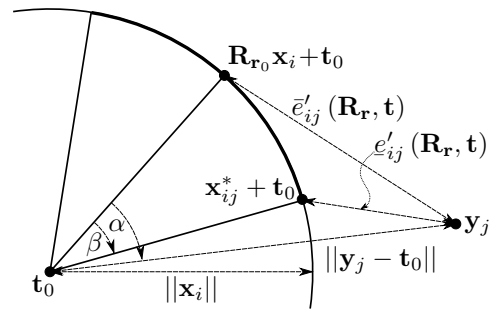
$$\min\|\mathbf{x}' - \mathbf{y}'\|^2 = \|\mathbf{x}^* - \mathbf{y}'\|^2 \tag{33}$$

$$\therefore \min\|\mathbf{x}' - \mathbf{y}'\| = \|\mathbf{x}^* - \mathbf{y}'\| \text{ for } \alpha > \beta. \tag{34}$$

The full proof is left for the appendix. □



(a) Case 1: $\mathbf{y}_j$ is within the rotation cone ($\alpha \leqslant \beta$).



(b) Case 2: $\mathbf{y}_j$ is outside the rotation cone ($\alpha > \beta$).

Figure 3. Upper and lower bound of the pairwise residual error, neglecting translation. A 2D cross-section of the plane defined by points $\{\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i + \mathbf{t}_0, \mathbf{y}_j, \mathbf{t}_0\}$ is shown.

The geometric intuition, shown in Figure 3, for Theorems 2 and 3 is that the minimum distance to the spherical cap is equal to (i) the radial distance to the sphere if $\mathbf{y}$ lies within the rotation cone ($\alpha \leqslant \beta$) or (ii) the distance to the edge of the cap. This 'spherical cap distance' pairwise lower bound is a tighter bound than (11) and the gap between it and the upper bound converges to zero. This can be shown by observing that the size of the spherical cap diminishes with the size of the rotation sub-cube (7) and likewise for the translation sphere and translation sub-cube (5). It is also a tighter bound than that in [54], which uses the distance to a sphere enclosing the spherical cap.

The bounds of the objective function are found by summing the kernelised upper and lower bounds of the pairwise residual errors in (10) and (20) for all $m \times n$ Gaussian pairs.

Corollary 1. *(Bounds of the objective function) For the 3D transformation domain $C_r \times C_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$ with translation sphere radius ρ, the upper bound $\bar{f}$ and the lower bound $\underline{f}$ of the optimal objective function value $f(\mathbf{R_r},\mathbf{t})$ can be chosen as*

$$\bar{f}(\mathbf{R_r},\mathbf{t}) = -\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\phi_i^{\mathcal{X}}\phi_j^{\mathcal{Y}}}{Z}\exp\left[-\frac{\left[\bar{e}_{ij}(\mathbf{R_r},\mathbf{t})\right]^2}{2\left[\sigma_{i\mathcal{X}}^2 + \sigma_{j\mathcal{Y}}^2\right]}\right] \tag{35}$$

$$\underline{f}(\mathbf{R_r},\mathbf{t}) = -\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\phi_i^{\mathcal{X}}\phi_j^{\mathcal{Y}}}{Z}\exp\left[-\frac{\left[\underline{e}_{ij}(\mathbf{R_r},\mathbf{t})\right]^2}{2\left[\sigma_{i\mathcal{X}}^2 + \sigma_{j\mathcal{Y}}^2\right]}\right]. \tag{36}$$

5. The GOGMA Algorithm

The Globally-Optimal Gaussian Mixture Alignment (GOGMA) algorithm is outlined in Algorithm 1. It employs branch-and-bound with depth-first search using a priority queue where the priority is inverse to the lower bound (Line 4). The algorithm terminates with ϵ-optimality, whereby the difference between the best function value f^* so far and the global lower bound $\underline{f}$ is less than ϵ (Line 5).

In this implementation, the upper and lower bounds of 4096 sub-cubes are found simultaneously on the GPU (Line 6). A higher branching factor can be used, although memory considerations must be taken into account to ensure that the priority queue does not increase much faster than it can be pruned. A branching factor of 4096 performs well and does not require a high-end GPU. Other than bound calculation, the code is executed entirely on the CPU.

Lines 1 and 8 show how the local Gaussian Mixture Alignment (GMA) algorithm is integrated. Firstly, the best-so-far function value f^* and the associated transformation parameters are initialised using GMA (Line 1). Within the main loop, GMA is run whenever the BB algorithm finds a sub-cube C_i that has an upper bound less than the best-so-far function value f^* (Line 8). GMA is initialised with $(\mathbf{r}_{0i}, \mathbf{t}_{0i})$, the centre transformation of C_i. In this way, BB and GMA collaborate, with GMA quickly converging to the closest local minimum and BB guiding the search into the convergence basins of increasingly lower local minima. Hence, BB jumps the search out of local minima and GMA accelerates convergence by refining f^*. Importantly, the faster f^* is refined, the more sub-cubes are discarded, since those with lower bounds higher than f^* are culled (Line 9).

The algorithm is designed in such a way that early termination outputs the best-so-far transformation. Hence, if a limit is set on the runtime, a best-guess transformation can be provided for those alignments that exceed the threshold. While ϵ-optimality will not be guaranteed for them, in practise this is often adequate. In view of this, and to accelerate the removal of redundant sub-cubes, Line 8 may be modified such that GMA is run for every sub-cube of the first subdivision and f^* is updated with the best function value of that batch. We denote this as batch-initialised GOGMA.

Algorithm 1 GOGMA: An algorithm for globally-optimal Gaussian mixture alignment in $SE(3)$

Input: mixture models $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$, parametrised by means $\mathbf{x}$ and $\mathbf{y}$ respectively, variances σ^2 and mixture weights ϕ; threshold ϵ; initial transformation hypercube $\mathcal{C} = \mathcal{C}_r \times \mathcal{C}_t$ centred at $(\mathbf{r}_0, \mathbf{t}_0)$.

Output: ϵ-optimal value f^* and corresponding $\mathbf{r}^*$, $\mathbf{t}^*$.

1: Run GMA: $(f^*, \mathbf{r}^*, \mathbf{t}^*) \leftarrow \text{GMA}(\mathbf{r}_0, \mathbf{t}_0)$
2: Add hypercube $\mathcal{C}$ to priority queue Q
3: **loop**
4: Remove cube C with lowest lower-bound $\underline{f}$ from Q
5: **if** $f^* - \underline{f} \leqslant \epsilon$ **then** Terminate
6: In parallel, evaluate $\bar{f}_i$ and $\underline{f}_i$ for all sub-cubes of C
7: **for all** sub-cubes C_i **do**
8: **if** $\bar{f}_i < f^*$ **then** $(f^*, \mathbf{r}^*, \mathbf{t}^*) \leftarrow \text{GMA}(\mathbf{r}_{0i}, \mathbf{t}_{0i})$
9: **if** $\underline{f}_i < f^*$ **then** Add C_i to queue: $Q \leftarrow C_i$

6. Experimental Results

GOGMA was evaluated using many different point-sets, including 3D data collected in the lab and in the field. In order to test the algorithms across a uniformly-distributed sample of $SO(3)$, the 72 base grid rotations from Incremental Successive Orthogonal Images (ISOI) [55] were used. Translation perturbations were not applied, since centring and scaling the point-sets to $[-1, 1]^3$ before conversion to GMM removes these perturbations. The transformation domain was set to be $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$. This corresponds to a minimum detectable bounding box overlap of ~42%.

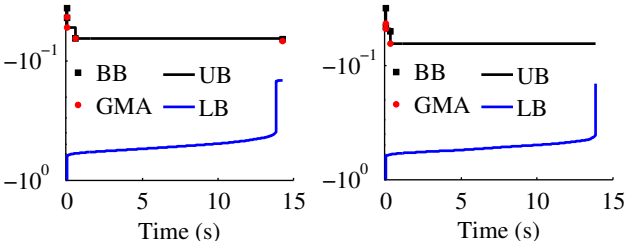
Except where otherwise specified, the convergence threshold was set to $\epsilon = 0.1$, the number of Gaussian components was set to $m, n \approx 50$, batch initialisation was used and the GMMs were Support Vector–parametrised Gaussian Mixtures (SVGMs), whereby an SVM and a mapping are used to efficiently construct an adaptive GMM from point-set data [8]. SVGMs allow the user to specify the approximate number of components and set equal variances automatically, based on the desired number of components.

Although GOGMA is a general-purpose Gaussian mixture alignment algorithm, the runtime results include the time required for GMM construction, to facilitate comparison with other point-set registration algorithms. All experiments were run on a PC with a 3.7GHz Quad Core CPU with 32GB of RAM and a Nvidia GeForce GTX 980 GPU. The GOGMA code is written in unoptimised C++ and uses the VXL numerics library [49] for local GMA optimisation.

It is crucial to observe that finding the global optimum does not necessarily imply finding the ground-truth transformation. For GMMs that describe partially overlapping point-sets, there may exist an alignment that produces a smaller function value than the ground-truth alignment. However, the L_2 density distance objective function is much less susceptible to partial overlap than others [8], including the L_2 norm closest point error that is minimised by ICP.

6.1. Optimality

To demonstrate optimality of the algorithm with respect to the objective function, the reconstructed DRAGON [14] and BUNNY [48] point-sets were aligned with transformed copies of themselves, using the 72 ISOI rotations. Identical point-sets were required to obtain the ground-truth optimal objective function values. The global optimum was found for all 144 registrations, with mean separations from the optimal value being 9×10^{-8} and 3×10^{-7} and mean runtimes being 17s and 14s for DRAGON and BUNNY respectively.



(a) DRAGON (b) BUNNY

Figure 4. Evolution of the upper and lower bounds for the reconstructed DRAGON and BUNNY models. The objective function value is plotted against time. Note the logarithmic scale.

Table 1. Effect of GMM type on the accuracy and runtime of the GOGMA algorithm. The mean/max translation error (in metres), rotation error (in degrees) and runtime (in seconds) are reported.

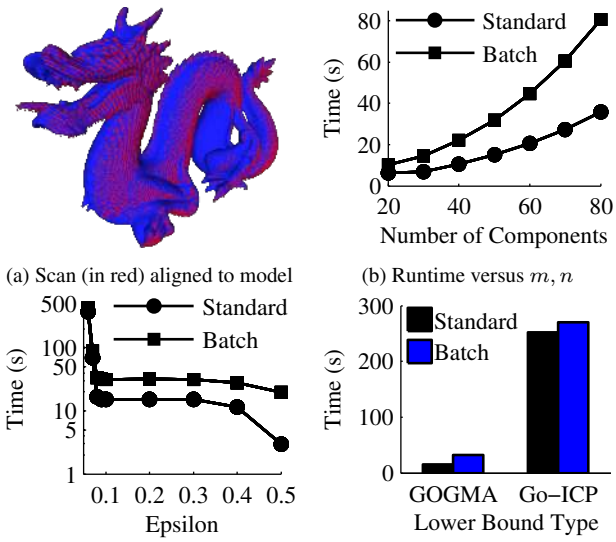
GMM Type	SVGM	KDE	EM
Translation	**0.004/0.008**	0.14/0.21	0.02/0.18
Rotation	**1.5/2.7**	116/167	7.2/80
Runtime	34/50	**15/15**	4960/4965

For batch initialisation, the mean runtimes were 33s and 29s. The evolution of the global upper and lower bounds is shown in Figure 4. It can be seen that BB and GMA collaborate to reduce the upper bound: BB guides the search into the convergence basins of increasingly lower local minima and GMA refines the bound by jumping to the nearest local minimum. Discontinuities in the lower bound occur when an entire sub-cube level has been explored. With batch initialisation, the global minimum is generally captured at the start of the algorithm. The remaining time is spent increasing the lower bound until ϵ-optimality can be guaranteed. While sometimes slower for simpler datasets or larger ϵ, it usually reduces runtime and is the preferred setting.

6.2. Fully-Overlapping Registration

In these experiments, we evaluated the performance of GOGMA by aligning single-view partial scans with a full 3D model. We used the DRAGON dataset, consisting of one reconstructed model (DRAGON-RECON) and 15 partial scans (DRAGON-STAND). The 72 base ISOI rotations were used as the initial transformations for the partial scans. For the standard parameter settings, GOGMA found the correct alignment in all 1080 cases, as shown in Table 1 (SVGM).

To investigate the effect of other GMM types on the accuracy and runtime of the algorithm, we repeated the experiment with fixed-bandwidth Kernel Density Estimation (KDE) [25] and Expectation Maximisation (EM) [15]. The number of components was fixed ($m, n = 50$), but the variances and mixture weights were set by the algorithms. For KDE, the variance was found by parameter search and the point-sets were randomly downsampled to m points. The results were poor, due to the small number of components imposed by GOGMA for tractability. The EM results show that it is a suitable input to GOGMA in terms of alignment



(a) Scan (in red) aligned to model (b) Runtime versus m, n

(c) Runtime versus ϵ (d) Runtime versus lower bound

Figure 5. Mean runtime of GOGMA on the DRAGON dataset with respect to different factors, for the alignment of DRAGON-RECON with point-set 0 of DRAGON-STAND, transformed by 72 uniformly distributed rotations. Note the logarithmic scale in (c).

accuracy, however the implementation [18] took 4 663s to process the model and ~256s to process each scan, making it impractical unless more efficiently implemented. Considering both speed and accuracy, SVGMs are recommended.

To investigate the effect of other factors on the runtime, one was varied while the others were kept at the defaults. The 72 ISOI rotations were applied to scan 0 and the mean runtimes were reported for standard and batch initialisations. The scan, aligned by GOGMA, is shown in Figure 5a in red. The results for differing numbers of Gaussian components m, n are shown in Figure 5b. The quadratic shape reflects the $O(mn)$ per-iteration complexity. The results for differing values of the convergence threshold ϵ are shown in Figure 5c. For values of ϵ close to zero, the runtime increases steeply, while larger values allow the algorithm to terminate quicker, albeit with a looser optimality guarantee. We found that $\epsilon = 0.1$ was a suitable default, having a 100% success rate. The runtime is also affected by the quality of the lower bound, as shown in Figure 5d. The GOGMA lower bound is tighter than the Go-ICP lower bound [54], which uses the distance to an uncertainty sphere containing the spherical cap, as reflected in the runtimes.

6.3. Partially-Overlapping Registration

To evaluate the performance of GOGMA on large-scale field datasets, we used the two challenging laser scanner datasets in Table 2 [38]. STAIRS is a structured in/outdoor dataset with large and rapid variations in scanned volumes. WOOD-SUMMER is an unstructured outdoor dataset with dynamic objects. The symmetric inlier fraction was used to calculate the overlap: the fraction of points from the joint

Table 2. Characteristics of the large-scale field datasets [38].

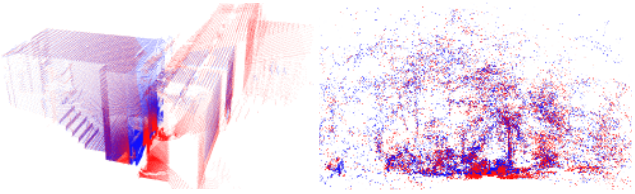
Dataset	STAIRS	WOOD-SUMMER
# Point-Sets	31	37
Mean # Points	191 000	182 000
Mean Overlap	76%	77%

Table 3. Alignment results for STAIRS. The mean translation error (in metres), rotation error (in degrees), coarse (C), medium (M) and fine (F) registration inlier rates (defined in the text) and mean runtime (in seconds) are reported. GOGMA is denoted by [*], GOGMA with refinement by $[*]_+$, Go-ICP with $\epsilon = 10^{-4}$ by $[54]_a$, Go-ICP with $\epsilon = 5 \times 10^{-5}$ by $[54]_b$, ICP by [4] and CPD by [33].

Method	[*]	$[*]_+$	$[54]_a$	$[54]_b$	[4]	[33]
Translation	0.26	**0.04**	1.63	1.17	4.67	5.24
Rotation	1.25	**0.32**	30.9	19.4	107	88.8
Inlier % (C)	**100**	**100**	71.8	80.9	15.5	38.8
Inlier % (M)	**100**	**100**	48.5	51.9	13.4	28.6
Inlier % (F)	80.0	**99.7**	19.6	21.2	6.5	7.1
Runtime	49.6	71.2	31.6	103	**0.38**	4.2

Table 4. Alignment results for WOOD-SUMMER.

Method	[*]	$[*]_+$	$[54]_a$	$[54]_b$	[4]	[33]
Translation	0.72	**0.13**	1.33	0.69	7.37	8.13
Rotation	3.09	**0.68**	9.66	5.19	109	90.7
Inlier % (C)	**100**	**100**	78.2	84.1	11.3	39.5
Inlier % (M)	75.0	**99.9**	36.6	64.5	10.8	19.3
Inlier % (F)	16.7	**99.9**	13.2	27.5	5.4	0.8
Runtime	29.5	49.6	26.2	77.7	**0.44**	4.2



(a) STAIRS point-sets 4 & 5 (b) WOOD-SUMMER point-sets 0 & 1

Figure 6. Qualitative results for two large-scale datasets. The blue scan was aligned by GOGMA from an arbitrary initial pose against the red scan, followed by GMA refinement. Best viewed in colour.

set within $10\bar{d}$ of a point from the other point-set, where $\bar{d}$ is the mean closest point distance. Sequential point-sets were aligned using GOGMA with/out refinement, Go-ICP [54], ICP [4] and CPD [33] with the 72 ISOI rotations as initial transformations. GOGMA was refined by running GMA from the output transformation with $m, n \approx 1000$ components. The coarse, medium and fine registration inlier rates are defined as the fraction of alignments with translation and rotation errors less than 2m/10°, 1m/5° and 0.5m/2.5° respectively. Tables 3 and 4 and Figure 6 show the results.

GOGMA significantly outperformed Go-ICP in these experiments, finding the correct transformation in all cases under the coarse criterion. Crucially, the success of the refinement step in finding the correct transformation in virtually all cases under the fine criterion indicates that GOGMA found the correct alignment, up to the granularity of the 50 component representation. Go-ICP performed poorly with the loose convergence threshold $\epsilon = 10^{-4}$ and $N = 50$ points. With ϵ an order of magnitude smaller (10^{-5}), N an order of magnitude greater (500), or any trimming, the runtime became prohibitively slow. The tightest feasible ϵ (5×10^{-5}) failed to coarsely align 19% of cases for STAIRS and 16% for WOOD-SUMMER. Finally, the results show that ICP and CPD both perform poorly without a good pose prior, converging to local minima for most initialisations.

Table 5. Sensor localisation results for APARTMENT. The mean translation error (in metres), rotation error (in degrees), runtime (in seconds) and fine (F) registration inlier rates are reported.

Room Scan	A	B	C	D
Translation	0.16	0.22	0.40	0.35
Rotation	0.93	0.89	1.95	2.35
Inlier % (F)	100	100	100	100
Runtime	328	383	379	409

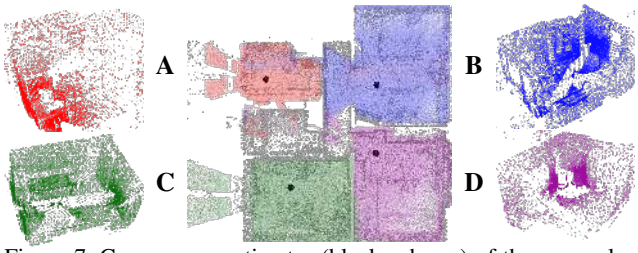


Figure 7. Coarse pose estimates (black spheres) of the sensor locations for 4 room scans (red, blue, green and purple) found by aligning each scan with the entire map (grey) using GOGMA.

A specific application is the kidnapped robot problem: finding the pose of a sensor within a 3D map. We perturbed 4 scans of different rooms in the APARTMENT dataset [38] by the 72 ISOI rotations to simulate being lost and used GOGMA to localise the scans within the map. As shown in Table 5 and Figure 7, all positions were correctly localised.

7. Conclusion

In this paper, we have introduced a globally-optimal solution to 3D Gaussian mixture alignment under the L_2 distance, with an application to point-set registration. The method applies the branch-and-bound algorithm to guarantee global optimality regardless of initialisation and uses local optimisation to accelerate convergence. The pivotal contribution is the derivation of the objective function bounds using the geometry of $SE(3)$. The algorithm performed very well on challenging field datasets, due to an objective function that is robust to outliers induced by partial-overlap and occlusion. There are several areas that warrant further investigation. Firstly, runtime benefits could be realised by implementing the local optimisation on the GPU. Dynamic branching factors would allow more parallelism for the same memory requirements. Finally, extending the lower bound to handle full covariances would enable the algorithm to be applied to more general Gaussian mixtures.

References

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

[2] J.-C. Bazin, Y. Seo, and M. Pollefeys. Globally optimal consensus set maximization through rotation search. In *Proc. Asian Conf. Comput. Vision*, pages 539–551. Springer, 2012. 2

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002. 1

[4] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992. 1, 8

[5] G. Blais and M. D. Levine. Registering multiview range data to create 3D computer objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):820–824, 1995. 1, 2

[6] A. J. P. Bustos, T.-J. Chin, and D. Suter. Fast rotation search with stereographic projections for 3d registration. In *Proc. 2014 Conf. Comput. Vision Pattern Recognition*, pages 3930–3937. IEEE, 2014. 2

[7] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Comput.*, 16(5):1190–1208, 1995. 3

[8] D. Campbell and L. Petersson. An adaptive data representation for robust point-set registration and merging. In *Proc. 2015 Int. Conf. Comput. Vision*, pages 4292–4300, Santiago, Chile, 2015. IEEE. 1, 2, 6

[9] D. Chetverikov, D. Stepanov, and P. Krsek. Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *J. Image Vision Comput.*, 23(3):299–309, 2005. 2

[10] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *Proc. 2000 Workshop Math. Methods Biomedical Image Anal.*, pages 190–197. IEEE, 2000. 1, 2

[11] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. 2000 Conf. Computer Vision and Pattern Recognition*, volume 2, pages 44–51. IEEE, 2000. 1

[12] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *J. Computer Vision Image Understanding*, 89(2):114–141, 2003. 2

[13] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):281–288, 2003. 2

[14] B. Curless and M. Levoy. Dragon, Stanford Computer Graphics Laboratory. http://graphics.stanford.edu/data/3Dscanrep/, 2014. 6

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, Jan. 1977. 2, 7

[16] T. Deselaers, G. Heigold, and H. Ney. Object classification by fusing SVMs and Gaussian mixtures. *J. Pattern Recognition*, 43(7):2476–2484, 2010. 2

[17] R. Detry, N. Pugeault, and J. H. Piater. A probabilistic framework for 3D visual object representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1790–1803, 2009. 2

[18] M. A. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):381–396, 2002. 7

[19] A. Fitzgibbon. Robust registration of 2D and 3D point sets. *J. Image Vision Comput.*, 21(13):1145–1153, 2003. 2

[20] A. Geiger, F. Moosmann, O. Car, and B. Schuster. Automatic camera and range sensor calibration using a single shot. In *Proc. 2012 Int. Conf. Robotics and Automation*, pages 3936–3943, 2012. 1

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

[22] S. Granger and X. Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *Proc. 2002 European Conf. Computer Vision*, volume 2353, pages 418–432, 2002. 2

[23] R. I. Hartley and F. Kahl. Global optimization through rotation space search. *Int. J. Comput. Vision*, 82(1):64–79, 2009. 4

[24] D. F. Huber and M. Hebert. Fully automatic registration of multiple 3D data sets. *J. Image Vision Comput.*, 21(7):637–650, 2003. 1

[25] B. Jian and B. C. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1633–1645, 2011. 1, 2, 7

[26] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999. 1, 2

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

[28] E. L. Lawler and D. E. Wood. Branch-and-Bound methods: A survey. *Operations Research*, 14(4):699–719, 1966. 3

[29] H. Li and R. Hartley. The 3D-3D registration problem revisited. In *Proc. 2007 Int. Conf. Computer Vision*, pages 1–8. IEEE, 2007. 2, 3

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

[31] N. Mellado, D. Aiger, and N. J. Mitra. Super 4PCS fast global pointcloud registration via smart indexing. *Comput. Graph. Forum*, 33(5):205–215, 2014. 2

[32] J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. *J. Numerical Analysis*, pages 105–116, 1978. 2

[33] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2262–2275, 2010. 1, 2, 8

[34] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM–3D mapping outdoor environments. *J. Field Robotics*, 24(8-9):699–722, 2007. 1

[35] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE Trans. Pattern Anal. Machine Intelligence*, 31(5):783–794, 2009. 2

[36] C. Papazov and D. Burschka. Stochastic global optimization for robust point set registration. *Computer Vision and Image Understanding*, 115(12):1598–1609, 2011. 2

[37] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013. 1, 2

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

[39] C. Robertson and R. B. Fisher. Parallel evolutionary registration of range data. *Computer Vision and Image Understanding*, 87(1):39–50, 2002. 2

[40] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. Int. Conf. 3D Digital Imaging and Modeling*, pages 145–152, 2001. 2

[41] R. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D registration. In *Proc. 2009 Int. Conf. Robotics and Automation*, pages 3212–3217, 2009. 2

[42] R. Sandhu, S. Dambreville, and A. Tannenbaum. Point set registration via particle filtering and stochastic dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1459–1473, 2010. 2

[43] D. W. Scott. Parametric statistical modeling by minimum integrated square error. *Technometrics*, 43(3):274–285, 2001. 2

[44] L. Silva, O. R. P. Bellon, and K. L. Boyer. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(5):762–776, 2005. 2

[45] T. D. Stoyanov, M. Magnusson, H. Andreasson, and A. Lilienthal. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *Int. J. Robot. Res.*, 2012. 2

[46] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graphics*, 19(7):1199–1217, 2013. 2

[47] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. *Proc. 2004 European Conf. Computer Vision*, pages 558–569, 2004. 1, 2

[48] G. Turk and M. Levoy. Stanford Bunny, Stanford Computer Graphics Laboratory. http://graphics.stanford.edu/data/3Dscanrep/, 2014. 6

[49] VXL. VXL 1.14.0: C++ Libraries for Computer Vision. http://vxl.sourceforge.net/, 2014. 6

[50] O. J. Woodford, M.-T. Pham, A. Maki, F. Perbet, and B. Stenger. Demisting the hough transform for 3D shape recognition and registration. *Int. J. Comput. Vision*, 106(3):332–341, 2014. 2

[51] H. Xiong, S. Szedmak, and J. Piater. A study of point cloud registration with probability product kernel functions. In *Proc. 2013 Int. Conf. 3D Vision*, pages 207–214, Seattle, USA, 2013. IEEE. 2

[52] J. Yang, Y. Dai, H. Li, H. Gardner, and Y. Jia. Single-shot extrinsic calibration of a generically configured RGB-D camera rig from scene constraints. In *Proc. Int. Symp. Mixed and Augmented Reality*, pages 181–188, Adelaide, Australia, 2013. 1

[53] 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.*, PP(99):1–1, 2016. 1, 2, 4

[54] J. Yang, H. Li, and Y. Jia. GoICP: Solving 3D registration efficiently and globally optimally. In *Proc. 2013 Int. Conf. Comput. Vision*, pages 1457–1464, Sydney, Australia, 2013. IEEE. 1, 2, 3, 5, 7, 8

[55] A. Yershova, S. Jain, S. M. Lavalle, and J. C. Mitchell. Generating uniform incremental grids on SO(3) using the Hopf fibration. *Int. J. Robot. Res.*, 2009. 6

[56] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Computer Vision*, 13(2):119–152, 1994. 1, 2