An Adaptive Data Representation for Robust Point-Set Registration and Merging

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

Abstract

This paper presents a framework for rigid point-set registration and merging using a robust continuous data representation. Our point-set representation is constructed by training a one-class support vector machine with a Gaussian radial basis function kernel and subsequently approximating the output function with a Gaussian mixture model. We leverage the representation's sparse parametrisation and robustness to noise, outliers and occlusions in an efficient registration algorithm that minimises the L_2 distance between our support vector–parametrised Gaussian mixtures. In contrast, existing techniques, such as Iterative Closest Point and Gaussian mixture approaches, manifest a narrower region of convergence and are less robust to occlusions and missing data, as demonstrated in the evaluation on a range of 2D and 3D datasets. Finally, we present a novel algorithm, GMMerge, that parsimoniously and equitably merges aligned mixture models, allowing the framework to be used for reconstruction and mapping.

1. Introduction

Point-set registration, the problem of finding the transformation that best aligns one point-set with another, is fundamental in computer vision, robotics, computer graphics and medical imaging. A general-purpose point-set registration algorithm operates on unstructured point-sets and may not assume other information is available, such as labels or mesh structure. Applications include merging multiple partial scans into a complete model [16]; using registration results as fitness scores for object recognition [2]; registering a view into a global coordinate system for sensor localisation [22]; and finding relative poses between sensors [36].

The dominant solution is the Iterative Closest Point (ICP) algorithm [3] and variants due to its conceptual simplicity, usability and good performance in practice. However, these are local techniques that are very susceptible to local minima and outliers and require a significant amount of overlap between point-sets. To mitigate the problem of local minima, other solutions have widened the region of convergence [14], performed heuristic global search [25], used feature-based coarse alignment [24] or used branch-and-bound techniques to find the global minimum [37].

Our method widens the region of convergence and is robust to occlusions and missing data, such as those arising when an object is viewed from different locations. The central idea is that the robustness of registration is dependent on the data representation used. We present a framework for robust point-set registration and merging using a continuous data representation, a Support Vector–parametrised Gaussian Mixture (SVGM). A discrete point-set is mapped to the continuous domain by training a Support Vector Machine (SVM) and mapping it to a Gaussian Mixture Model (GMM). Since an SVM is parametrised by a sparse intelligently-selected subset of data points, an SVGM is compact and robust to noise, fragmentation and occlusions [33], crucial qualities for efficient and robust registration. The motivation for a continuous representation is that a typical scene comprises a single, seldom-disjoint continuous surface, which cannot be fully modelled by a discrete point-set sampled from the scene.

Our Support Vector Registration (SVR) algorithm minimises an objective function based on the L_2 distance between SVGMs. Unlike the benchmark GMM registration algorithm GMMReg [17], SVR uses an adaptive and sparse representation with non-uniform and data-driven mixture weights, enabling faster performance and improving the robustness to outliers, occlusions and partial overlap.

Finally, we propose a novel merging algorithm, GMMerge, that parsimoniously and equitably merges aligned mixtures. Merging SVGM representations is useful for applications where each point-set may contain unique information, such as reconstruction and mapping. Our registration and merging framework is visualised in Figure 1.

2. Related Work

The large volume of work published on ICP, its variants and other registration techniques precludes a comprehensive list, however the reader is directed to recent surveys on ICP variants [23] and 3D point-set and mesh registration

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

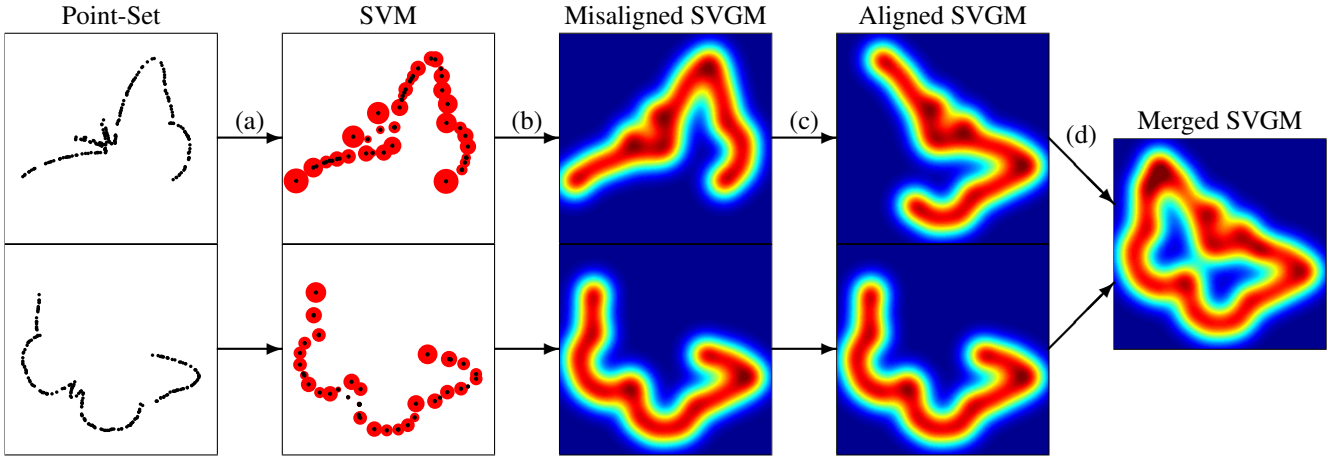


Figure 1. Robust point-set registration and merging framework. An nD point-set is represented as an SVGM by training a one-class SVM (a) and then mapping it to a GMM (b). The SVR algorithm is used to minimise the L_2 distance between two SVGMs in order to align the densities (c). Finally, the GMMerge algorithm is used to parsimoniously fuse the two mixtures. The SVMs are visualised as support vector points scaled by mixture weight and the SVGMs are coloured by probability value. Best viewed in colour.

techniques [31] for additional background. Of relevance to our work are extensions that improve its occlusion robustness, such as trimming [6]. Local methods that seek to improve upon ICP's basin of convergence and sensitivity to outliers include LM-ICP [14], which uses a distance transform to optimise the ICP error without establishing explicit point correspondences.

Another family of approaches, to which ours belongs, is based on the Gaussian Mixture Model (GMM) and show an improved robustness to poor initialisations, noise and outliers. Notable GMM algorithms for rigid and non-rigid registration include Robust Point Matching [7], using soft assignment and deterministic annealing, Coherent Point Drift [21], Kernel Correlation [32] and GMMReg [17]. The latter two do not establish explicit point correspondences and both minimise a distance measure between mixtures. GMMReg [17] defines an equally-weighted Gaussian at every point in the set with identical and isotropic covariances and minimises the L_2 distance between mixtures. The Normal Distributions Transform (NDT) algorithm [19] is a similar method, defining Gaussians for every cell in a grid discretisation and estimating full data-driven covariances, like [34]. Unlike our method, however, it imposes external structure on the scene and uses uniform mixture weights.

In contrast, globally-optimal techniques avoid local minima by searching the entire transformation space. Existing 3D methods [18, 37] are often very slow or make restrictive assumptions about the point-sets or transformations. There are also many heuristic or stochastic methods for global alignment that are not guaranteed to converge, such as particle filtering [25], genetic algorithms [29] and feature-based alignment [24]. A recent example is SUPER 4PCS, a four-points congruent sets method that exploits a clever data structure to achieve linear-time performance [20].

The rest of the paper is organised as follows: we present the SVGM representation, its properties and implementation in Section 3, we develop a robust framework for SVGM registration in Section 4, we propose an algorithm for merging SVGMs in Section 5, we experimentally demonstrate the framework's effectiveness in Section 6 and we discuss the results and conclude in Sections 7 and 8.

3. Adaptive Point-Set Representation

A central idea of our work is that the robustness of point-set registration is dependent on the data representation used. Robustness to occlusions or missing data, more so than noise, is of primary concern, because point-sets rarely overlap completely, such as when an object is sampled from a different sensor location. Another consideration is the class of optimisation problem a particular representation admits. Framing registration as a continuous optimisation problem involving continuous density functions may make it more tractable than the equivalent discrete problem [17]. Consequently, we represent discrete point-sets with Gaussian Mixture Models (GMMs). Crucially, we first train a Support Vector Machine (SVM) and then transform this into a GMM. Since the output function of the SVM only involves a sparse subset of the data points, the representation is compact and robust to noise, fragmentation and occlusions [33], attributes that persist through the GMM transformation.

3.1. One-Class Support Vector Machine

The output function of an SVM can be used to approximate the surface described by noisy and incomplete point-set data, providing a continuous implicit surface representation. Nguyen and Porikli [33] demonstrated that this representation is robust to noise, fragmentation, missing data and other artefacts for 2D shapes, with the same behaviour

expected in 3D. An SVM classifies data by constructing a hyperplane that separates data of two different classes, maximising the margin between the classes while allowing for some mislabelling [10]. Since point-set data contains only positive examples, one-class SVM [26] can be used to find the hyperplane that maximally separates the data points from the origin or viewpoint in feature space. The training data is mapped to a higher-dimensional feature space, where it may be linearly separable from the origin, with a non-linear kernel function.

The output function $f(\mathbf{x})$ of one-class SVM is given by

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho \tag{1}$$

where $\mathbf{x}_i$ are the point vectors, α_i are the weights, $\mathbf{x}$ is the input vector, ρ is the bias, ℓ is the number of training samples and K is the kernel function that evaluates the inner product of data vectors mapped to a feature space. We use a Gaussian Radial Basis Function (RBF) kernel

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}\right\|_2^2\right) \tag{2}$$

where γ is the Gaussian kernel width.

The optimisation formulation in [26] has a parameter $\nu \in (0, 1]$ that controls the trade-off between training error and model complexity. It is a lower bound on the fraction of support vectors and an upper bound on the misclassification rate [26]. The data points with non-zero weights α_i^{SV} are the support vectors $\mathbf{x}_i^{\mathrm{SV}} \in \{\mathbf{x}_i : \alpha_i > 0, i = 1, \ldots, \ell\}$.

We estimate the kernel width γ automatically for each point-set by noting that it is inversely proportional to the square of the scale σ. For an $\ell \times D$ point-set $\mathbf{X}$ with mean $\bar{\mathbf{x}}$, the estimated scale $\hat{\sigma}$ is proportional to the $2D$th root of the generalised variance

$$\hat{\sigma} \propto \left| \frac{1}{\ell - 1} (\mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^{\mathsf{T}})^{\mathsf{T}} (\mathbf{X} - \mathbf{1}\bar{\mathbf{x}}^{\mathsf{T}}) \right|^{1/2D}. \tag{3}$$

If a training set is available, better performance can be achieved by finding γ using cross-validation, imposing a constraint on the registration accuracy and searching in the neighbourhood of $1/2\hat{\sigma}^2$.

3.2. Gaussian Mixture Model Transformation

In order to make use of the trained SVM for point-set registration, it must first be approximated as a GMM. We use the transformation identified by Deselaers *et al.* [12] to represent the SVM in the framework of a GMM, without altering the decision boundary. A GMM converted from an SVM will necessarily optimise classification performance instead of data representation, since SVMs are discriminative models, unlike standard generative GMMs. This allows it to discard redundant data and reduces its susceptibility to varying point densities, which are prevalent in real datasets.

The decision function of an SVM with a Gaussian RBF kernel can be written as

$$r(\mathbf{x}) = \underset{k \in \{-1, 1\}}{\arg\max} \left\{ \sum_{i=1}^{\ell^{\mathrm{SV}}} k \alpha_i^{\mathrm{SV}} \mathrm{e}^{-\gamma \left\|\mathbf{x}_i^{\mathrm{SV}} - \mathbf{x}\right\|_2^2} - k\rho \right\} \tag{4}$$

where ℓ^{SV} is the number of support vectors and k is the class, positive for inliers and negative otherwise for one-class SVM. The GMM decision function can be written as

$$r'(\mathbf{x}) = \underset{k \in \{-1, 1\}}{\arg\max} \left\{ \sum_{i=1}^{I_k} p(k) p(i|k) \mathcal{N}\left(\mathbf{x} \middle| \boldsymbol{\mu}_{ki}, \sigma_k^2\right) \right\} \tag{5}$$

where I_k is the number of clusters for class k, $p(k)$ is the prior probability of class k, $p(i|k)$ is the cluster weight of the ith cluster of class k and $\mathcal{N}\left(\mathbf{x} \middle| \boldsymbol{\mu}_{ki}, \sigma_k^2\right)$ is the Gaussian representing the ith cluster of class k with mean $\boldsymbol{\mu}_{ki}$ and variance σ_k^2, given by

$$\mathcal{N}\left(\mathbf{x} \middle| \boldsymbol{\mu}_{ki}, \sigma_k^2\right) = \frac{1}{(2\pi\sigma_k^2)^{D/2}} \exp\left(-\frac{\left\|\mathbf{x} - \boldsymbol{\mu}_{ki}\right\|_2^2}{2\sigma_k^2}\right). \tag{6}$$

Noting the similarity of (4) and (5), the mapping

$$\boldsymbol{\mu}_{ki} = \begin{cases} \mathbf{x}_i^{\mathrm{SV}} & \text{if } k = +1 \\ \mathbf{0} & \text{else} \end{cases} \tag{7}$$

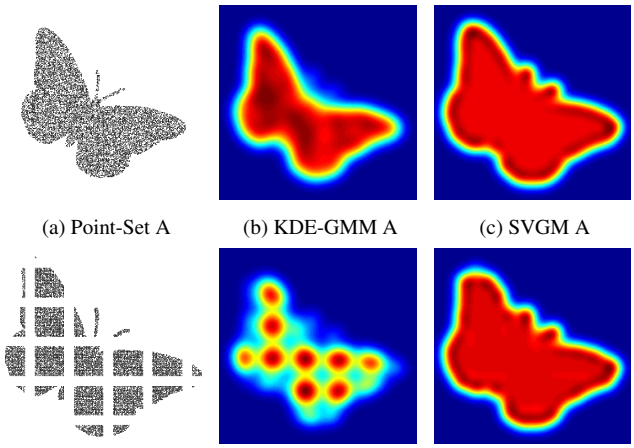
$$\sigma_k^2 = \begin{cases} 1/2\gamma & \text{if } k = +1 \\ N_\infty & \text{else} \end{cases} \tag{8}$$

$$\phi_i = p(k) p(i|k) = \begin{cases} \alpha_i^{\mathrm{SV}} (2\pi\sigma_k^2)^{D/2} & \text{if } k = +1 \\ \rho (2\pi\sigma_k^2)^{D/2} & \text{else} \end{cases} \tag{9}$$

can be applied, where ϕ_i is the mixture weight, that is, the prior probability of the ith component. The bias term ρ is approximated by an additional density given to the negative class with arbitrary mean, very high variance N_∞ and a cluster weight proportional to ρ. We omit this term from the registration framework because it does not affect the optimisation. The resulting GMM is parametrised by

$$\mathcal{G} = \left\{\boldsymbol{\mu}_i,\ \sigma^2,\ \phi_i\right\}_{i=1}^{\ell^{\mathrm{SV}}}. \tag{10}$$

While we transform an SVM into a GMM, there are many other ways to construct a GMM from point-set data. Kernel Density Estimation (KDE) with identically-weighted Gaussian densities has frequently been used for this purpose, including fixed-bandwidth KDE with isotropic covariances [17, 13], variable-bandwidth KDE with non-identical covariances [9] and non-isotropic covariance KDE [34]. The primary disadvantage of these methods is that the number of Gaussian components is equal to the point-set size, which can be very large for real-world datasets. In contrast, our work intelligently selects a sparse subset of



(a) Point-Set A (b) KDE-GMM A (c) SVGM A

(d) Point-Set B (e) KDE-GMM B (f) SVGM B

Figure 2. The effect of significant occlusion on two point-set representations, using the same parameters for both. Our SVGM representation is, qualitatively, almost identical when occluded (f) and unoccluded (c), whereas the fixed-bandwidth KDE representation is much less robust to occlusion (e). Best viewed in colour.

the data points to locate the Gaussian densities and weights them non-identically, making it more robust to occlusions and missing data, as demonstrated in Figure 2.

Expectation Maximisation (EM) [11] can also be used to construct a GMM with fewer components than KDE. EM finds the maximum likelihood estimates of the GMM parameters, where the number of densities is specified a priori, unlike our method. To initialise the algorithm, the means can be chosen at random or using the k-means algorithm; or, an initial Gaussian can be iteratively split and re-estimated until the number of densities is reached [12]. However, deliberately inflating the number of components can be slow and sensitive to initialisation [28, p. 326].

4. Support Vector Registration

Once the point-sets are in mixture model form, the registration problem can be posed as minimising the distance between mixtures. Like Jian and Vemuri [17], we use the L_2 distance, which can be expressed in closed-form. The L_2E estimator minimises the L_2 distance between densities and is known, counter-intuitively, to be inherently robust to outliers [27], unlike the maximum likelihood estimator that minimises the Kullback-Leibler divergence.

Let $\mathcal{X}$ be the moving model point-set, $\mathcal{Y}$ be the fixed scene point-set, $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$ be GMMs converted from SVMs trained on $\mathcal{X}$ and $\mathcal{Y}$ respectively, and $T(\mathcal{G}, \boldsymbol{\theta})$ be the transformation model parametrised by $\boldsymbol{\theta}$. The L_2 distance between transformed $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$ is given by

$$D_{L_2}(\mathcal{G}_\mathcal{X}, \mathcal{G}_\mathcal{Y}, \boldsymbol{\theta}) = \int_{\mathbb{R}^D} \left(p\left(\mathbf{x}|T(\mathcal{G}_\mathcal{X}, \boldsymbol{\theta})\right) - p\left(\mathbf{x}|\mathcal{G}_\mathcal{Y}\right)\right)^2 \, \mathrm{d}\mathbf{x} \tag{11}$$

where $p\left(\mathbf{x}|\mathcal{G}\right)$ is the probability of observing a point $\mathbf{x}$ given a mixture model $\mathcal{G}$ with ℓ components, that is

$$p\left(\mathbf{x}|\mathcal{G}\right) = \sum_{i=1}^{\ell} \phi_i \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_i, \sigma^2\right). \tag{12}$$

Expanding (11), the last term is independent of $\boldsymbol{\theta}$ and the first term is invariant under rigid transformations. Both are therefore removed from the objective function. The middle term is the inner product of two Gaussian mixtures and has a closed form that can be derived by applying the identity

$$\begin{aligned}\int_{\mathbb{R}^D} \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_1, \sigma_1^2\right) \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_2, \sigma_2^2\right) \, \mathrm{d}\mathbf{x} \\ = \mathcal{N}\left(\mathbf{0}|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2, \sigma_1^2 + \sigma_2^2\right).\end{aligned} \tag{13}$$

Therefore, noting that $\sigma_\mathcal{X}^2 = \sigma_\mathcal{Y}^2$ in our formulation, the objective function for rigid registration is defined as

$$f\left(\boldsymbol{\theta}\right) = -\sum_{i=1}^{m}\sum_{j=1}^{n} \phi_{i,\mathcal{X}}\phi_{j,\mathcal{Y}}\mathcal{N}\left(\mathbf{0}|\boldsymbol{\mu}'_{i,\mathcal{X}} - \boldsymbol{\mu}_{j,\mathcal{Y}}, 2\sigma^2\right) \tag{14}$$

where m and n are the number of components in $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$ respectively and $\boldsymbol{\mu}'_{i,\mathcal{X}} = T(\boldsymbol{\mu}_{i,\mathcal{X}}, \boldsymbol{\theta})$. This can be expressed in the form of a discrete Gauss transform, which has a computational complexity of $\mathcal{O}(mn)$, or the fast Gauss transform [15], which scales as $\mathcal{O}(m+n)$.

The gradient vector is derived as in [17]. Let $\mathbf{M}_0 = \left[\boldsymbol{\mu}_{1,\mathcal{X}}, \ldots, \boldsymbol{\mu}_{m,\mathcal{X}}\right]^\intercal$ be the $m \times D$ matrix of the means from $\mathcal{G}_\mathcal{X}$ and $\mathbf{M} = T(\mathbf{M}_0, \boldsymbol{\theta})$ be the transformed matrix, parametrised by $\boldsymbol{\theta}$. Using the chain rule, the gradient is $\frac{\partial f}{\partial \boldsymbol{\theta}} = \frac{\partial f}{\partial \mathbf{M}} \frac{\partial \mathbf{M}}{\partial \boldsymbol{\theta}}$. Let $\mathbf{G} = \frac{\partial f}{\partial \mathbf{M}}$ be an $m \times D$ matrix, which can be found while evaluating the objective function by

$$\mathbf{G}_i = -\frac{1}{2\sigma^2}\sum_{j=1}^{m} f_{ij}\left(\boldsymbol{\mu}'_{i,\mathcal{X}} - \boldsymbol{\mu}_{j,\mathcal{Y}}\right) \tag{15}$$

where $\mathbf{G}_i$ is the ith row of $\mathbf{G}$ and f_{ij} is a summand of f. For rigid motion, $\mathbf{M} = \mathbf{M}_0\mathbf{R}^\intercal + \mathbf{t}$ where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector. The gradients with respect to each motion parameter are given by

$$\frac{\partial f}{\partial \mathbf{t}} = \mathbf{G}^\intercal \mathbf{1}_m \tag{16}$$

$$\frac{\partial f}{\partial r_i} = \mathbf{1}_D^\intercal \left((\mathbf{G}^\intercal \mathbf{M}_0) \circ \frac{\partial \mathbf{R}}{\partial r_i} \right) \mathbf{1}_D \tag{17}$$

where $\mathbf{1}_i$ is the i-dimensional column vector of ones, $\circ$ is the Hadamard product and r_i are the elements parametrising $\mathbf{R}$: rotation angle α for 2D and a unit quaternion for 3D. For the latter, the quaternion is projected back to the space of valid rotations after each update by normalisation.

Since the objective function is smooth, differentiable and convex in the neighbourhood of the optimal motion parameters, gradient-based numerical optimisation methods can be

used, such as nonlinear conjugate gradient or quasi-Newton methods. We use an interior-reflective Newton method [8] since it is time and memory efficient and scales well. However, since the objective function is non-convex over the search space, this approach is susceptible to local minima, particularly for large motions and point-sets with symmetries. A multi-resolution approach can be adopted, increasing γ at each iteration and initialising with the currently optimal transformation. SVR is outlined in Algorithm 1.

Algorithm 1 Support Vector Registration (SVR): A robust algorithm for point-set registration using one-class SVM

Input: model point-set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{\ell_\mathcal{X}}$, scene point-set $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^{\ell_\mathcal{Y}}$, transformation model T parametrised by $\boldsymbol{\theta}$, initial parameter $\boldsymbol{\theta}_0$ such as the identity transformation

Output: locally optimal transformation parameter $\boldsymbol{\theta}^*$ such that $T(\mathcal{X}, \boldsymbol{\theta}^*)$ is best aligned with $\mathcal{Y}$

Select ν and γ by estimation or cross-validation
Initialise transformation parameter: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$
repeat
  Train SVMs:
  $\mathcal{S}_\mathcal{X} = \left\{\mathbf{x}_i^{\mathrm{SV}},\ \alpha_{i,\mathcal{X}}^{\mathrm{SV}}\right\}_{i=1}^{m} \leftarrow \mathrm{trainSVM}(\mathcal{X}, \nu, \gamma)$
  $\mathcal{S}_\mathcal{Y} = \left\{\mathbf{y}_i^{\mathrm{SV}},\ \alpha_{i,\mathcal{Y}}^{\mathrm{SV}}\right\}_{i=1}^{n} \leftarrow \mathrm{trainSVM}(\mathcal{Y}, \nu, \gamma)$
  Convert SVMs to GMMs using (7), (8) and (9):
  $\mathcal{G}_\mathcal{X} = \left\{\boldsymbol{\mu}_{i,\mathcal{X}},\ \sigma^2,\ \phi_{i,\mathcal{X}}\right\}_{i=1}^{m} \leftarrow \mathrm{toGMM}(\mathcal{S}_\mathcal{X}, \gamma)$
  $\mathcal{G}_\mathcal{Y} = \left\{\boldsymbol{\mu}_{i,\mathcal{Y}},\ \sigma^2,\ \phi_{i,\mathcal{Y}}\right\}_{i=1}^{n} \leftarrow \mathrm{toGMM}(\mathcal{S}_\mathcal{Y}, \gamma)$
  Optimise the objective function f (14) using the gradient (16), (17) with a trust region algorithm
  Update the parameter $\boldsymbol{\theta} \leftarrow \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$
  Anneal: $\gamma \leftarrow \delta\gamma$
until change in f or iteration number meets a condition

5. Merging Gaussian Mixtures

For an SVGM to be useful for applications where each point-set may contain unique information, such as mapping, an efficient method of merging two aligned mixtures is desirable. A naïve approach is to use a weighted sum of the Gaussian mixtures [12], however, this would result in an unnecessarily high number of components with substantial redundancy. Importantly, the probability of regions not observed in both point-sets would decrease, meaning that regions that are often occluded would disappear from the model as more mixtures were merged. While the time-consuming process of sampling the combined mixture and re-estimating it with EM would eliminate redundancy, it would not alleviate the missing data problem. The same applies to faster sample-free variational-Bayes approaches [4]. Sampling (or merging the point-sets) and re-estimating an SVGM would circumvent this problem, since the discriminative framework of the SVM is insensitive to higher-density overlapping regions, but this is not time efficient.

Algorithm 2 outlines GMMerge, our efficient algorithm for parsimoniously approximating the merged mixture without weighting the intersection regions disproportionately. Each density of $\mathcal{G}_\mathcal{X}$ is re-weighted using a sparsity-inducing piecewise linear function. The parameter $t \in [0, \infty)$ controls how many densities are added. For $t = 0$, $\mathcal{G}_{\mathcal{XY}}$ contains only $\mathcal{G}_\mathcal{Y}$. As $t \to \infty$, $\mathcal{G}_{\mathcal{XY}}$ additionally contains every non-redundant density from $\mathcal{G}_\mathcal{X}$. Figure 3 shows the SVGM representations of two 2D point-sets, the naïvely merged mixture and the GMMerge mixture.

Algorithm 2 GMMerge: An algorithm for parsimonious Gaussian mixture merging

Input: aligned mixture models with unknown overlap $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$, parametrised by means $\boldsymbol{\mu}$, variances σ^2 and mixture weights ϕ, and merging parameter t

Output: merged model $\mathcal{G}_{\mathcal{XY}}$

Initialise merged model: $\mathcal{G}_{\mathcal{XY}} \leftarrow \mathcal{G}_\mathcal{Y}$
for $i = 1 \ldots m$ **do**
  For the ith density of $\mathcal{G}_\mathcal{X}$, calculate:
  $\Delta = p\left(\boldsymbol{\mu}_{i,\mathcal{X}} \middle| \mathcal{G}_{i,\mathcal{X}}\right) - p\left(\boldsymbol{\mu}_{i,\mathcal{X}} \middle| \mathcal{G}_\mathcal{Y}\right)$
  Update weight using sparsity-inducing function:
  $\phi_{i,\mathcal{X}} \leftarrow \phi_{i,\mathcal{X}} \max\left(0, \min\left(1, t\Delta\right)\right)$
  if $\phi_{i,\mathcal{X}} > 0$ **then**
    Add to merged mixture: $\mathcal{G}_{\mathcal{XY}} \leftarrow \mathcal{G}_{i,\mathcal{X}} \cdot \mathcal{G}_{\mathcal{XY}}$
  end if
end for
Renormalise $\mathcal{G}_{\mathcal{XY}}$

6. Experimental Results

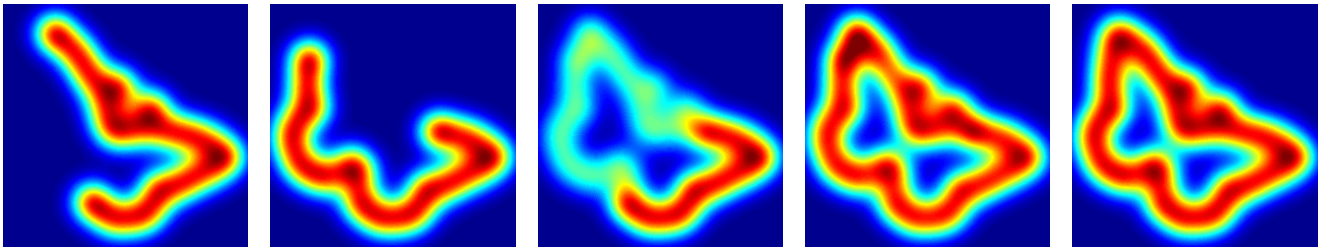
SVR was tested using many different point-sets, including synthetic and real datasets in 2D and 3D, at a range of motion scales and outlier, noise and occlusion fractions. In all experiments, the initial transformation parameter $\boldsymbol{\theta}$ was the identity, ν was 0.01 and γ was selected by cross-validation, except where otherwise noted. For all benchmark methods, parameters were chosen using a grid search.

6.1. 2D Registration

To test the efficacy of SVR for 2D registration, the four point-sets in Figure 4 were used: ROAD[1], CONTOUR, FISH and GLYPH[2]. Three benchmark algorithms were chosen: Gaussian Mixture Model Registration (abbreviated to GMR) [17], Coherent Point Drift (CPD) [21] and Iterative Closest Point (ICP) [3]. Annealing was applied for both SVR ($\delta = 10$) and GMR. Note that the advantages of SVR manifest themselves more clearly on denser point-sets.

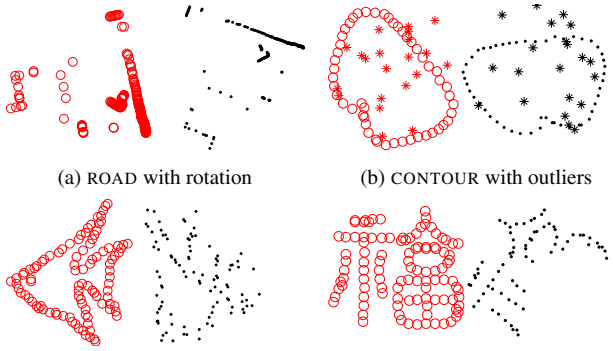
[1]Point-set from Tsin and Kanade [32], available at `http://www.cs.cmu.edu/~ytsin/KCReg/KCReg.zip`

[2]Point-sets from Chui and Rangarajan [7], available at `http://cise.ufl.edu/~anand/students/chui/rpm/TPS-RPM.zip`



(a) Aligned mixture $\mathcal{G}_{\mathcal{X}}$ (b) Aligned mixture $\mathcal{G}_{\mathcal{Y}}$ (c) Naïve merge (d) GMMerge (e) Ground truth merge

Figure 3. Merging Gaussian mixtures (a) and (b) with a naïve weighted sum (c) and GMMerge (d). The mixture produced by GMMerge is almost identical to the ground truth (e), while the naïve approach over-emphasises overlapping regions. Best viewed in colour.



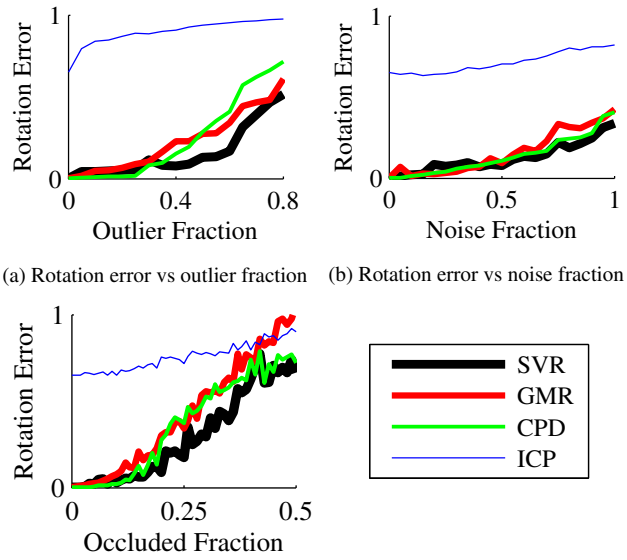
(a) ROAD with rotation (b) CONTOUR with outliers

(c) FISH with noise (d) GLYPH with occlusion

Figure 4. Sample scene (left) and model (right) point-sets from each 2D dataset, undergoing a range of perturbations.

Table 1. Convergence range (in radians). All rotation initialisations within these ranges converged (rotation error $\leq 1°$).

Point-Set	SVR	GMR	CPD	ICP
ROAD	**-3.1–3.1**	-3.0–3.0	-1.6–1.6	-0.8–0.8
CONTOUR	**-1.6–1.6**	-1.5–1.5	-1.5–1.5	-0.1–0.1
FISH	**-1.6–1.6**	-1.5–1.5	-1.2–1.3	-0.4–0.5
GLYPH	**-1.6–1.6**	**-1.6–1.6**	-1.6–1.5	-0.4–0.4



(a) Rotation error vs outlier fraction (b) Rotation error vs noise fraction

(c) Rotation error vs occluded fraction

Figure 5. Outlier, noise and occlusion results for the 2D point-sets. The mean rotation error (in radians) of 50 repetitions is reported for each and the results show that SVR is relatively robust to a large range of perturbations commonly found in real data.

The range of motions for which a correct registration result was attained was tested by rotating the model point-set by $\alpha \in [-3.14, 3.14]$ radians with a step size of 0.01. In Table 1, we report the range of contiguous initial rotations for which the algorithm converged, chosen as a rotation error $\leq 1°$. They show that SVR has a wider basin of convergence than the other methods, even for sparse point-sets.

To test the algorithm's robustness to outliers, additional points were randomly drawn from the uniform distribution and were concatenated with the model and scene point-sets separately. To avoid bias, the outliers were sampled from the minimum covering circle of the point-set. The motion was fixed to a rotation of 1 radian ($57°$) and the experiment was repeated 50 times with different outliers each time. The mean rotation error for a range of outlier fractions is shown in Figure 5a and indicates that the proposed method is more robust to outliers than the others for large outlier fractions.

To test for robustness to noise, a noise model was applied to the model point-set by adding Gaussian noise to each point sampled from the distribution $\mathcal{N}(\mathbf{0}, (\lambda\hat{\sigma})^2)$, where λ is the noise fraction and $\hat{\sigma}$ is the estimated generalised standard deviation across the entire point-set (3). A fixed rotation of 1 radian was used and the experiment was repeated 50 times, resampling each time. The average rotation error for a range of noise fractions is shown in Figure 5b and indicates that SVR is comparable to the other methods.

To test for robustness to occlusions, we selected a random seed point and removed a fraction of the model point-set using k-nearest neighbours. A fixed rotation of 1 radian was used and the experiment was repeated 50 times with different seed points. The mean rotation error for a range of occlusion fractions is shown in Figure 5c and indicates that the algorithm is more robust to occlusion than the others.

Table 2. Number of point-set pairs that converged for a range of relative poses. Mean computation time in seconds is also reported.

	Local				Global	
Pose	**SVR**	**GMR**	**CPD**	**ICP**	**GOI**	**S4P**
±24°	**30**	29	26	28	**30**	29
±48°	**29**	20	18	19	27	24
±72°	16	13	14	13	**18**	17
±96°	4	2	3	1	10	**13**
Runtime	0.2	19.2	5.7	**0.04**	1407	399

6.2. 3D Registration

The advantages of SVR are particularly apparent with dense 3D point-sets. For evaluation, we used DRAGON-STAND[3], AASS-LOOP[4] and HANNOVER2[5] and seven benchmark algorithms: GMMReg (abbreviated to GMR) [17], CPD [21], ICP [3], NDT Point-to-Distribution (NDP) [19] and NDT Distribution-to-Distribution (NDD) [30], Globally-Optimal ICP (GOI) [37] and SUPER 4PCS (S4P) [20]. Annealing was used only where indicated.

To evaluate the performance of the algorithm with respect to motion scale, we replicated the experiment in [17] using the DRAGON-STAND dataset. This contains 15 self-occluding scans of the dragon model acquired from different directions. We registered all 30 point-set pairs with a relative rotation of ±24° and repeated this for ±48°, ±72° and ±96°. As per [17], the criterion for convergence was $\hat{q} \cdot q > 0.99$, where $\hat{q}$ and q are the estimated and ground truth quaternions respectively. While γ was selected by cross-validation, using the estimate $\hat{\sigma}$ yielded a very similar result. The number of correctly converged registrations is reported in Table 2, showing that SVR has a significantly larger basin of convergence than the other local methods and is competitive with the slower global methods.

A representative sensitivity analysis is shown in Figure 6 for the DRAGON-STAND dataset. It indicates that rotation error is quite insensitive to perturbations in γ and is very insensitive to ν, justifying the choice of fixing this parameter.

To evaluate occlusion robustness, the same procedure was followed as for 2D, using the DRAGON-STAND dataset. The mean rotation error (in radians) and the fraction of correctly converged point-set pairs with respect to the fraction of occluded points is shown in Figure 7, for relative poses of ±24° and ±48°. The results show that SVR is significantly more robust to occlusion than the other methods.

Finally, we report registration results on two large real-world 3D datasets shown in Figure 8: AASS-LOOP (60 indoor point-sets with ∼13 500 points on average) and HANNOVER2 (923 outdoor point-sets with ∼10 000 points on average), after downsampling using a 0.1 m grid. Both were captured using a laser scanner and ground truth was provided. These are challenging datasets because sequential point-sets overlap incompletely and occluded regions are present. The results for registering adjacent point-sets are shown in Table 3 for AASS-LOOP and Table 4 for HANNOVER2. The ICP and annealed NDT results are reported directly from Stoyanov *et al.* [30] and we use their criteria for a successful registration (inlier): a translation error less than 0.5 m and a rotation error less than 0.2 radians. SVR outperforms the other methods by a significant margin, even more so when annealing ($\delta = 2$) is applied (SVR$^+$).

The mean computation speeds of the experiments, regardless of convergence, are reported in Tables 2, 3 and 4. All experiments were run on a PC with a 3.4 GHz Quad Core CPU and 8 GB of RAM. The SVR code is written in unoptimised MATLAB, except for a cost function in C++, and uses the LIBSVM [5] library. The benchmark-

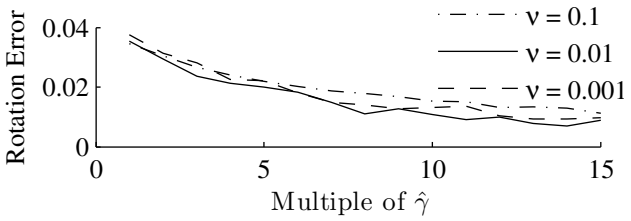
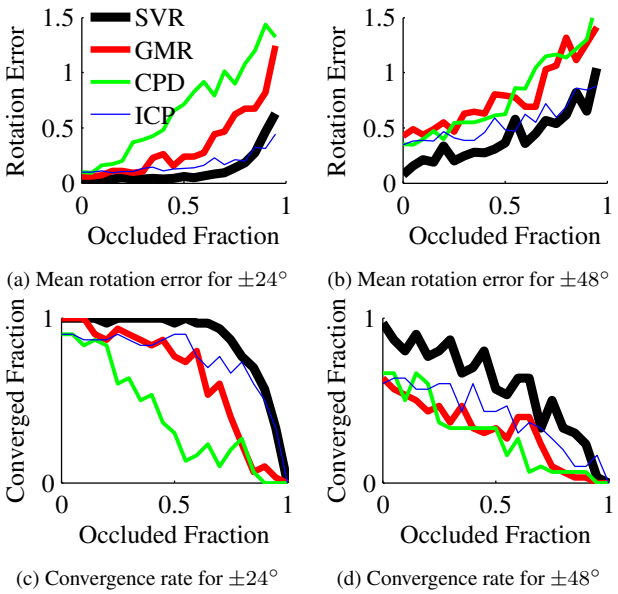


Figure 6. Sensitivity analysis for γ and ν. The median rotation error (in radians) of all DRAGON-STAND point-sets with ±24° pose differences are plotted with respect to multiples of $\hat{\gamma} = {}^1/_{2\hat{\sigma}^2}$.



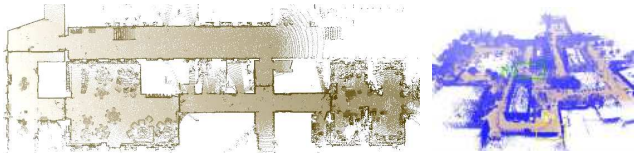
(a) Mean rotation error for ±24°
(b) Mean rotation error for ±48°
(c) Convergence rate for ±24°
(d) Convergence rate for ±48°

Figure 7. Mean rotation error (in radians) and convergence rate of all DRAGON-STAND point-sets with ±24° and ±48° pose differences, with respect to the fraction of occluded points.

[3]Point-set from Brian Curless and Marc Levoy, Stanford University, at http://graphics.stanford.edu/data/3Dscanrep/

[4]Point-set from Martin Magnusson, Örebro University, at http://kos.informatik.uni-osnabrueck.de/3Dscans/

[5]Point-set from Oliver Wulf, Leibniz University, at http://kos.informatik.uni-osnabrueck.de/3Dscans/



(a) AASS-LOOP (b) HANNOVER2

Figure 8. Two large-scale 3D datasets.

Table 3. Registration results for AASS-LOOP. While mean translation error (in metres) and rotation error (in radians) are commonly reported, the percentage of inliers (successful registrations) is a more useful metric for comparison. The mean computation time (in seconds) is also reported. SVR$^+$ is SVR with annealing.

Metric	SVR	SVR$^+$	GMR	ICP	NDP	NDD	S4P
Transl.	0.95	**0.67**	1.61	0.99	1.10	0.85	0.71
Rotation	0.08	0.06	0.12	0.04	**0.02**	0.06	0.32
Inlier %	81.4	**86.4**	18.6	55.2	50.0	63.8	78.0
Runtime	3.43	29.7	599	10.8	9.12	**1.02**	60.7

Table 4. Registration results for HANNOVER2. The mean translation error (in metres), rotation error (in radians), inlier percentage and mean runtime (in seconds) are reported. SVR$^+$ uses annealing.

Metric	SVR	SVR$^+$	GMR	ICP	NDP	NDD	S4P
Transl.	0.10	**0.09**	1.32	0.43	0.79	0.40	0.40
Rotation	**0.01**	**0.01**	0.05	0.05	0.05	0.05	0.03
Inlier %	**99.8**	**99.8**	8.88	74.4	54.2	76.4	75.0
Runtime	14.0	32.6	179	5.68	4.03	**0.51**	39.7

ing code was provided by the respective authors, except for ICP, for which a standard MATLAB implementation with k-d tree nearest-neighbour queries was used. For the DRAGON-STAND speed comparison, all point-sets were randomly downsampled to 2 000 points, because GMR, CPD, GOI and S4P were prohibitively slow for larger point-sets.

7. Discussion

The results show that SVR has a larger region of convergence than the other methods and is more robust to occlusions. This is an expected consequence of the SVGM representation, since it is demonstrably robust to missing data. In addition, the computation time results show that it scales well with point-set size, unlike GMR and CPD, largely due to the data compression property of the one-class SVM. There is a trade-off, controlled by the parameter γ, between registration accuracy and computation time.

For the application of accurate reconstruction using our framework, the one-class SVM may be replaced with a two-class SVM to better model the fine details of a scene. To generate negative class (free space) training points, surface points were displaced along their approximated normal vectors by a fixed distance d and then those points that were closer than $0.9d$ to their nearest surface point were discarded. The SVGMs constructed using this approach may be fused using GMMerge. However, for the purposes of registration, capturing fine detail in this way is unnecessary, counter-productive and much less efficient.

While SVR is a local algorithm, it can still outperform global algorithms on a number of measures, particularly speed, for certain tasks. In Section 6.2, we compared SVR with the guaranteed-optimal method Globally-Optimal ICP (GOI) [37] and the faster but not optimal method SUPER 4PCS (S4P) [20]. The motion scale results of GOI were comparable to our method, while the average runtime was four orders of magnitude longer. Note that, for point-sets with missing data or partial overlap, a globally-optimal alignment is not necessarily correct. S4P had a more favourable runtime–accuracy trade-off but was nonetheless outperformed by SVR.

8. Conclusion

In this paper, we have presented a framework for robust point-set registration and merging using a continuous data representation. Our point-set representation is constructed by training a one-class SVM and then approximating the output function with a GMM. This representation is sparse and robust to occlusions and missing data, which are crucial attributes for efficient and robust registration.

The central algorithm, SVR, outperforms state-of-the-art approaches in 2D and 3D rigid registration, exhibiting a larger basin of convergence. In particular, we have shown that it is robust to occlusion and missing data and is computationally efficient. The GMMerge algorithm complements the registration algorithm by providing a parsimonious and equitable method of merging aligned mixtures, which can subsequently be used as an input to SVR.

There are several areas that warrant further investigation. Firstly, there is significant scope for optimising the algorithm using, for example, approximations like the improved fast Gauss Transform [35] or faster optimisation algorithms that require an analytic Hessian. Secondly, non-rigid registration is a natural extension to this work and should benefit from the robustness of SVR to missing data. It may also be useful to train the SVM with full data-driven covariance matrices [1] and use the full covariances for registration [30]. Finally, methods of constructing tight bounds for an efficient branch-and-bound framework based on SVR could be investigated in order to implement a globally-optimal registration algorithm.

References

[1] S. Abe. Training of support vector machines with Mahalanobis kernels. In *Proc. 15th Int. Conf. Artificial Neural Networks*, pages 571–576. Springer-Verlag, Sept. 2005. 8

[2] 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

[3] 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, 5, 7

[4] P. Bruneau, M. Gelgon, and F. Picarougne. Parsimonious reduction of Gaussian mixture models with a variational-Bayes approach. *Pattern Recognition*, 43(3):850–858, 2010. 5

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology*, 2:27:1–27:27, 2011. 7

[6] 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

[7] 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, 5

[8] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optimization*, 6(2):418–445, 1996. 5

[9] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):281–288, 2003. 3

[10] C. Cortes and V. Vapnik. Support-vector networks. *J. Machine Learning*, 20(3):273–297, 1995. 3

[11] 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. 4

[12] T. Deselaers, G. Heigold, and H. Ney. Object classification by fusing SVMs and Gaussian mixtures. *J. Pattern Recognition*, 43(7):2476–2484, 2010. 3, 4, 5

[13] 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. 3

[14] A. Fitzgibbon. Robust registration of 2D and 3D point sets. *J. Image Vision Comput.*, 21(13):1145–1153, 2003. 1, 2

[15] L. Greengard and J. Strain. The fast Gauss transform. *SIAM J. Scientific Statistical Comput.*, 12(1):79–94, 1991. 4

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

[17] 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, 3, 4, 5, 7

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

[19] 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, 7

[20] N. Mellado, D. Aiger, and N. J. Mitra. Super 4PCS fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014. 2, 7, 8

[21] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2262–2275, 2010. 2, 5, 7

[22] 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

[23] 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

[24] 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. 1, 2

[25] 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. 1, 2

[26] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001. 3

[27] D. W. Scott. Parametric statistical modeling by minimum integrated square error. *Technometrics*, 43(3):274–285, 2001. 4

[28] D. W. Scott and W. F. Szewczyk. From kernels to mixtures. *Technometrics*, 43(3):323–335, 2001. 4

[29] 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

[30] 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. Robotics Research*, 2012. 7, 8

[31] 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

[32] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. *Proc. 2004 European Conf. Computer Vision*, pages 558–569, 2004. 2, 5

[33] H. Van Nguyen and F. Porikli. Support Vector Shape: A classifier-based shape representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(4):970–982, 2013. 1, 2

[34] 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, 3

[35] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. In *Proc. 2003 Int. Conf. Computer Vision*, pages 664–671, Nice, France, 2003. IEEE. 8

[36] 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

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