Robust Feature-Based Point Registration Using Directional Mixture Model [★]

Saman Fahandezh-Saadi [*] **Di Wang** [**] **Masayoshi Tomizuka** [*]

[*] *Mechanical Systems Control (MSC) Lab, University of California, Berkeley, USA. e-mail: {samanfahandej, tomizuka}@berkeley.edu*
[**] *Visual Cognitive Computing and Intelligent Vehicle (VCC&IV) Lab, Xi'an Jiaotong University, Xi'an, P.R. China. e-mail: de2wang@stu.xjtu.edu.cn*

Abstract: This paper presents a robust probabilistic point registration method for estimating the rigid transformation (i.e. rotation matrix and translation vector) between two pointcloud dataset. The method improves the robustness of point registration and consequently the robot localization in the presence of outliers in the pointclouds which always occurs due to occlusion, dynamic objects, and sensor errors. The framework models the point registration task based on directional statistics on a unit sphere. In particular, a Kent distribution mixture model is adopted and the process of point registration has been carried out in the two phases of Expectation-Maximization algorithm. The proposed method has been evaluated on the pointcloud dataset from LiDAR sensors in an indoor environment.

Keywords: Intelligent Autonomous Vehicles, Perception, Localization, Mobile Robots, Robust Estimation, 3D Point Registration

1. INTRODUCTION

The problem of point registration or matching plays a crucial role in many engineering applications and scientific disciplines from robot navigation, odometry and autonomous vehicles to graphics, object modeling, and medical imaging. In all of these applications, it is important to acquire a very accurate point registration, despite the sensors errors and limitations. The goal of point registration is to use 3D dataset observation and try to find the best rigid transformation hypothesis that maps one frame to the next. The rigid transformation $(\mathbf{R}, \mathbf{t}) \in SE(3)$ consists of a rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$; and a 3D point $\mathbf{x} \in \mathbb{R}^3$ can be accordingly transformed rigidly as $\mathbf{T}(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t}$.

Recently, the point registration algorithms have received extensive focus from academia and industries in autonomous driving, since the rapidly developed 3D sensing technology endows the intelligent vehicle the capacity of accurate mapping and localization. Superiority of 3D LiDAR which can reliably employed during night, bad weather (e.g. rain, snow), and in terms of computing complexity, makes it a suitable choice of 3D perception for intelligent vehicles. However, the limited sensor's coverage, intrinsic sensor errors, and outliers caused by occlusion or unpredictable traffic participators in the real traffic scenarios are formidable challenges which require to be addressed when trying to deploy the point registration algorithm in intelligent vehicles.

A standard approach for point registration problem is the Iterative Closest Point (ICP) algorithm (Besl and McKay (1992)) which solves the problem in two iterative steps: finding the correspondence points and then minimizing the sum of squared residuals to find the best transformation. The ICP method is sensitive to the outliers and the solution is heavily influenced by the initialization. There are several variants of ICP method that try to improve the original method in different regards: ICPp2pl (Low (2004)) ICPpl2pl (Serafin and Grisetti (2015)), GICP (Segal et al. (2009)). In all of these methods both rotation matrix and translation vector are found together iteratively. In addition, the correspondence phase is hard-assignment or one-to-one.

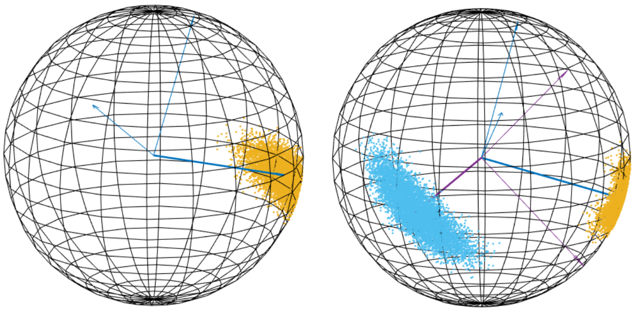

Fig. 1. Two data samples drawn from Kent distribution. **Left:** Samples drawn from a single Kent PDF. Arrows show $\boldsymbol{\mu}$ the mean direction of samples as well as $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2$ major and minor axes which depend on the orientation of samples. **Right:** Shows a mixture of two Kent distributions with different parameters. The pointcloud surface normals with different concentrations, can be modeled as a mixture of Kent distributions.

[★] This work was supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1106400.

Our proposed method solves the problem of point registration in a probabilistic fashion by incorporating the orientational information from the pointcloud. The method is mainly motivated by the fact that it can robustify the registration to noise and outliers by incorporating surface normals. In particular, the surface normals are assumed to be drawn from a directional distribution called Kent distribution. Directional statistic is a suitable way to represent pointcloud features like surface normals, since they represent a directional pattern in the pointcloud and also they are invariant with respect to translation. In this approach the estimation of rigid transformation can be decoupled into two steps: first, the rotation matrix between two frames can be estimated using the surface normals, and second, the translation vector can be found using the positional information of the pointcloud. The decoupling of rotation matrix from translation vector is also addressed in Thomas et al. (2019).

Many previous works use the probabilistic approach (i.e. soft-assignment) for the correspondence phase (Rangarajan et al. (1996); Luo and Hancock (2001); Myronenko and Song (2010)). In contrast to hard-assignment or one-to-one correspondence in classical ICP methods, in this method the points are associated to each other in two frames according to a probability distribution. Myronenko and Song (2010) formulated the point registration problem as a maximum likelihood (ML) estimation problem by using Gaussian mixture model (GMM). Evangelidis and Horaud (2018) also use GMM to describe and register multiple pointclouds jointly that are assumed to be drawn from an underlying distribution. There are other variants of GMM method with different noise and outlier conditions (Jian and Vemuri (2011); Horaud et al. (2011)).

Recently, Min et al. (2019) and Billings and Taylor (2015) proposed point registration with Von Mises–Fisher distribution which is also a directional distribution defined on a unit sphere for 3D points. The method is a hybrid method that combines both directional and positional information into a hybrid of Gaussian and Von Mises distributions. Although the method improves the registration with regard to outliers, its isotropic assumption is not realistic in the real world applications. In contrast, Kent distribution accounts for anisotropic (i.e. the ovalness of surface normals on a unit sphere) dataset as seen in Fig. 1. This will further address the problem of outliers in dataset. Based on the proposed statistical framework, the iterations of finding correspondences and updating transformations are now considered as a type of Expectation-Maximization (EM) procedure.

The rest of this paper is organized as follows; first in Section 2 we start by preliminaries and introducing nomenclatures, including the definition of Kent distribution. Then we introduce the mixture model that describes the probabilistic relationship between points in two frames. In Section 3 we detail the steps in the EM algorithm in the proposed method. In Section 4 we present the results from the experiments on ETH dataset, and finally in Section 5 we make the concluding remarks and discuss the future directions.

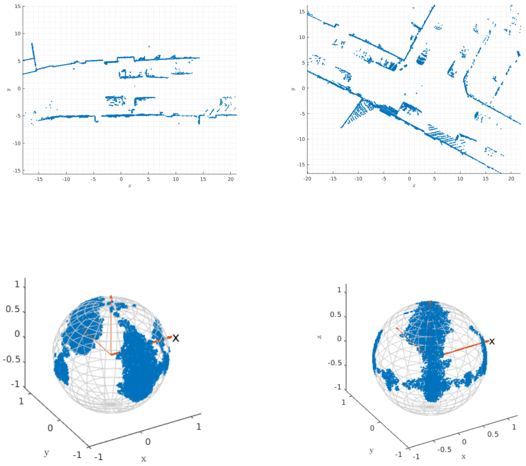

Fig. 2. Example from the KITTI dataset Geiger et al. (2013). Figures show two urban scenes pointcloud data and associated surface normals on a unit sphere. As seen, as the vehicle moves, the surface normals are also moving on the unit sphere which indicates the rotation between consecutive frames.

2. METHODS

2.1 Preliminaries

Throughout the paper we use the following notations for pointclouds:

- M, N– number of points in the *model* and *observed* pointclouds, respectively,
- $\mathbf{Y} = \{\mathbf{y}_i\}_{m=1,\ldots,M}$, $\mathbf{y}_i \in \mathbb{R}^3$– the *model* pointcloud,
- $\mathbf{X} = \{\mathbf{x}_i\}_{n=1,\ldots,N}$, $\mathbf{x}_i \in \mathbb{R}^3$– the *observed* pointcloud,
- $\tilde{\mathbf{Y}} = \{\tilde{\mathbf{y}}_m \in \mathbb{R}^3 : \|\tilde{\mathbf{y}}_i\|_2 = 1\}_{i=1,\ldots,M}$– the surface normals of *model* data,
- $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_n \in \mathbb{R}^3 : \|\tilde{\mathbf{x}}_\mathbf{i}\|_2 = 1\}_{i=1,\ldots,N}$– the surface normals of *observed* data,
- $\boldsymbol{\mu}_y$ the positional mean of the *model* pointcloud,
- $\boldsymbol{\mu}_x$ the positional mean of the *observed* pointcloud,
- $\mathbf{z}_n$, $1 \leq n \leq N$, the hidden random variable in the mixture model.

Kent distribution is the spherical analogous of the bivariate Gaussian distribution which was introduced by Kent (1982). Since the distribution is defined over the set of unit vector on a sphere, it is suitable for representing surface normals of data points in a pointcloud. The distribution can be represented with 5 parameters $(\kappa, \beta, \boldsymbol{\mu}, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)$ as [1] ,

$$\mathrm{FB}_5(\tilde{\mathbf{x}}) = c(\kappa, \beta)^{-1} \exp\left\{\kappa \boldsymbol{\mu}^\top \tilde{\mathbf{x}} + \beta\left[(\boldsymbol{\gamma}_1^\top \tilde{\mathbf{x}})^2 - (\boldsymbol{\gamma}_2^\top \tilde{\mathbf{x}})^2\right]\right\},$$

where $\kappa \geq 0$ is the *concentration*, $\beta \geq 0$ describes *ovalness*, $\boldsymbol{\mu} \in \mathbb{R}^3$ is the *mean direction*, and $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 \in \mathbb{R}^3$ are the *major axis* and *minor axis* of the orientation of the distribution, respectively. Together, $\boldsymbol{\Gamma} = (\boldsymbol{\mu}, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)$ creates an orthogonal matrix representing the total orientation of data on the sphere. $c(\kappa, \beta)$ is the normalization term which can be expressed, in general, with an infinite sequence of the Modified Bessel functions, but it can be

[1] In the original paper the distribution was called Fisher-Bingham with 5 parameters, therefore we use the same notation $\mathrm{FB}_5(.)$ in this paper as well.

simplified under the assumption of $2\beta < \kappa$ and large κ as $c(\kappa, \beta) = \frac{2\pi e^{\kappa}}{(\kappa^2 - 4\beta^2)^{1/2}}$.

2.2 Problem Formulation

The point registration problem can be modeled as a mixture of Kent distributions. The distribution which describes the probability of an observed data point corresponding to a model data point has the form (Billings and Taylor (2015)),

$$\begin{aligned} &\mathrm{FB}_5(\tilde{\mathbf{x}}_n | \mathbf{z}_n = \tilde{\mathbf{y}}_m) = \\ &c(\kappa, \beta)^{-1} \exp\left\{\kappa \tilde{\mathbf{y}}_m^{\top} \tilde{\mathbf{x}}_n + \beta\left[(\boldsymbol{\gamma}_1^{\top} \tilde{\mathbf{x}}_n)^2 - (\boldsymbol{\gamma}_2^{\top} \tilde{\mathbf{x}}_n)^2\right]\right\}, \end{aligned} \tag{1}$$

where $\tilde{\mathbf{y}}_m$ are substituted as the *mean directions*, and consequently the marginal distribution of an observed data point $\tilde{\mathbf{x}}_n$ can be computed as,

$$p(\tilde{\mathbf{x}}_n) = \sum_{m=1}^{M} \pi_m \mathrm{FB}_5(\tilde{\mathbf{x}}_n | \mathbf{z}_n = \tilde{\mathbf{y}}_m) + \pi_0 p_0(\tilde{\mathbf{x}}_n). \tag{2}$$

In general, the membership probabilities are part of parameter estimation in mixture models, but here they are assumed to be constant and identical. We also add a uniform distribution which accounts for the noise/outlier in the data points as $p_0(\tilde{\mathbf{x}}_n) = \frac{1}{N}$. π_0 can be chosen empirically based on the approximate amount of noise/outlier in the dataset. Fig. 2 shows an example of two urban scenes and the estimated surface normals which represents a mixture of Kent distributions. Normals rotate on the sphere in consecutive frames and are invariant to the translation along the trajectory of the vehicle.

By assuming that all surface normals of the observed data points $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N$ are independent, the likelihood function is

$$L(\boldsymbol{\Theta}) = p(\tilde{\mathbf{X}} | \tilde{\mathbf{Y}}; \boldsymbol{\Theta}) = \prod_{n=1}^{N} p(\tilde{\mathbf{x}}_n), \tag{3}$$

and the log-likelihood function can be computed as

$$\begin{aligned} \log L(\boldsymbol{\Theta}) &= \sum_{n=1}^{N} \log p(\tilde{\mathbf{x}}_n) \qquad (4) \\ &= \sum_{n=1}^{N} \log \sum_{m=1}^{M} \left(\pi_m \mathrm{FB}_5(\tilde{\mathbf{x}}_n | \mathbf{z}_n = \tilde{\mathbf{y}}_m) + \pi_0 p_0(\tilde{\mathbf{x}}_n)\right), \end{aligned}$$

where the parameters are $\boldsymbol{\Theta} = (\kappa, \beta, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \mathbf{R})$. The matrix $\mathbf{R}$ rotates the surface normals of the *observed* pointcloud $\tilde{\mathbf{X}}$ to the surface normals of the **model** pointcloud $\tilde{\mathbf{Y}}$, and has been expressed implicitly in (1) as $\tilde{\mathbf{y}}_m = \mathbf{R}\tilde{\mathbf{x}}_n$. Rotating surface normals does not change other parameters in the distribution as stated in the following lemma.

Lemma 1. If $\mathbf{R}$ is an orthogonal matrix, and $\tilde{\mathbf{x}} \sim \mathrm{FB}_5(\kappa, \beta, \boldsymbol{\mu}, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)$, then, $\tilde{\mathbf{y}} \sim \mathrm{FB}_5(\kappa, \beta, \mathbf{R}\boldsymbol{\mu}, \mathbf{R}\boldsymbol{\gamma}_1, \mathbf{R}\boldsymbol{\gamma}_2)$ when $\tilde{\mathbf{y}} = \mathbf{R}\tilde{\mathbf{x}}$.

Proof. From the definition of Kent distribution we know that we always have $\tilde{\mathbf{x}}^* \sim \mathrm{FB}_5(\kappa, \beta, \mathbf{I})$ when $\tilde{\mathbf{x}}^* = \boldsymbol{\Gamma}^{\top}\tilde{\mathbf{x}}$. Then if $\tilde{\mathbf{x}}^* = (\mathbf{R}\boldsymbol{\Gamma})^{\top}\tilde{\mathbf{y}}$, therefore, $\tilde{\mathbf{x}}^* = \boldsymbol{\Gamma}^{\top}\mathbf{R}^{\top}\tilde{\mathbf{y}}$, which means $\tilde{\mathbf{y}} = \mathbf{R}\tilde{\mathbf{x}}$.

This is also intuitive that the orientation parameter does not change the compactness of the Kent distribution, since the rotation matrix is orthogonal, therefore κ and β are preserved. This helps us to estimate the change of the orientation of surface normals without considering any changes in the compactness and ovalness.

Algorithm 1 Main Algorithm

```
Input: observed and model pointclouds: $\mathbf{X}$, $\mathbf{Y}$.
Output: $\mathbf{R}$, $\mathbf{t}$
Surface normals estimation $\tilde{\mathbf{X}}$, $\tilde{\mathbf{Y}}$ according to Section 3.1
Divide $\tilde{\mathbf{Y}}$, $\tilde{\mathbf{X}}$ to $K$ groups according to Section 3.2
Update $\boldsymbol{\Theta}_k$ in parallel for each group $k = 1, \ldots, K$ according to Algorithm 2.
Average $\mathbf{R}_k$ according to equation (14).
Update $\mathbf{t}$ according to equation (15).
Return $\mathbf{R}$, $\mathbf{t}$.
```

Log-likelihood function in (4) assumes a prior knowledge of point correspondence (i.e. the correspondence between points in two pointclouds). However, in practice we don't know those correspondences and we refer to the observed data as *incomplete.* Since the correspondence between surface normals in *observed* and *model* data is missing, the hidden variable $\mathbf{z}_n$ is introduced in EM algorithm to represent their probabilistic assignment. We describe an EM-like approach in the next section in order to iteratively solve for *incomplete-data* version of the maximum likelihood (ML) problem.

3. POINT REGISTRATION ALGORITHM

In this section we describe the proposed approach for 3D point registration using Kent distribution. The method relies on decoupling of rotation from translation by estimating rotation based on surface normals and translation from the positional data points. Algorithms 1 and 2 outline the steps of the method.

3.1 Surface Normals Estimation

Several methods exist to estimate the surface normals associated with each point in the pointcloud (Klasing et al. (2009)). In this paper we compute the normals using principal component analysis (PCA) method. To improve the performance of the proposed method, we pre-process the surface normals with: a) orientation consistency and b) outliers removal. The orientations of surface normals computed by PCA are ambiguous (i.e. it is impossible to solve for the sign of normal vectors). To ensure we have consistent surface normals in the algorithm, we use a defined viewpoint $\mathbf{v}_p$ to make their orientation consistent (Rusu (2013)). All normals $\mathbf{n}_i$ of points $\mathbf{x}_i$ with the same directions should satisfy $\mathbf{n}_i \cdot (\mathbf{v}_p - \mathbf{x}_i) > 0$. We also use a modified version of outlier removal method for the surface normals. We utilize *cosine similarity* in order to remove normals that are far away from their *mean direction.* The method is detailed in Zhang (1994) and Rusu et al. (2008).

3.2 Clustering with Spherical k-means

In the most of urban scene pointclouds, the majority of surface normals are estimated based on prominent surfaces such as buildings which generate the pattern of surface normals concentrated in specific areas on the sphere (for

Algorithm 2 EM-like Algorithm

```
Initialization: $\kappa, \beta, \gamma_1, \gamma_2, \mathbf{R}$
while not converged do
    E-step: Compute posterior probabilities $\tau_{mn}$ according to equation (6)
    M-step:
    Update $\mathbf{R}$ according to equation (12)
    Compute parameters using moment estimates according to Section 3.5.2
end while
Return $\kappa, \beta, \gamma_1, \gamma_2, \mathbf{R}$
```

example see Fig. 2). Therefore we use spherical k-means approach (Hornik et al. (2012)) to cluster surface normals and use those normals from the same cluster in both observed and model data points. This helps to run the algorithm in parallel for the smaller size of data points in each cluster and finally take an average to find the overall rotation matrix in each step. It is worth noting that the spherical clustering is closely related to ML estimate of Von-Mises distribution which is the same as Kent distribution when $\beta = 0$ (Banerjee et al. (2005)). Fig. 3 depicts steps in order to prepare surface normals for the EM algorithm which will be explained in the following subsection.

3.3 EM-like Algorithm

A variant of EM approach is adopted here to estimate the parameter $\mathbf{\Theta}$. As stated in Section 2, since the point correspondence is not available (i.e. the observed data is viewed as being *incomplete*), log-likelihood function in (4) cannot be optimized directly. The *complete-data log-likelihood* function is given by

$$\log L_c(\mathbf{\Theta}) = \sum_{n=1}^{N}\sum_{m=1}^{M} P(\mathbf{z}_n|\tilde{\mathbf{x}}_n)\Big(\log \pi_0 + \log p_0(\tilde{\mathbf{x}}_n) + \log \pi_m + \log \mathrm{FB}_5(\tilde{\mathbf{x}}_n|\mathbf{z}_n = \tilde{\mathbf{y}}_m)\Big), \qquad (5)$$

where $P(\mathbf{z}_n|\tilde{\mathbf{x}}_n)$ is the posterior correspondence probability that links the observed data to the realization of hidden variables. The parameters will be found by minimizing the negative of the conditional expectation of $\log L_c(\mathbf{\Theta})$ iteratively in the following two steps.

3.4 E-step

The hidden correspondence-label $\mathbf{z}_n$ will be handled by the *E-step*, which is computing the posterior probability distribution τ_{mn}. This posterior distribution plays the role of *soft-assignment* between the model normals $\tilde{\mathbf{y}}_m$ and the observed normals $\tilde{\mathbf{x}}_n$, and can be computed as

$$\tau_{mn} = P(\mathbf{z}_n = \tilde{\mathbf{y}}_m|\tilde{\mathbf{x}}_n) = \frac{\pi_m \mathrm{FB}_5(\tilde{\mathbf{x}}_n|\mathbf{z}_n = \tilde{\mathbf{y}}_m)}{p(\tilde{\mathbf{x}}_n)}, \qquad (6)$$

for $m = 1, \dots, M$, $n = 1, \dots, N$, and where the denominator is defined in (2). The posterior probability of assigning observed normal to an outlier is $\tau_{0n} = 1 - \sum_{m=1}^{M} \tau_{mn}$. By replacing τ_{mn} in (5) we have the function that needs to be maximized in the *M-step*,

$$Q(\mathbf{\Theta}) = \sum_{n=1}^{N}\sum_{m=1}^{M} \tau_{mn}\big(\log \pi_0 + \log p_0(\tilde{\mathbf{x}}_n) + \log \pi_m + \log \mathrm{FB}_5(\tilde{\mathbf{x}}_n|\mathbf{z}_n = \tilde{\mathbf{y}}_m)\big). \qquad (7)$$

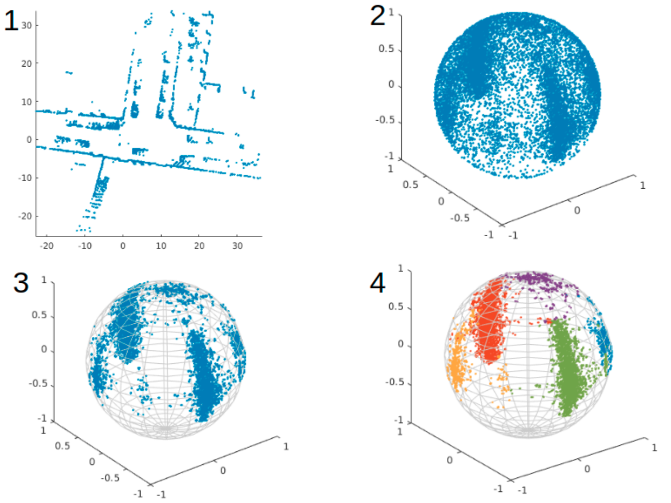

Fig. 3. The process of preparing surface normals from pointcloud data. 1) pointcloud 2) estimating surface normals 3) outlier removal 4) *spherical* clustering.

3.5 M-step

This step requires to find the parameter $\mathbf{\Theta}$ of the distribution by maximizing the $Q(\mathbf{\Theta})$ function

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} \ Q(\mathbf{\Theta}). \qquad (8)$$

Although the optimization problem in (8) has closed form solution for some types of probability distributions, unfortunately due to the orthogonality constraints on Kent distribution parameters and rotation matrix, this constrained nonlinear problem is hard to solve. Therefore we solve the problem in two sub-steps: finding rotation matrix and then updating Kent distribution parameters.

Rotation Matrix Update: Substituting in (7) we have

$$Q(\mathbf{\Theta}) = \sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\Big(\log \pi_0 + \log p_0(\tilde{\mathbf{x}}_n) + \log \pi_m + \log c(\kappa, \beta)^{-1} + \kappa \tilde{\mathbf{y}}_m^\top \mathbf{R}\tilde{\mathbf{x}}_n + \beta\big[(\gamma_1^\top \mathbf{R}\tilde{\mathbf{x}}_n)^2 - (\gamma_2^\top \mathbf{R}\tilde{\mathbf{x}}_n)^2\big]\Big). \qquad (9)$$

The optimal rotation matrix can be found by maximizing Q-function with respect to the rotation matrix which belongs to the set of *special orthogonal* matrices $SO(3)$ defined as

$$SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} | \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det \mathbf{R} = 1\}. \qquad (10)$$

After removing constant terms and also minimizing negative of Q-function, the constraint optimization problem can be written as,

$$\mathbf{R}^* = \underset{\mathbf{R}\in SO(3)}{\arg\min} \; -Q(\mathbf{\Theta}) = \underset{\mathbf{R}\in SO(3)}{\arg\min}$$
$$\sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\big(-\kappa\tilde{\mathbf{y}}_m^\top \mathbf{R}\tilde{\mathbf{x}}_n + \beta\big[(\gamma_2^\top \mathbf{R}\tilde{\mathbf{x}}_n)^2 - (\gamma_1^\top \mathbf{R}\tilde{\mathbf{x}}_n)^2\big]\big)$$
$$= \underset{\mathbf{R}\in SO(3)}{\arg\min} -\kappa \sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\tilde{\mathbf{y}}_m^\top \mathbf{R}\tilde{\mathbf{x}}_n$$
$$+ \beta \sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}(\tilde{\mathbf{x}}_n^\top \mathbf{R}^\top(\gamma_2\gamma_2^\top - \gamma_1\gamma_1^\top)\mathbf{R}\tilde{\mathbf{x}}_n). \qquad (11)$$

Using matrix trace and its *cyclic property*, the first term above can be rewritten as

$$\sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\tilde{\mathbf{y}}_m^\top \mathbf{R}\tilde{\mathbf{x}}_n = \sum_{m=1}^{M}\sum_{n=1}^{N} \mathrm{trace}(\tau_{mn}\tilde{\mathbf{y}}_m^\top \mathbf{R}\tilde{\mathbf{x}}_n) =$$
$$\sum_{m=1}^{M}\sum_{n=1}^{N} \mathrm{trace}(\mathbf{R}\tau_{mn}\tilde{\mathbf{x}}_n\tilde{\mathbf{y}}_m^\top) = \mathrm{trace}(\mathbf{R}\sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\tilde{\mathbf{x}}_n\tilde{\mathbf{y}}_m^\top).$$

The second term in (11) also can be simplified the same way,

$$\sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\tilde{\mathbf{x}}_n^\top \mathbf{R}^\top(\gamma_2\gamma_2^\top - \gamma_1\gamma_2^\top)\mathbf{R}\tilde{\mathbf{x}}_n =$$
$$\sum_{m=1}^{M}\sum_{n=1}^{N} \mathrm{trace}\big(\tau_{mn}\tilde{\mathbf{x}}_n^\top \mathbf{R}^\top(\gamma_2\gamma_2^\top - \gamma_1\gamma_1^\top)\mathbf{R}\tilde{\mathbf{x}}_n\big) =$$
$$\sum_{m=1}^{M}\sum_{n=1}^{N} \mathrm{trace}\big(\mathbf{R}^\top(\gamma_2\gamma_2^\top - \gamma_1\gamma_1^\top)\mathbf{R}\tau_{mn}\tilde{\mathbf{x}}_n\tilde{\mathbf{x}}_n^\top\big) =$$
$$\mathrm{trace}\Bigg(\mathbf{R}^\top(\gamma_2\gamma_2^\top - \gamma_1\gamma_1^\top)\mathbf{R}\sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\tilde{\mathbf{x}}_n\tilde{\mathbf{x}}_n^\top\Bigg).$$

By substituting the following matrices

$$\mathbf{A} = \gamma_2\gamma_2^\top - \gamma_1\gamma_1^\top$$
$$\mathbf{B} = \sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\tilde{\mathbf{x}}_n\tilde{\mathbf{x}}_n^\top$$
$$\mathbf{C} = \sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}\tilde{\mathbf{x}}_n\tilde{\mathbf{y}}_m^\top,$$

and rearranging the terms, we can rewrite (11) in a compact matrix form as

$$\mathbf{R}^* = \underset{\mathbf{R}\in SO(3)}{\arg\min} \; \beta\mathrm{trace}(\mathbf{R}^\top\mathbf{A}\mathbf{R}\mathbf{B}) - \kappa\mathrm{trace}(\mathbf{R}\mathbf{C}). \qquad (12)$$

Since the constraint in (10) is the set of *special orthogonal group* or the set of rotation matrices which are smooth, it is convenient to use Riemannian manifold optimization. We use the manifold optimization solver from the package in Boumal et al. (2014).

Distribution Parameters Update: After updating the rotation matrix, we need to update distribution parameters as well. Recalling equation (9), we just retain terms that depend on the parameters and also substitute the optimal rotation matrix from previous step,

$$(\kappa^*, \beta^*, \gamma_1^*, \gamma_2^*) = \underset{\gamma_1^\top\cdot\gamma_2=0}{\arg\min} \sum_{m,n=1}^{M,N} \tau_{mn}\Bigg(-\log c(\kappa,\beta)^{-1}$$
$$-\kappa\tilde{\mathbf{y}}_m^\top \mathbf{R}^*\tilde{\mathbf{x}}_n + \beta\big[(\gamma_2^\top \mathbf{R}^*\tilde{\mathbf{x}}_n)^2 - (\gamma_1^\top \mathbf{R}^*\tilde{\mathbf{x}}_n)^2\big]\Bigg). \qquad (13)$$

There are some attempts to solve variant of this problem (in the context of mixture of Kent distributions) iteratively using BFGS quasi-Newton method with reparametrization of variables (Billings and Taylor (2015)) or BSLM approach (Nguyen (2017)), but there is no guarantee to find a solution considering the nonconvex nature of the problem. A more convenient method to solve for parameters, as described in Peel et al. (2001), is the *moment estimation* method. The method should be adapted for the mixture of Kent distribution, as it is only described before for the unimodal Kent distributions. Here we compute the *weighted sample moments* as follows,

$$\bar{\tilde{\mathbf{x}}} = \frac{\sum_{n=1}^{N} \mathcal{T}_n(\mathbf{R}^*\tilde{\mathbf{x}}_n)}{N_p}$$
$$\mathbf{S} = \frac{\sum_{n=1}^{N} \mathcal{T}_n(\mathbf{R}^*\tilde{\mathbf{x}}_n)(\mathbf{R}^*\tilde{\mathbf{x}}_n)^\top}{N_p}$$

where $\mathcal{T}_n = \sum_{m=1}^{M} \tau_{mn}$ and $N_p = \sum_{m=1}^{M}\sum_{n=1}^{N} \tau_{mn}$. Using the above *weighted moments*, the rest of the method has been explained in Kent (1982), and we exclude the details here for brevity.

Using the moment estimate instead of ML estimate makes the EM algorithm to lose the theoretical convergence guarantees, however as stated in Kent (1982), the moment estimation is very close to ML estimate in case of small eccentricity $\frac{2\beta}{\kappa}$ or large κ, which is a usual case in practice.

Rotation averaging and Translation: Finally we need to compute average of the rotations from different clusters and estimate the translation. Rotation averaging in Euclidean sense can be computed as,

$$\mathbf{R} = \overline{\mathbf{R}}\mathbf{U}\mathrm{diag}(\frac{1}{\sqrt{\Lambda_1}}, \frac{1}{\sqrt{\Lambda_2}}, \frac{s}{\sqrt{\Lambda_3}})\mathbf{U}^\top \qquad (14)$$

where $\overline{\mathbf{R}} = \frac{\sum_i^N \mathbf{R}_i^*}{N}$, $N^2\mathbf{U}^\top\mathbf{D}\mathbf{U} = N^2\overline{\mathbf{R}}^\top\overline{\mathbf{R}}$, and $\mathbf{D} = \mathrm{diag}(\Lambda_1, \Lambda_2, \Lambda_3)$. Also $s = 1$ if $\det(\overline{\mathbf{R}})$ is positive and $s = -1$ otherwise (Moakher (2002)). $\mathbf{R}_i^*$ are the optimal rotation matrices found by the algorithm for each cluster in the dataset. The translation vector also can easily be computed based on the positional mean of rotated *observed* data and *model* data,

$$\mathbf{t}^* = \boldsymbol{\mu}_y - \mathbf{R}^*\boldsymbol{\mu}_x. \qquad (15)$$

This is valid, since it is the optimal translation in the sense of Euclidean norm which has been used in ICP-based methods as well.

4. EXPERIMENTAL RESULTS

In this section the proposed method is tested and verified via 3D pointcloud registration on real indoor scanner pointclouds (Pomerleau et al. (2012)). We pre-processed the pointcloud frames before applying the algorithm. In the first stage, frames were down sampled to 10% of the original size (about $10,000$ points) which was empirically chosen. In addition, all the processes on the surface normals described in Section 3.1 was performed. The proposed method was implemented in MATLAB. The algorithm was initialized with identity rotation matrix and zero translation vector. The distribution parameters were initialized in E-step based on the current frame. We empirically found best results with 15 neighborhood points for surface normals estimation. We adopt the error metric for comparison

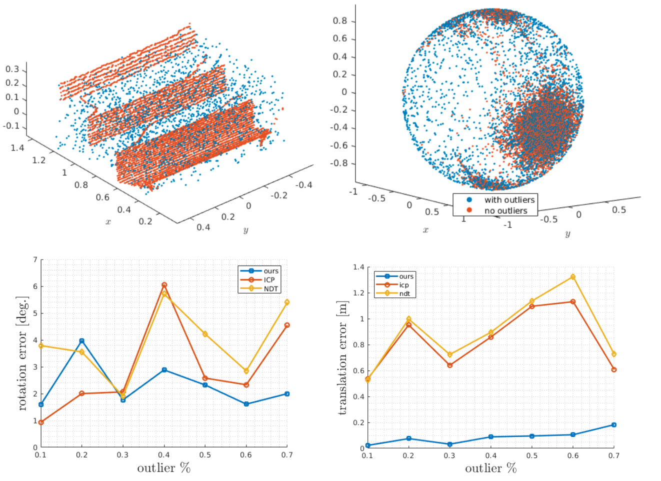

Fig. 4. Top: Example of staircase pointcloud with 20% outliers (blue points) and its associated surface normals. Bottom: Comparison of rotation and translation errors.

of estimated and true rotation matrices from Wang et al. (2018),

$$e_{\widehat{\mathbf{R}}} = \cos^{-1}\left(\frac{\text{trace}(\mathbf{R}^\top\widehat{\mathbf{R}}) - 1}{2}\right)$$

where $\widehat{\mathbf{R}}$ and $\mathbf{R}$ are estimated and true rotation matrix, respectively. And for the translation error we have $l2$-norm of the difference of estimated and true translation $e_{\widehat{\mathbf{t}}} = ||\widehat{\mathbf{t}} - \mathbf{t}||_2$.

4.1 Outlier Robustness

We performed a set of point registration experiments comparing the robustness of the proposed method with ICP and NDT methods. Fig. 4 shows an example of frame with outlier points in blue. It can be seen that how the positional outliers impact and spread their associated surface normals on the unit sphere. The bottom plots illustrate the average of rotation and translation errors with different level of outliers injected into the pointclouds over $N_{trials} = 100$ trials. As seen, ICP and NDT methods have inconsistent and higher error with regard to outliers which makes them unreliable in practice. In contrast, the proposed method has relatively lower errors and consistent with the level of outliers.

4.2 Performance Evaluation

The results from comparison of different point registration methods are tabulated in table 1. The results show the average rotation and translation error on the sequence of same frames from the dataset. For NDT (Stoyanov et al. (2012)), the grid size is set as $2.0, 4.0, 2.0, 1.0$ meters with aligns the pointclouds in a finer-coarser-to-finer manner, and for CICP (Du et al. (2018)) the scale factor is $\sigma = 1.0$ meter with a decay rate of 0.96. Fig. 5 represents an example of the point matching of two consecutive frames. The green is the pointcloud that is transformed back into the model pointcloud based on the estimated rigid transformation found by the proposed method.

Table 1. Comparison of PR Methods

Trials	ICP	ICPp2pl	CICP	GICP	NDT	Ours
Avg. $e_{\widehat{\mathbf{R}}}$	1.27	0.57	1.162	0.23	0.52	0.66
Avg. $e_{\widehat{\mathbf{t}}}$	0.403	0.164	0.260	0.025	0.070	0.018

As seen in the table 1, the proposed algorithm outperforms other methods with regard to translation error and generates satisfying rotation errors compared with the state-of-the-art NDT and GICP. It is reasonable to infer that the more information involved, the more accurate of the point registration algorithm. For instance, the ICP and CICP based on point-to-point distance produce the worse matching results. Likewise, GICP and NDT utlize more geometrical information like curvatures and more abstracted covariance matrices, which guarantees the better results.

The proposed feature-based point registration method has two advantages over other methods: firstly only the directional information is utilized for rotation estimation and secondly the computationally expensive correspondence phase in ICP- or NDT-based methods is avoided, but the registration results are still competitive even with NDT or GICP. On the other hand, since the indoor pointcloud is free of most of missing data comparing with outdoor scenes, we expect our method would perform better in the presence of outliers, occluded, and missing data as presented in the previous experiment case.

Finally as stated in Section 3, although using *moment estimate* method makes the EM algorithm without convergence guarantee, since the algorithm utilizes the highly concentrated surface normals on the unit sphere (i.e. large κ) the moment estimate is closely related to ML estimates. In fact, the empirical results from our experiment show that (7) the Q-function is not decreasing after each iteration of the algorithm.

5. CONCLUSION

In this paper, our proposed feature-based method for 3D point registration was presented and evaluated with examples of indoor pointclouds. The algorithm incorporates surface normals in the scene as directional information to be used within our cohesive and probabilistic framework which improves the registration accuracy comparing to ICP-based methods. The method utilizes the fact that the translation-invariant property of surface normals decouples the estimation of rotation from translation. Additionally, the proposed method provides a robust mechanism that rejects outliers in the correspondence phase in a probabilistic fashion. The computation time comparison is not analyzed in this paper and one possible future direction can be reducing the computation time of the proposed algorithm.

REFERENCES

Banerjee, A., Dhillon, I.S., Ghosh, J., and Sra, S. (2005). Clustering on the unit hypersphere using Von Mises-Fisher distributions. *J. Mach. Learn. Res.*, 6, 1345–1382.

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

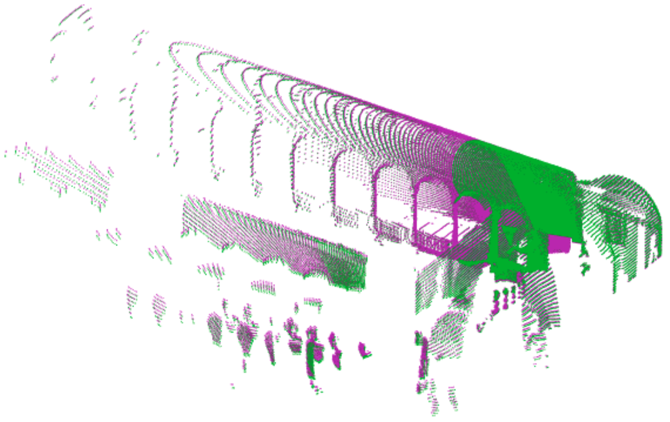

Fig. 5. An example of matching quality (i.e. frames alignment) in two consecutive frames from indoor scanning data set Pomerleau et al. (2012) using the proposed method. Purple: the model pointcloud. Green: the transformed pointcloud.

Billings, S. and Taylor, R. (2015). Generalized iterative most likely oriented-point (G-IMLOP) registration. *International Journal of Computer Assisted Radiology and Surgery*, 10(8), 1213–1226.

Boumal, N., Mishra, B., Absil, P.A., and Sepulchre, R. (2014). Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15, 1455–1459.

Du, S., Xu, G., Zhang, S., Zhang, X., Gao, Y., and Chen, B. (2018). Robust rigid registration algorithm based on pointwise correspondence and correntropy. *Pattern Recognition Letters.*

Evangelidis, G.D. and Horaud, R. (2018). Joint alignment of multiple point sets with batch and incremental expectation-maximization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6), 1397–1410.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR).*

Horaud, R., Forbes, F., Yguel, M., Dewaele, G., and Zhang, J. (2011). Rigid and articulated point registration with expectation conditional maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3), 587–602.

Hornik, K., Feinerer, I., Kober, M., and Buchta, C. (2012). Spherical k-means clustering. *Journal of Statistical Software*, 50, 1–22.

Jian, B. and Vemuri, B. (2011). Robust point set registration using Gaussian mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33, 1633 – 1645.

Kent, J.T. (1982). The Fisher-Bingham distribution on the sphere. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(1), 71–80.

Klasing, K., Althoff, D., Wollherr, D., and Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. *Proceedings - IEEE International Conference on Robotics and Automation*, 3206–3211.

Low, K.L. (2004). Linear least-squares optimization for point-to-plane ICP surface registration.

Luo, B. and Hancock, E.R. (2001). Structural graph matching using the EM algorithm and singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10), 1120–1136.

Min, Z., Wang, J., and Meng, M.Q. (2019). Robust generalized point cloud registration with orientational data based on expectation maximization. *IEEE Transactions on Automation Science and Engineering*, 1–15.

Moakher, M. (2002). Means and averaging in the group of rotations. *SIAM J. Matrix Anal. Appl.*, 24(1), 1–16.

Myronenko, A. and Song, X. (2010). Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32, 2262–75.

Nguyen, H. (2017). A novel algorithm for clustering of data on the unit sphere via mixture models.

Peel, D., Whiten, W.J., and McLachlan, G.J. (2001). Fitting mixtures of Kent distributions to aid in joint set identification. *Journal of the American Statistical Association*, 96(453), 56–63.

Pomerleau, F., Liu, M., Colas, F., and Siegwart, R. (2012). Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14), 1705–1711.

Rangarajan, A., Mjolsness, E., Pappu, S., Davachi, L., Goldman-Rakic, P.S., and Duncan, J.S. (1996). *A robust point matching algorithm for autoradiograph alignment*, 277–286. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rusu, R.B. (2013). *Semantic 3D Object Maps for Everyday Robot Manipulation.* Springer Publishing Company, Incorporated.

Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., and Beetz, M. (2008). Towards 3D point cloud based object maps for household environments. *Robot. Auton. Syst.*, 56(11), 927–941.

Segal, A., Hähnel, D., and Thrun, S. (2009). Generalized-ICP. *Proc. of Robotics: Science and Systems.*

Serafin, J. and Grisetti, G. (2015). NICP: Dense normal based point cloud registration. *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 742–749.

Stoyanov, T., Magnusson, M., and Lilienthal, A.J. (2012). Point set registration through minimization of the l2 distance between 3d-ndt models. In *2012 IEEE International Conference on Robotics and Automation*, 5196–5201.

Thomas, Q.M., Wasenmüller, O., and Stricker, D. (2019). DeLiO: Decoupled LiDAR odometry. *CoRR*, abs/1904.12667.

Wang, D., Xue, J., Tao, Z., Zhong, Y., Cui, D., Du, S., and Zheng, N. (2018). Accurate mix-norm-based scan matching. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1665–1671. IEEE.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13, 119–152.