Constrained Gaussian Mixture Models Based Scan Matching Method

Jiaheng Zhao[1,2], Shoudong Huang[1], Liang Zhao[1]
[1]Centre for Autonomous Systems, University of Technology Sydney, Australia
jiaheng.zhao@student.uts.edu.au, {shoudong.huang;liang.zhao}@uts.edu.au
[2]Beijing Institute of Technology, China

Abstract

This paper presents a Gaussian mixture model (GMM) based robust scan matching method which implements GMM to represent 2D scan points and improves the accuracy of scan matching. The proposed method transfers each new scan to GMM first, exploiting the covariance of every GMM component to represent scan points. Compared with the conventional GMM based method of scan matching, our technique implements GMM similarity comparison to evaluate the overlaps between scans. In order to get rid of the poor convergence due to the inaccurate initial value given to the iteration process, we proposed a geometry-constraint-based GMM similarity calculation method, which is one contribution of this paper. Another contribution is we propose a dynamic scale factor making the cost function more adapted to different initial value. Experiments on simulated data are employed and the results indicate that our method is able to enlarge the valid range of initial value and accumulate small errors after sequential matchings.

1 Introduction

Scan matching is an important problem in the area of mobile robot localization and mapping. The robot's poses can be estimated by means of matching consective scans collected from laser sensors integrated in the robot system. Two classes of methods are adopted in recent years research, namely, Iterated Closest Point (ICP) [Besl and McKay, 1992] based method and GMM based method [Jian and Vemuri, 2011]. The latter one can be extended to a famous special case named Normal Distribution Transform (NDT) [Stoyanov *et al.*, 2012].

ICP is a method concentrates on point-to-point registration between scans pairwise (ICP has been extended to optimize point-to-line and point-to-plane registration). The quality of the ICP results highly depends on the initial value. In other words, if the initial value is not near the ground truth, the result of ICP leads to be unauthentic. On the other hand, ICP algorithm is time comsuming when the number of points increases and thus costs too much resources. GMM based method (or NDT based method) is another way to estimate the relative transformation between two scans. NDT methods can be characterized into two classes, one is so-called point-to-distribution method (P2D) as used in [Biber and Wolfgang, 2003; Takeuchi and Tsubouchi, 2006; Magnusson, 2009]. Generally, P2D method registers newly observed points to a reference NDT or NDTs. The other one is usually called distribution-to-distribution (D2D) method according to [Stoyanov *et al.*, 2012], which is, by definition, a method registering the new NDT to the reference NDT.

Both methods try to calculate a metric distance between points and distribution or distribution and distribution. In order to acquire a better performance, several metric distances are introduced to measure the similarity of GMMs. The well-know Kullback-Leibler (KL) divergence is an important metric distance which can be used to maximize the maximum likelihood [Myronenko and Song, 2010], and L_2 distance is imported as an Euclidean distance measurement [Jian and Vemuri, 2011], both of which will be discussed in Section 2. Also [Li *et al.*, 2018] implemented signature quadratic form distance as the metric distance. However, calculating similarity is still restricted to an initial value relatively close to the true value, and most common situation is, unfortunately, it is easy to converge to a local minima or even diverge. [Pu *et al.*, 2018] proposed a dynamic uncertainty-based Gaussian mixtrue alignment method, they exploit Expectation-Maximization (EM) algorithm to estimate transformation from a Mahalanobis distance-like metric.

The performance of GMM based scan matching depends not only on the chosen of metric distance, but also on the adjustment of covariance, cell size and number of normal distributions. [Hong and Lee, 2017] proposed a probabilistic normal distributions transform method.

They considered sensor's uncertainty and joined it with GMMs' covariance, so as to make the algorithm easily converged if the grid size is small. [Das and Waslander, 2014] proposed a segmented region growing NDT method which can improve convergence by means of data segmentation and cluster. They first separated and neglected ground points, then fit and merge the points cloud following rules of distance limitation.

In this paper, we are focusing on GMM based algorithm. We proposed a geometry-constraint-based GMM similarity calculation method, and we also expanded the model, introducing a dynamic scale factor to make the cost function more adaptable to different initial values. The idea of employing geometry constraints is inspired by [Kunjin *et al.*, 2016]. The distinction is that we adopted geometry constraints to provide a criteria of corresponding distributions. Then a scale factor is adjusted during every iteration to provide a more general description of cost function. Section 2 introduces background of GMM based scan matching which is related to our work. Section 3 presents our method and contributions. Experimental results are presented in Section 4. Finally, Section 5 summarizes the phenomenon and conclusion demonstrated in Section 4 and gives a future directions for this work.

2 GMM Based Registration

GMM based scan registration first replace discrete points by a different representation, namely, GMMs. When a new scan is obtained by a lidar, denoted as Z_k, the data is firstly fitted into GMM G_{Z_k}. By doing this, a large number of discrete data points can be represented by a limited number of Gaussian distributions. As shown in Figure 1, the data collected from each scan is fitted to form Gaussian distributions before continuing scan matching. Then scan matching problem is related to GMM's similarity calculation. The GMM based representation is able to represent the shape of different environment (2D or 3D) with an advantage of decreasing complexity. In this paper, we mostly focus on 2D scan matching.

A general GMM is defined as:

$$G\left(x\right)=\sum_{k=1}^{n}\pi_{d_k}\mathcal{N}\left(x\mid\mu_{\mathbf{k}},\Sigma_{\mathrm{k}}\right) \tag{1}$$

where

$$\mathcal{N}\left(x\mid\mu_{\mathbf{k}},\Sigma_{\mathrm{k}}\right)=\frac{\exp\left(-\frac{1}{2}\left(x-\mu_k\right)^T\Sigma_k^{-1}\left(x-\mu_k\right)\right)}{\sqrt{\left(2\pi\right)^d\left|\Sigma_k\right|}} \tag{2}$$

where π_{d_k} is the weight parameters, d is the dimension of estimated variables.

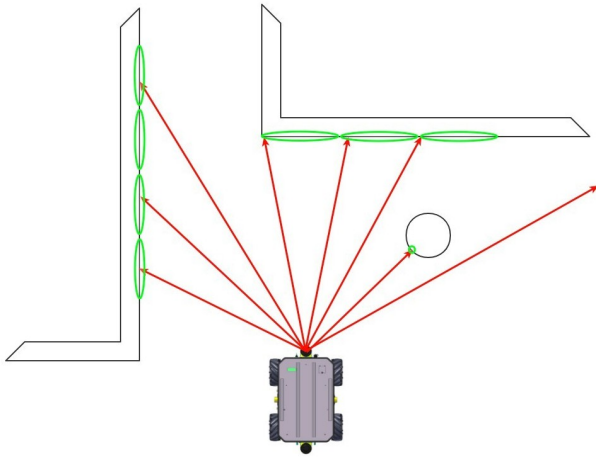

Figure 1: GMM based scan registration in 2-D. Data is fitted into Gaussian distributions as a GMM.

If no prior information is given, we can fit consecutive scans data into GMMs respectively by expectation-maximization method and then calculate the similarity of pairwise GMMs. A special case is Three-Dimensional Normal Distributions Transform (3D-NDT) [Stoyanov *et al.*, 2012]. At this stage, two general directions are adopted for the evaluation, namely, P2D and D2D. P2D is aimed to maximize the likelihood of point set P in distribution G, while D2D cares more on minimizing distance metrics between two distributions. Both algorithms can be solved by general non-linear optimization approaches such as Newton method and Levenberg-Marquardt method. The cost function of P2D is reviewed as follows:

$$f_{P2D}\left(P_{new},G_{ref},\tilde{\Theta}_k\right)=-d_1\sum_{i=1}^{\hat{n}}\exp\left(-\frac{d_2}{2}\left(T\left(P_i,\tilde{\Theta}_k\right)-\mu_G\right)^T\Sigma_G^{-1}\left(T\left(P_i,\tilde{\Theta}_k\right)-\mu_G\right)\right) \tag{3}$$

where $T\left(P_i,\tilde{\Theta}_k\right)$ represents transformation relationship from point set P_{new} to GMM G_{ref} abiding by

$$T\left(P_i,\tilde{\Theta}_k\right)=RP_i+t \tag{4}$$

P_{new} is a point set of $\{P_i\}$ with $i=1,2,3,\cdots,\hat{n}$. G_{ref} is the reference GMM, $\tilde{\Theta}_k$ is the kth estimated R and t during the k^{th} iteration. d_1 and d_2 are positive regularizing factors, μ_G and Σ_G are the mean and covariance of the reference distribution.

For D2D method, several methods have been adopted to calculate the distance of distributions thus comparing the similarity. One is KL divergence. As [Kunjin *et al.*, 2016] shows, the similarity of two Gaussians based on

KL divergence is calculated by

$$\begin{aligned} & S\left(N\left(\bar{z}_{new}, \bar{\Sigma}_{new}\right), N\left(\bar{z}_{ref}, \bar{\Sigma}_{ref}\right)\right) \\ & = -D_{KL}\left(N\left(\bar{z}_{new}, \bar{\Sigma}_{new}\right) \mid\mid N\left(\bar{z}_{ref}, \bar{\Sigma}_{ref}\right)\right) \\ & = -\frac{1}{2}\left\{\mathrm{tr}\left(\left(\bar{\Sigma}_{ref}\right)^{-1}\bar{\Sigma}_{new}\right)\right. \\ & + \left(\bar{z}_{ref}-\bar{z}_{new}\right)^{T}\left(\bar{\Sigma}_{ref}\right)^{-1}\left(\bar{z}_{ref}-\bar{z}_{new}\right) \\ & \left. -\ln\frac{\det\left(\bar{\Sigma}_{new}\right)}{\det\left(\bar{\Sigma}_{ref}\right)}-\lambda\right\} \end{aligned} \tag{5}$$

where λ is the dimension of the distributions, $N\left(\bar{z}_{new}, \bar{\Sigma}_{new}\right)$ and $N\left(\bar{z}_{ref}, \bar{\Sigma}_{ref}\right)$ are the scan NDs and reference NDs, respectively. The aim is to maximize the similarity value.

However, KL divergence is not symmetric, that is $D_{KL}\left[P\left(X\right) \mid\mid Q\left(X\right)\right] \neq D_{KL}\left[Q\left(X\right) \mid\mid P\left(X\right)\right]$, which means the performance is limited. [Jian and Vemuri, 2011] proposed a L_2 distance based algorithm and [Stoyanov *et al.*, 2012] expanded analyses on it. Assuming two GMMs are defined by $G_1 = \sum_i^m w_i N\left(\mu_i, \Sigma_i\right)$ and $G_2 = \sum_j^n w_j N\left(\mu_j, \Sigma_j\right)$.

According to [Jian and Vemuri, 2011] and [Stoyanov *et al.*, 2012], the L_2 distance between two GMMs is defined as:

$$\begin{aligned} D_{L_2}\left(G_1, G_2, p\right) = \int & \left(pdf\left(x \mid G_2\right) - \right. \\ & \left. pdf\left(x \mid T\otimes\left(G_1, p\right)\right)\right)^2 dx \end{aligned} \tag{6}$$

where p is the transformation parameter to be estimated and $T\otimes\left(G_1, p\right)$ denotes the transformation correlation of means and covariances from G_1 to G_2:

$$\begin{aligned} T\otimes\left(\mu_i, p\right) &= R^T\mu_i + t \\ T\otimes\left(\Sigma_i, p\right) &= R^T\Sigma_i R \end{aligned} \tag{7}$$

After expansion,

$$\begin{aligned} D_{L_2}\left(G_1, G_2, p\right) = & \int pdf\left(x \mid G_2\right)^2 dx \\ & + \int pdf\left(x \mid T\otimes\left(G_1, p\right)\right)^2 dx \\ & - 2\int pdf\left(x \mid G_1\right) pdf\left(x \mid T\otimes\left(G_2, p\right)\right) dx \end{aligned} \tag{8}$$

Noting that a GMM can be approximated as another GMM after any arbitrary transformation, then we can simplify equation (8) by neglecting the first two components. And according to the following identity [Jian and Vemuri, 2011]:

$$\begin{aligned} & \int \mathcal{N}\left(x \mid \mu_i, \Sigma_i\right)\mathcal{N}\left(x \mid \mu_j, \Sigma_j\right) dx \\ & = \mathcal{N}\left(0 \mid \mu_i - \mu_j, \Sigma_i + \Sigma_j\right) \end{aligned} \tag{9}$$

thus the distance function is derived as:

$$D_{L_2} \sim \sum_{i=1}^{m}\sum_{j=1}^{n}\mathcal{N}\left(0 \mid T\left(\mu_i, p\right) - \mu_j, T\otimes\left(\Sigma_i, p\right) + \Sigma_j\right) \tag{10}$$

where p is the transformation vector and is parameterized by a vector $p = \left(r_x, r_y, r_z, t_x, t_y, t_z\right)$. The cost function of D2D is derived as:

$$f_{D2D}\left(p\right) = \sum_{i=1}^{m}\sum_{j=1}^{n} -c_1 \exp\left(-\frac{c_2}{2}\mu_{ij}^T\Sigma_{ij}^{-1}\mu_{ij}\right) \tag{11}$$

where

$$\begin{aligned} \mu_{ij} &= T\otimes\left(\mu_i, p\right) - \mu_j = R^T\mu_i + t - \mu_j \\ \Sigma_{ij} &= T\otimes\left(\Sigma_i, p\right) + \Sigma_j = R^T\Sigma_i R + \Sigma_j \end{aligned} \tag{12}$$

c_1 and c_2 are positive regularizing factors. Then the gradient vector can be derived by equation (11) as:

$$\begin{aligned} \frac{\partial f_{D2D}}{\partial p_a} = & \frac{c_1 c_2}{2}\left(2\mu_{ij}^T B j_a - \mu_{ij}^T B Z_a B\mu_{ij}\right) \\ & \exp\left(-\frac{c_2}{2}\mu_{ij}^T B\mu_{ij}\right) \end{aligned} \tag{13}$$

where p_a means one element in vector p and

$$\begin{aligned} B &= \left(R^T\Sigma_i R + \Sigma_j\right)^{-1} \\ j_a &= \frac{\partial\left(R^T\mu_i + t - \mu_j\right)}{\partial p_a} \\ Z_a &= \frac{\partial\left(R^T\Sigma_i R\right)}{\partial p_a} \end{aligned} \tag{14}$$

Similarly, we can obtain Hessian matrix as follows:

$$\begin{aligned} \frac{\partial^2 f_{D2D}}{\partial p_a \partial p_b} = & \frac{c_1 c_2}{2}\left(2j_b^T B j_a - 2\mu_{ij}^T B Z_b B j_a\right. \\ & + 2\mu_{ij}^T B H_{ab} - 2j_b^T B Z_a B\mu_{ij} \\ & + 2\mu_{ij}^T B Z_b B Z_a B\mu_{ij} - \mu_{ij}^T B Z_{ab} B\mu_{ij} \\ & \left. -\frac{c_2}{2}q_a q_b\right)\exp\left(-\frac{c_2}{2}\mu_{ij}^T B\mu_{ij}\right) \end{aligned} \tag{15}$$

where

$$\begin{aligned} H_{ab} &= \frac{\partial^2}{\partial p_a p_b}\left(R^T\mu_i + t - \mu_j\right) \\ Z_{ab} &= \frac{\partial^2}{\partial p_a p_b}\left(R^T\Sigma_i R\right) \\ j_b &= \frac{\partial\left(R^T\mu_i + t - \mu_j\right)}{\partial p_b} \\ Z_b &= \frac{\partial\left(R^T\Sigma_i R\right)}{\partial p_b} \\ q_a &= 2\mu_{ij}^T B j_a - \mu_{ij}^T B Z_a B\mu_{ij} \\ q_b &= 2\mu_{ij}^T B j_b - \mu_{ij}^T B Z_b B\mu_{ij} \end{aligned} \tag{16}$$

The gradient and Hessian are needed in Newton method for optimization.

3 Proposed Method

3.1 Algorithm Overview

The pipeline of our proposed method is shown in Figure 2. Two scans are collected at time k and $k+1$, denoted as Z_k and Z_{k+1} respectively, where $Z_k = \{z_{ki} \mid i = 1, 2, 3, ..., m\}$, m is the number of beams in each scan at time k.

The first step is to fit the collected data into Gaussian mixture model. In this paper, we separate each scan into cells before generate GMM. Data points $M_i = \{m_1, m_2, m_3, ..., m_{\hat{n}}\}$ in each cell i, where $\hat{n}$ is the number of points in a cell, are fitted into one normal distribution $N(\mu_i, \Sigma_i)$ via EM method. It should be noted that we consider points in a cell as one single distribution even if these points are presented to be a corner feature. Then in the following step, all of the distributions are integrated as a GMM as in equation (1).

In the similarity calculation process, we choose L_2 distance as the metric distance and adopt D2D algorithm. The difference from [Stoyanov *et al.*, 2012] is that we consider the covariance of each distribution as a geometry constraint lies between scan pairs. Then by finding the closest distribution during matching, the optimization problem can be more adaptive to a large range of initial value. During the process of iteration, we introduce a so-called dynamic scale factor which is discussed as follows.

After aforementioned procedure, the estimated transformation vector $T = [R, t]$ can be computed by general non-linear optimization methods. In this paper, we use Newton method to solve the non-linear optimization problem.

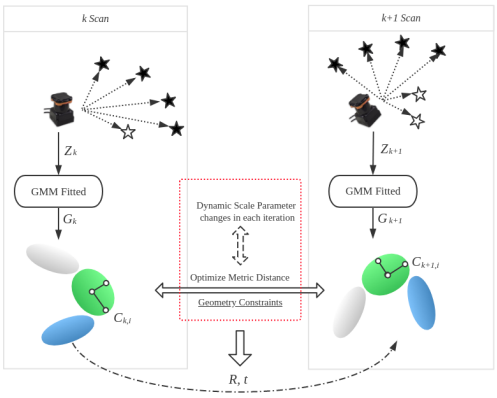

Figure 2: Pipeline of proposed scan matching method. When scans are collected from sensors, the points clouds are first fitted into a GMM. The transformation variables can then be obtained by D2D method with geometry constraint and dynamic scale factor proposed in this paper.

3.2 Geometry Constraint

In Section 2 we discussed the relationship of transformed GMM pairs. In 2D case, it can be clearly written as follows:

$$\operatorname{argmin} \sum_i \sum_j dist\left(N(\mu_i, \Sigma_i) - T\left(N(\mu_j, \Sigma_j)\right)\right)^2 \quad (17)$$

where

$$\begin{aligned} T\left(N(\mu_j, \Sigma_j)\right) &= N(\mu_i, \Sigma_i) \\ &\Rightarrow \begin{cases} \hat{\mu}_j = R\mu_j + t = \mu_i \\ \hat{\Sigma}_j = R^T \Sigma_j R = \Sigma_i \end{cases} \\ R &= \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \\ t &= \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned} \quad (18)$$

In this paper, we concentrate more on the geometry relationship of covariance because we exploit covariance of each distribution to represent the uncertainty of sensor data. In other words, the shape of the covariance is intended to describe possible range of real points' positions. Hence we do not care about the accurate coordinates of each single point but consider the potential associations between pairwise distributions.

Another reason why we consider geometry constraint is that for GMM registration, especially using L_2 distance, one significant pre-requisite is the covariance of two corresponding distributions should be similar, or it will undoubtedly give rise to a bad convergence. This deficiency exists in [Kunjin *et al.*, 2016] because NDs are fitted by each cell's points, which cannot ensure that the corresponding NDs have the same covariance. In order to get over it, we use the long axis and short axis of the covariance ellipse as a constraint.

As shown in Figure 3, denote the reference distribution as Σ_i and the transformed distribution as Σ_j. After the rotation of angle θ, it is derived according to (18) as $\Sigma_i = R^T \Sigma_j R$. If define Σ_i and Σ_j as:

$$\begin{aligned} \Sigma_i &= \begin{bmatrix} C_{i1} & 0 \\ 0 & C_{i2} \end{bmatrix} = \begin{bmatrix} \sigma_{ixx} & 0 \\ 0 & \sigma_{iyy} \end{bmatrix} \\ \Sigma_j &= \begin{bmatrix} \hat{C}_{j11} & \hat{C}_{j12} \\ \hat{C}_{j21} & \hat{C}_{j22} \end{bmatrix} = \begin{bmatrix} \hat{\sigma}_{jxx} & \hat{\sigma}_{jxy} \\ \hat{\sigma}_{jyx} & \hat{\sigma}_{jyy} \end{bmatrix} \end{aligned} \quad (19)$$

then the eigenvalues of Σ_j are calculated. The eigenvalues represent the spread in the direction of the eigenvectors, which are the variances under a rotated coordinate system. By definition a covariance matrix is positive definite therefore all eigenvalues are positive and can be seen as a linear transformation to the data.

The axes can be obtained from

$$[V, D] = \operatorname{eig}(\Sigma_j) \quad (20)$$

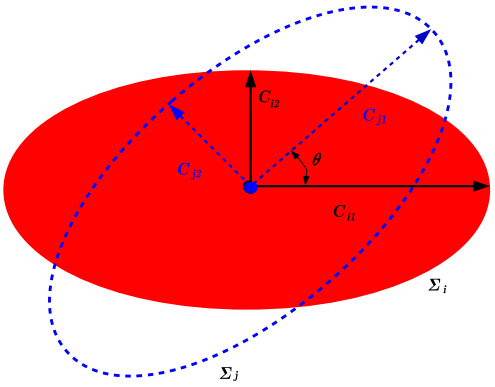

Figure 3: Transform of covariance ellipse. The ellipse filled in red is denoted as Σ_i with axes C_{i1} and C_{i2}, while the ellipse with axes C_{j1} amd C_{j2} in blue dashed line Σ_j is the corresponding observation of Σ_i after a rotation θ.

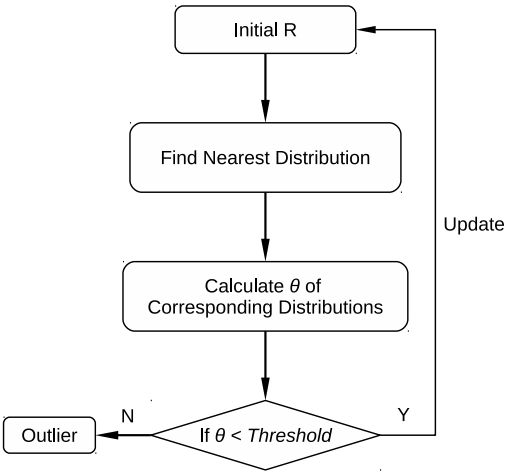

Figure 4: Flow chart of implementing geometry constraint. The correspondence of distributions can be derived when an initial rotation R is given. Then the rotation angle θ is to be calculated by eigenvalue decompression. Once we get θ, a criteria of judging outlier is adopted to remove outliers or update initial rotation R.

where D is a diagonal matrix with eigenvalue of Σ_j which should be same as Σ_i representing the axes of covariance ellipse, V is a transform matrix to rotate covariance ellipse from state i to state j. Then the rotation angle θ can be derived by transferring V to angle.

Once we get θ, we can use the angle to judge the correspondence of matched distributions and help update the initial prior rotation. The flow chart is show as Figure 4. Attempts are taken to find the nearest corresponding distributions when an initial rotation matrix R is given as an input. Then we calculate θ of each corresponding distribution pairs and compare the θ with a threshold (in our work we choose an empirical value of 0.05 rad). It should be noted that the distributions are transferred into one coordinate with initial R, hence θ should be close to 0 if the new distributions are transferred correctly with rotation R. If θ is bigger than the threshold, we just reject the corresponding distributions as outliers. Conversely, the angle θ will be added to the initial rotation matrix as a supplement value.

3.3 Dynamic scale factor

As is introduced in (11), c_1 and c_2 are supposed to be positive regularizing factors. However, it is not accurate enough to keep c_1 as a constant factor during iteration. Actually c_1 is a value in relation with estimated variable R according to its definition:

$$c_1 = \frac{1}{\sqrt{(2\pi)^k \left|\Sigma_{ij}\right|}} = \frac{1}{\sqrt{(2\pi)^k \left|R^T \Sigma_i R + \Sigma_j\right|}} \tag{21}$$

Here R is rotation matrix and k is the dimensions of the mean. Equation (11) is always reasonable, but getting the derivative of c_1 is difficult for the sake of determinant, which is the reason why we need to approximate Gradient and Hessian matrix. Different from other methods such as [Stoyanov *et al.*, 2012; Hong and Lee, 2017; Jian and Vemuri, 2011], we calculate the value of c_1 in each iteration and multiple it with a ratio coefficient ω to limit the value range of cost function, then the optimization problem can be adapted for iterating and changing of estimated variables, which is so-called dynamic scale factor as $\hat{c}_1 = \omega c_1$.

Algorithm 1 presents the proposed GMM based scan matching method with dynamic scale factor. $g(i, j, Init)$ is the expression of (21). During mapping among every components in both GMMs, we calculate changeable $\hat{c}_1$ in each iteration and revise the cost function, then using non-linear optimization method to obtain the optimized transformation R, t.

4 Experiment

A general indoor environment contains many chairs and tables. It's hard to acquire regular geometry shapes in 2D laser scanning as well as in 3D environment. For example in Figure 5, in some real indoor world chair legs and table legs are mostly common seen by a laser. The data shown in the below image is of irregular shape. Then the data is divided by cells and fitted into Gaussian

Data: Two GMMs G_1, G_2
Result: Estimated transformation R, t

```
while ( DError < Threshold ) do
    initR, initT;
    for i < G1.Ncomponents do
        for j < G2.Ncomponents do
            [id,Error]=FindNearest(G1i,G2j,initR,initT);
            ĉ1 = g(i,j,Init);
        end
    end
    [R,T] = min(CostFunction);
    #defined in equation (17);
    DError = abs(preError-Error);
    preEror=Error;
    [initR,initT]=[R,t];
    if NumIter < MaxIter then
        return
    end
end
```

Algorithm 1: GMM Based Matching with Dynamic Scale Factor

Table 1: Four tests settings

Method	Settings
GMM_uC_uD	Conventional GMM
GMM_uC_yD	√ Dynamic Scalar Factor × Geometry Constraint
GMM_yC_uD	× Dynamic Scalar Factor × Geometry Constraint
GMM_yC_yD	Our method

distributions in each cell.

Our method is compared with conventional GMM methods and implemented on core i7 CPU without GPU. At this stage we only experiment on simulation data. We first test the performance of four different settings of GMM based methods in a four-steps simulation environment, as shown in Table 1, namely GMM_uC_uD (Conventional GMM method, [Jian and Vemuri, 2011; Stoyanov *et al.*, 2012]), GMM_uC_yD (GMM method with Dynamic Scalar Factor, without Geometry Constraint), GMM_yC_uD (GMM method with Geometry Constraint, without Dynamic Scalar Factor), and GMM_yC_yD (Our method) with four different transformations $[2, 3, \pi/4]$, $[-1, 5, -\pi/5]$, $[-2, 1, \pi/6]$, $[2.5, 2.1, -\pi/10]$. As shown in Figure 6, the position 1 is the initial position. The ellipses are one GMM fitted from the scans. Arrows represent the direction of robot's heading. At this stage we assume observations in each step are precise corresponding and no missing data exists.

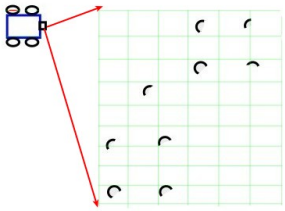

Figure 5: 2D scanning of chairs and tables. The data collected from 2D laser is generally presented as shown in the below image. The data is irregular compared with that of flat walls, boxes or pillars.

Table 2: Relative Errors (m) of different methods

Transform	GMM_yC_yD	GMM_uC_yD	GMM_uC_uD	GMM_yC_uD
Step 1	1.0e-03 * 0.8822	0.0001	0.0003	0.0024
Step 2	1.0e-03 * 0.8525	0.0028	0.0003	0.0053
Step 3	1.0e-03 * 0.3996	0.0053	0.0022	0.0093
Step 4	1.0e-03 * 0.4261	0.0051	0.0020	0.0090

Table 2 gives the relative errors in each steps of transformation. It is clear that our method (GMM_yC_yD) provide a more accurate result compared with other three GMM methods. Dramatically the result of GMM_uC_yD performs a little bit worse than GMM_uC_uD. The reason is that in this experiment we set the factor c_1 to 0.1, which is suitable for this model. If we set the factor c_1 to 10, however, we get an opposite result. Hence it is an evidence that dynamic scale factor plays a positive role in GMM based scan matching.

A circular trajectory motion is simulated to further compare the performance. Figure 7 illustrates the simulation of robot movement. The robot moves around a circle with the center at point $(3, 3)$ and a radius of 6m. Three points clouds are considered as features existed in the environment.

First we need to fit the points clouds into one Gaussian mixture model with the assumption of knowing components number. Then we test four GMM based methods as well as ICP method.

As shown in Figure 8, all of the methods can obtain

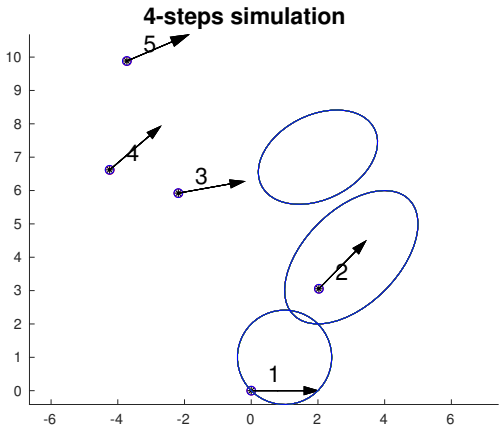

Figure 6: The four step simulation for comparing the performance of different GMM based methods. The position 1 is the initial position. The ellipses are one GMM fitted from the scans collected at the initial position. Arrows represent the direction of robot's heading.

Table 3: RMSE (m) of different methods

ICP	GMM_yC_yD	GMM_uC_yD	GMM_uC_uD	GMM_yC_uD
0.1318	**0.1317**	0.1322	0.1727	0.1573

relatively good estimation compared with ground truth. Then we calculate the root-mean-squared-error (RMSE) and the results are listed in Table 3.

From the results of RMSE, it is clear that although all of the RMSE are small enough, our method (GMM with Geometry Constraint and Dynamic Scale Factor) arrives in a tiny win compared with ICP method. The reason may lie on the approximated ellipses in calculation. Then compared in pairwise, for example, GMM_yC_yD vs. GMM_yC_uD and GMM_yC_yD vs. GMM_uC_yD, it can be concluded that both Geometry Constraint and Dynamic Scale Factor can increase the accuracy slightly and Dynamic Scale Factor makes more contribution on the improvement of performance.

Figure 9 depicts the range of convergence. From the upper two pictures we can conclude that Dynamic Scale Factor is able to improve the efficiency of convergence because the cost function with Dynamic Scale Factor have a smaller minimum, as is noted with red lines intersection. But the convergence ranges (red circle on the surface with projection to x-y plane) of both situations are nearly the same. From the left two graphs we come to the conclusion that Geometry Constraint is able to enlarge the range of descend area, which is more robust responding to a relatively bad initial value.

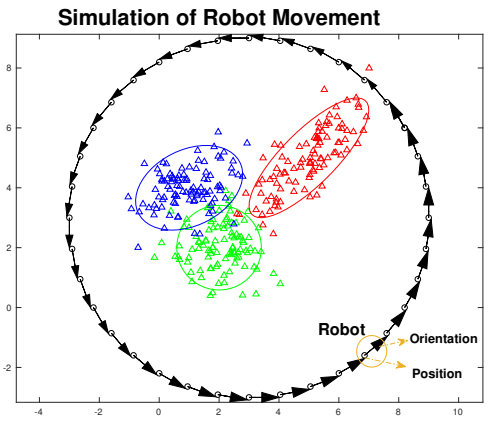

Figure 7: Simulation scenario. The position of robot is represented by a black circle while the orientation is represented by an arrow. The colored triangles are scan points describing features in the environment. Three different colors means the points cloud are divided into three normal distributions and fitted as one Gaussian mixture model, which is depicted as colored ellipses intuitively.

5 Conclusion

In this paper, we proposed a GMM based scan matching method with geometry constraint and dynamic scale factor. Geometry constraint can provide a prior information for selecting nearest distribution and help update the initial rotation, while dynamic scale factor makes the cost function more accurate to describe the model. Simulation experiments demonstrate that our method can improve the accuracy of the result. In future research we will concentrate on extend the method to 3D making it more general and use more experimental datasets to evaluate the performance.

References

[Besl and McKay, 1992] Besl Paul J. and Neil D. McKay. Method for registration of 3-D shapes. *Sensor Fusion IV: Control Paradigms and Data Structures.*, Vol. 1611. International Society for Optics and Photonics, 1992.

[Biber and Wolfgang, 2003] Peter Biber and Straer Wolfgang. The normal distributions transform: A new approach to laser scan matching. *IIntelligent Robots and Systems, 2003 IEEE/RSJ International Conference on. IEEE,*,Vol. 3. 2003.

[Takeuchi and Tsubouchi, 2006] Eijiro Takeuchi and Takashi Tsubouchi. A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping. *Intelligent Robots and Sys-*

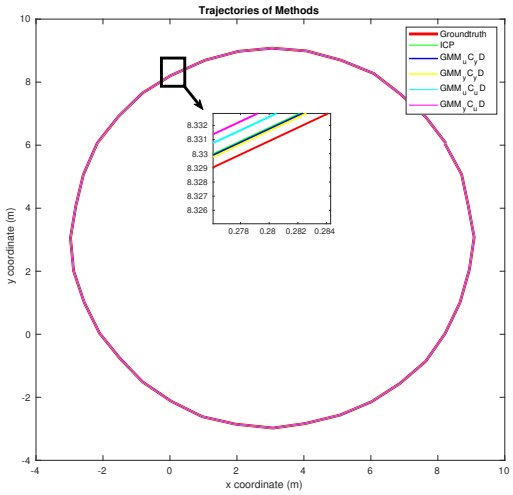

Figure 8: Estimated trajectories of different methods.

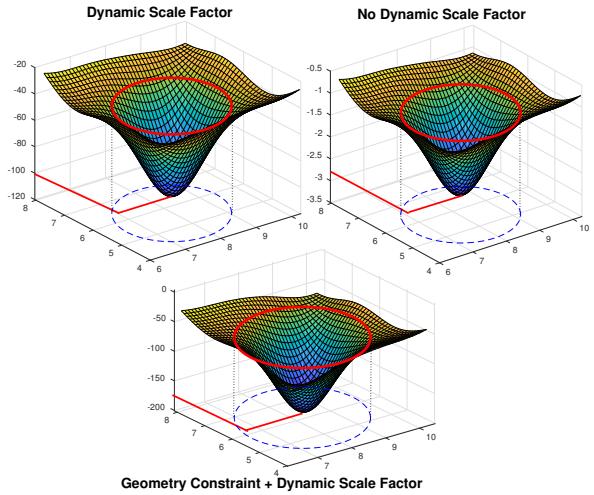

Figure 9: Cost function and the range of convergence. The top left is with dynamic scale factor while the top right is without dynamic scale factor. The figure below shows the cost function of our method.

tems, 2006 IEEE/RSJ International Conference on. IEEE,(pp. 3068-3073)

[Magnusson, 2009] Martin Magnusson The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection. *Diss. rebro universitet*2009.

[Jian and Vemuri, 2011] Bing Jian and Baba C. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 33.8 (2011): 1633-1645.

[Stoyanov *et al.*, 2012] Todor Stoyanov, Martin Magnusson, Henrik Andreasson, and Achim J. Lilienthal. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *he International Journal of Robotics Research*, 31, no. 12 (2012): 1377-1393.

[Kunjin *et al.*, 2016] Ryu Kunjin, Lakshitha Dantanarayana, Tomonari Furukawa, and Gamini Dissanayake. Grid-based scan-to-map matching for accurate 2D map building. *Advanced Robotics*, 30, no. 7 (2016): 431-448.

[Myronenko and Song, 2010] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32.12 (2010): 2262-2275.

[Li *et al.*, 2018] Liang Li, Ming Yang, Chunxiang Wang, and Bing Wang. Robust Point Set Registration Using Signature Quadratic Form Distance. *IEEE transactions on cybernetics*, 99 (2018): 1-13.

[Pu *et al.*, 2018] Can Pu, Nanbo Li, Radim Tylecek, and Robert B. Fisher. DUGMA: Dynamic Uncertainty-Based Gaussian Mixture Alignment. *arXiv preprint*, arXiv:1803.07426 (2018).

[Hong and Lee, 2017] Hyunki Hong and B. H. Lee. Probabilistic normal distributions transform representation for accurate 3D point cloud registration. *In Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 3333-3338. IEEE, 2017.

[Das and Waslander, 2014] Arun Das and Steven L. Waslander. Scan registration using segmented region growing NDT. *The International Journal of Robotics Research*, 33, no. 13 (2014): 1645-1663.