DUGMA: Dynamic Uncertainty-Based Gaussian Mixture Alignment

Can Pu
can.pu@ed.ac.uk

Nanbo Li
nanbo.li@ed.ac.uk

Radim Tylecek
rtylecek@inf.ed.ac.uk

Robert B. Fisher
rbf@inf.ed.ac.uk
School of Informatics, University of Edinburgh

Abstract

Accurately registering point clouds from a cheap low-resolution sensor is a challenging task. Existing rigid registration methods failed to use the physical 3D uncertainty distribution of each point from a real sensor in the dynamic alignment process. It is mainly because the uncertainty model for a point is static and invariant and it is hard to describe the change of these physical uncertainty models in different views. Additionally, the existing Gaussian mixture alignment architecture cannot efficiently implement these dynamic changes.

This paper proposes a simple architecture combining error estimation from sample covariances and dynamic global probability alignment using the convolution of uncertainty-based Gaussian Mixture Models (GMM). Firstly, we propose an efficient way to describe the change of each 3D uncertainty model, which represents the structure of the point cloud better. Unlike the invariant GMM (representing a fixed point cloud) in traditional Gaussian mixture alignment, we use two uncertainty-based GMMs that change and interact with each other in each iteration. In order to have a wider basin of convergence than other local algorithms, we design a more robust energy function by convolving efficiently the two GMMs over the whole 3D space.

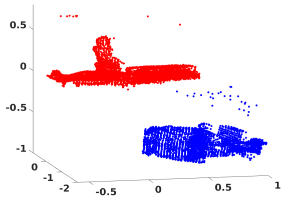
Tens of thousands of trials have been conducted on hundreds of models from multiple datasets to demonstrate the proposed method's superior performance compared with the current state-of-the-art methods. All the materials including our code are available from `https://github.com/Canpu999/DUGMA`.

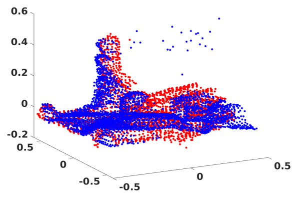
1. Introduction

With recent improvements in depth sensing devices and algorithms, 3D point clouds are more accessible to researchers. However, using an expensive high-precision 3D scanner to get accurate and large-scale 3D point clouds is still not popular. Accurate alignment of noisy and partial point clouds with many outliers from cheap low-resolution sensors is still a core technique in various fields, such as computer vision, robotics, virtual and augmented reality.

Finding the accurate transformation between two noisy rigid point clouds is generally hard: true point-to-point correspondences seldom exist, which limits the accuracy of the methods based on ICP [29, 32, 23, 1, 5, 12, 21]. As for the methods based on local descriptors [16, 30, 22, 25], coarse points tend to cause inaccurate local descriptors to mismatch with each other. Also, the variable density (distant areas have a lower point cloud density) will make them unstable. Aligning probabilistic models can effectively mitigate the problems above. Thus, many researchers have been exploring different kinds of probabilistic models [24, 3, 31, 15, 19, 26] to represent the real surface structure.

However, as far as we can tell, no one has incorporated the physical 3D uncertainty distribution information for each point from a real sensor into the probabilistic model, which allows describing the real surface structure more accurately. The challenges mainly lie in two parts: the first is how to get the real uncertainty distribution information from the real sensors. In the recent years, an increasing number of researchers have been investigating how to estimate the uncertainty of the acquired data for different sensors, such as the Kinect sensor [20], the time of flight sensor [7], the structure from motion sensor [10] and the stereo vision sensor [18]. These suggest using physical noise models for each point to represent their individual occurrence probability in 3D space. The second challenge is how to use physical uncertainty information for each


(a) Before registration


(b) After registration

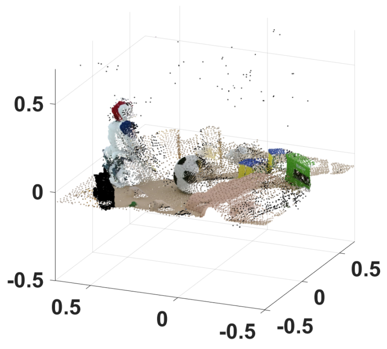

(c) After registration (texture mapped)

Figure 1: In this work, DUGMA incorporates the 3D uncertainty distribution of each 3D point from a sensor into a dynamic Gaussian mixture alignment system. The figure shows a critical example of aligning two noisy and partial 3D scans with many outliers and different densities from two real low-resolution scanners using our algorithm.

point from each specific view in the registration process. Specifically, if we use a Gaussian function with a covariance to represent the uncertainty of one point in the 3D space, the covariance should change with each different coordinate system in each iteration. Thus, the registration process is dynamic. Moreover, the use of the covariance estimated from different viewpoints leads to position estimates that are compatible with each physical covariance. After that, we build a bridge to make two point clouds interact with each other by sharing information, which is obviously different from traditional Gaussian mixture alignment [15, 4]. The GMM of the fixed point cloud in their methods is invariant and can't share their state with the GMM of the moving point cloud, reducing the registration accuracy and also making the usage of physical uncertainty models unavailable.

In this paper, we propose a simple architecture combining error estimation from sample covariances and dual dynamic global probability alignment using the convolution of uncertainty-based GMMs from point clouds in the whole 3D space. Firstly, we propose an effective way to describe the change of each 3D uncertainty model in the dynamic registration process, which represents the structure of the point cloud much better. Unlike the invariant GMM (representing the fixed point cloud) in traditional Gaussian mixture alignment, we use a dynamic uncertainty-based GMM for each point set, which interact in each iteration. To be less susceptible to local minima, we define a more robust energy function by convolving the two dynamic GMMs over the whole 3D space rather than use time-consuming optimization methods, such as branch and bound [29, 4, 24]. The proposed dual dynamic uncertainty-based GMM's alignment can be optimized efficiently by the EM algorithm [8, 2], which experimentally shows a wider basin of convergence than other local algorithms. A new empirical approximation will be proposed to reduce the amount of calculation drastically.

The rest of this paper is organized as the following. In Section 2, key previous registration algorithms will be reviewed briefly and also recent advances of the methods for estimating the uncertainty of the acquired data from different sensors. In Section 3, the dynamic uncertainty-based Gaussian mixture alignment theory is presented. In Section 4, tens of thousands of trials have been conducted on multiple real datasets through simulation, which is more systematic testing than that done for the compared algorithms. Also, we show real application tests with most recent and advanced algorithms. Our accuracy improvement comes from the following key contributions:

Key contributions:

1) Incorporation of the invariant 3D uncertainty distribution information (represented by a Gaussian function with a physical covariance) into the dynamic registration;

2) A bridge to make the two point clouds interact with each other by creating a novel point proximity weight term;

3) A more robust energy function and efficient approximation to the optimization step that greatly reduces computational complexity.

2. Related Work

In 1992, Besl and McKay [1] first introduced the Iterative Closest Point (ICP) algorithm to compute the rigid transformation between two point clouds by minimizing the Euclidean distance between the corresponding points. Since then, a large number of variants have been proposed and the reader could be directed to the survey [21]. To enhance robustness to noise, Segal et al. proposed Generalized-ICP [23] in 2009, which considered the probability distribution of each point but was still based on binary correpondence search. To be robust to occlusion and small partial overlap, researchers [32] built bilateral correspondence using bidirectional distances. To widen the convergence zone, GO-ICP [29] used a branch-and-bound method to avoid getting stuck in local minima. Exact point-to-point correspondences seldom exist and the correspondence definition (two points have the minimum distance rather than are in the same place) is coarse, which makes it hard for the ICP family to achieve accurate results.

The second class of the alignment approaches is feature-based methods, which first extract and match local descriptors (e.g. FPFH [22], SHOT [25]) from two point clouds and then estimate the relative pose using random sampling [22], RANSAC [11], Hough transforms [27], etc. Recently, Zeng et al. [30] used a siamese neural network to learn a local patch descriptor in a volumetric grid to establish the correspondences between two point clouds. Similarly, Elbaz et al. [9] used a deep neural network auto-encoder to encode local 3D geometric structures instead of traditional descriptors. Lei et al. [16] proposed a fast descriptor based on eigenvalues and normals computed from multiple scales to extract the local structure of the point clouds and then recovered the transformation from matches. However, they are sensitive to noisy point clouds. Additionally, the density of the point cloud influences the extraction of local descriptors and even makes them completely break down if the density is too low.

Aligning probabilistic models which represent the structure of the point cloud can efficiently mitigate the problems above. Our method belongs to this class. One key factor to an accurate and robust registration is the data representation used. Since 2003, many approaches based on a variety of probabilistic models have been explored to represent the structure of the point cloud such as Robust Point Matching [6], Kernel Correlation [26], Coherent Point Drift [19]. In 2011, GMMREG [15] used two Gaussian mixture models with the same isotropic covariances for each point and minimized the L_2 distance between the two GMMs to get the transformation. The GMM which represented the fixed point cloud was regarded as invariant and thus could not receive the current registration state from the other point cloud. In 2014, Zhou et al. [31] proposed to use the Student's-t mixture model to represent the point cloud in the registration algorithm. In 2015, Campbell et al. [3] used a Support Vector Machine to learn and construct SVGM (a GMM with non-uniform weights) to represent the point cloud. In the next year, Campbell et al. used SVGM [3] and the architecture in GMMREG [15] to get the globally-optimal transformation using a branch and bound approach in order not to be vulnerable to local minimum. Recently, Straub et al. [24] used a Dirichlet process Gaussian mixture (DP-GMM) and a Dirichlet process von Mises-Fisher mixture (DP-vMF-MM) to represent the geometric information of the point cloud. The mathematical probabilistic model used to represent the point cloud has become considerably complex. Nevertheless, we will use the physical 3D uncertainty distribution from real sensors to construct a simple uncertainty-based GMM to represent the structure of the point cloud, which will fit the real surface geometry better.

The acquisition of a physical 3D uncertainty distribution for each point from a real sensor is a difficult task. With the development of different depth sensors, more effective uncertainty estimation methods for various sensors have been designed. In 2012, Nguyen et al. [20] used the distance and angle between the Kinect sensor and observed surface to estimate both axial and lateral noise distributions. In 2013, Engel et al. [10] used the geometric disparity error and photometric disparity error for the structure from motion sensor to estimate 3D point error. Recently, many researchers [7, 18] have estimated the uncertainty for the ToF (Time of Flight) sensor based on the physical properties of the sensor (eg. the IR frequency). Meanwhile, [18] developed an empirical uncertainty model for the stereo vision sensor based on the relationship between the local cost and global cost.

In summary, there are previously developed methods for both robust rigid point cloud registration and modelling the 3D uncertainty distribution of the points in a 3D point cloud. This paper improves registration accuracy and robustness using an approach that combines these two themes and redesigns a dynamic Gaussian mixture alignment system using invariant 3D uncertainty information from each point cloud.

3. Methodology

First we introduce the change of 3D uncertainty distribution, then build a bidirectional dynamic bridge between the two point clouds, and finally introduce the framework of our math model. Table 1 lists some of the symbols and their notations.

Table 1: Symbols & Notations

Symbol	Notation
$\mathbf{X}, \mathbf{Y}$	Two point clouds
D	Dimensionality of the point clouds
N, M	Number of points in X, Y point cloud
x_n	One point in X point cloud
y_m	One point in Y point cloud
Σ_{x_n}, Σ_{y_m}	Covariance for point x_n and y_m
$\mathbf{I}$	Identity matrix
$\mathbf{0}$	Column vector of all zeros

3.1. Change of 3D Uncertainty Distribution

We will use a Gaussian function with a covariance to represent the distribution of one point in 3D space. The specific covariance for each point represents the physical 3D uncertainty distribution for that point from a real sensor.

(1) If a point with covariance Σ_{orig} has been rotated by $\mathbf{R}$, then Σ will be $\Sigma = \mathbf{R}\Sigma_{orig}\mathbf{R}'$.

(2) A scaling factor for the covariance of a point is proportional to the average minimum distance σ between two point clouds to ensure that the probability of all the points in the other point cloud will not become too small when the two point clouds are far away from each other. See Algorithm 1.

3.2. Dynamic Gaussian Mixture Alignment

The Gaussian function $g_{x_n}(\tau)$ of the point x_n predicts the probability that x_n appears at the position τ in its own coordinate system. Based on Gaussian weights around each point, we will define a probability-like function that not only depends on the distribution of the point (represented by isotropic or anisotropic covariance) but also whether a possible corresponding point cp_{x_n} in the other point cloud is nearby. We model the presence of a corresponding point by a weight function $w_{x_n}(\tau, cp_{x_n})$ that has significant value only when a potential corresponding point cp_{x_n} from point cloud $\mathbf{Y}$ is near the position τ. A similar definition holds for $g_{y_m}(\tau)$, $w_{y_m}(\tau, cp_{y_m})$. Thus either point cloud can receive and send current state information from or to the other point cloud bi-directionally to evaluate the current registration quality.

In the analysis below, we assume the $\mathbf{Y}$ point cloud has been already transformed from the initial point cloud $\mathbf{Y_0}$ by rotation $\mathbf{R}$ and translation $\mathbf{t}$ (which then become the domain for the optimization of the evaluation function). The product $g_{x_n}(\tau)g_{y_m}(\tau)$ represents x_n, y_m appearing at the same position τ in the same coordinate system. Thus, it encodes the underlying prior knowledge, ie. x_n, y_m are possible corresponding points from two point clouds. In other words, any two points from the fixed and moving point cloud can be a corresponding pair in our system and the likelihood depends on the probability of correspondence that x_n, y_n appear at the same position τ, which is different from soft assignment [13] in essence.

3.3. The description of our model

Based on the previous discussion, we design the uncertainty-based GMM as follows:

$$G_{\mathbf{X}}^{\mathbf{I},\mathbf{0}}(\tau) = \sum_{n=1}^{N} w_{x_n}(\tau, cp_{x_n}) g_{x_n}(\tau), \tag{1}$$

$$G_{\mathbf{Y}}^{\mathbf{R},\mathbf{t}}(\tau) = \sum_{m=1}^{M} w_{y_m}(\tau, cp_{y_m}) g_{y_m}(\tau), \tag{2}$$

where the gaussian kernels are given by

$$g_{x_n}(\tau) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_{x_n}|}} e^{-\frac{1}{2}(\tau - x_n)^T \Sigma_{x_n}^{-1} (\tau - x_n)} \tag{3}$$

$$g_{y_m}(\tau) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_{y_m}|}} e^{-\frac{1}{2}(\tau - y_m)^T \Sigma_{y_m}^{-1} (\tau - y_m)} \tag{4}$$

$$w_{x_n}(\tau, cp_{x_n}) = e^{-\frac{1}{2}(cp_{x_n} - \tau)^T \Sigma_{x_n}^{-1} (cp_{x_n} - \tau)} \tag{5}$$

$$w_{y_m}(\tau, cp_{y_m}) = e^{-\frac{1}{2}(cp_{y_m} - \tau)^T \Sigma_{y_m}^{-1} (cp_{y_m} - \tau)} \tag{6}$$

$G_{\mathbf{X}}^{\mathbf{I},\mathbf{0}}(\tau)$ denotes the GMM from the fixed point cloud $\mathbf{X}$ and $G_{\mathbf{Y}}^{\mathbf{R},\mathbf{t}}(\tau)$ represents the GMM from the moving point cloud $\mathbf{Y}$ after rotation $\mathbf{R}$ and translation $\mathbf{t}$. Thus $y_m = \mathbf{R} y_{m0} + \mathbf{t}$, $\Sigma_{y_m} = \mathbf{R}\Sigma_{y_{m0}}\mathbf{R}'$, $|\Sigma_{y_m}| = |\mathbf{R}\Sigma_{y_{m0}}\mathbf{R}'| = |\Sigma_{y_{m0}}|$ due to $|\mathbf{R}| = 1$. $\Sigma_{y_m}^{-1} = (\mathbf{R}\Sigma_{y_{m0}}\mathbf{R}')^{-1} = \mathbf{R}\Sigma_{y_{m0}}^{-1}\mathbf{R}'$ due to $\mathbf{R}\mathbf{R}' = \mathbf{I}$. At all times, $x_n = x_{n0}$ and $\Sigma_{x_n} = \Sigma_{x_{n0}}$. Each point has its own covariance.

Integrating the product of the two dynamic GMMs (representing the overlapping effect of the two point clouds) over the whole 3D space, as we shall show in our experiments, makes the energy function more robust, accurate and have a wider convergence basin compared with [19, 15, 4].

We now formulate the optimization over rotation $\mathbf{R}$ and translation $\mathbf{t}$ as an EM-like process. First, an energy function is defined as the following

$$E = \int P(\tau) \log \left[G_{\mathbf{X}}^{\mathbf{I},\mathbf{0}}(\tau) G_{\mathbf{Y}}^{\mathbf{R},\mathbf{t}}(\tau) \right] \mathrm{d}\tau, \tag{7}$$

where τ integrates over all the domain of the point clouds; $P(\tau)$ is the probability that there is a point at the position τ. We design it as the sum of the probability that all the possible corresponding pairs appear at the position τ in

$$P(\tau) = \sum_{i=1}^{N} \sum_{j=1}^{M} P(\tau, x_i, y_j) = \sum_{i=1}^{N} \sum_{j=1}^{M} g_{x_i}(\tau) g_{y_j}(\tau). \tag{8}$$

Equation (7) can be rewritten as

$$E = \int P(\tau) \log \left[\sum_{i=1}^{N} \sum_{j=1}^{M} F^{\mathbf{R},\mathbf{t}}(\tau, x_i, y_j) \right] \mathrm{d}\tau, \tag{9}$$

with a combined term

$$F^{\mathbf{R},\mathbf{t}}(\tau, x_i, y_j) = w_{x_i}(\tau, cp_{x_i}) g_{x_i}(\tau) w_{y_j}(\tau, cp_{y_j}) g_{y_j}(\tau). \tag{10}$$

By the definitions above, the weight term $w_{x_i}(\tau, cp_{x_i})$ is nearly zero when point x_i is far from any point in $\{y_j\}$. This allows us to avoid having to compute correspondences by using all y_j in place of cp_{x_i} (and similarly for cp_{y_j}) and simplify $F^{\mathbf{R},\mathbf{t}}$ with

$$\tilde{F}(\tau, x_i, y_j) = w_{x_i}(\tau, y_j) g_{x_i}(\tau) w_{y_j}(\tau, x_i) g_{y_j}(\tau). \tag{11}$$

We maximize Equation (9) to get the estimated rotation and translation by minimizing its negative. We adopt the EM algorithm [8, 2] to solve for $\mathbf{R}$, $\mathbf{t}$. Its main idea is: guess the values of the parameters firstly in the last iteration (denoted by 'old') and calculate the posteriori probability $P^{old}(x_i, y_j|\tau)$ using Bayes' theorem then, which corresponds to the expectation stage. After that, minimize the expectation of the negative log-likelihood function $\mathcal{L}$ to find the parameters in the current iteration (denoted by 'new'), which corresponds to the maximization stage. Thus, we get

$$\mathcal{L} = -\int \sum_{i=1}^{N} \sum_{j=1}^{M} B(\tau, x_i, y_j) \log(\tilde{F}^{new}(\tau, x_i, y_j)) \, \mathrm{d}\tau, \tag{12}$$

$$B(\tau, x_i, y_j) = P(\tau) \, P^{old}(x_i, y_j \mid \tau). \tag{13}$$

Neglecting the constant term and using $P(\tau) \approx P^{old}(\tau)$, we simplify the target function to

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{j=1}^{M} \int P^{old}(\tau, x_i, y_j) \, Mah^{new}(\tau, x_i, y_j) \, \mathrm{d}\tau, \tag{14}$$

where a term similar to Mahalanobis distance is obtained:

$$\begin{aligned} Mah^{new}(\tau, x_i, y_j) = &\frac{1}{2}(\tau - x_i)^T(\Sigma_{x_i}^{-1} + \Sigma_{y_j}^{-1})(\tau - x_i) \\ &+ \frac{1}{2}(\tau - y_j)^T(\Sigma_{x_i}^{-1} + \Sigma_{y_j}^{-1})(\tau - y_j). \end{aligned} \tag{15}$$

As we will justify below, there is no real benefit to integrate the whole 3D space, because the values of the Gaussian functions are only significant near the data points themselves. In fact, because most values are quite low, we approximate the integral by a sum at only the data points to speed up the algorithm drastically (unlike [4]). Thus we need evaluate only each term $P^{old}(\tau, x_i, y_j)Mah^{new}(\tau, x_i, y_j)$ at the positions of x_i and y_j, which will reduce the time complexity greatly to only $\mathcal{O}(MN)$. Applying this simplification, the approximated energy function becomes

$$\tilde{\mathcal{L}} = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{\tau\in\{x_i,y_j\}} P^{old}(\tau, x_i, y_j)\, Mah^{new}(\tau, x_i, y_j). \tag{16}$$

Expanding the last sum and uniting like terms we get

$$\tilde{\mathcal{L}} = \sum_{i=1}^{N}\sum_{j=1}^{M} C_{i,j}^{old} \cdot (y_j - x_i)^T(\Sigma_{x_i}^{-1} + \Sigma_{y_j}^{-1})(y_j - x_i), \tag{17}$$

where

$$\begin{aligned} C_{i,j}^{old} = &(2\pi)^{-D}|\Sigma_{x_i}|^{-\frac{1}{2}}|\Sigma_{y_j}|^{-\frac{1}{2}} \\ &\left(e^{-\frac{1}{2}(y_j - x_i)^T\Sigma_{x_i}^{-1}(y_j - x_i)} + e^{-\frac{1}{2}(x_i - y_j)^T\Sigma_{y_j}^{-1}(x_i - y_j)}\right). \end{aligned} \tag{18}$$

The x_i, y_j, Σ_{x_i} and Σ_{y_j} in $C_{i,j}^{old}$ are from the previous iteration. We then minimize Equation (17) over the rotation $\mathbf{R}$ and translation $\mathbf{t}$ domain, using interior point optimization [28] as summarized in Algorithm 1.

Algorithm 1 DUGMA Point Cloud Registration.

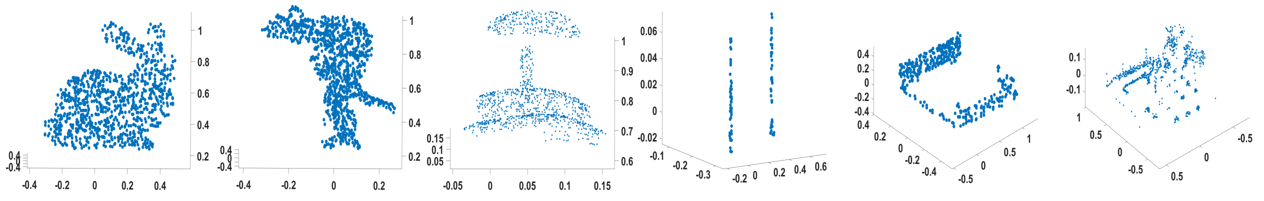
Input: Two point clouds $\mathbf{X}$, $\mathbf{Y}$ and their covariances Σ_x, Σ_y, initial transformation $\mathbf{R} = \mathbf{I}$, $\mathbf{t} = \mathbf{0}$.

```
1: EM-like optimization, repeat until convergence:
2: procedure E-STEP                                        ▷ Update $\mathbf{Y}$, $\sigma$, $\Sigma_y$, $C_{i,j}^{old}$
3:     $\mathbf{Y} \leftarrow \mathbf{RY} + \mathbf{t}$
4:     $\sigma \leftarrow \frac{1}{M}\sum_{j=1}^{M} d_{min}(y_j, \mathbf{X})$      ▷ Minimum distance
5:     $\Sigma_y \leftarrow \sigma\ \mathbf{R}\Sigma_y\mathbf{R}'$
6:     $C_{i,j}^{old} \leftarrow$ compute Eq. (18)
7: procedure M-STEP                                        ▷ Solve for $\mathbf{R}$,$\mathbf{t}$
8:     Use interior point algorithm to solve Eq. (17):
9:     $(\mathbf{R}, \mathbf{t}) \leftarrow \arg\min_{\mathbf{R},\mathbf{t}} \tilde{\mathcal{L}}$
```

4. Experiments

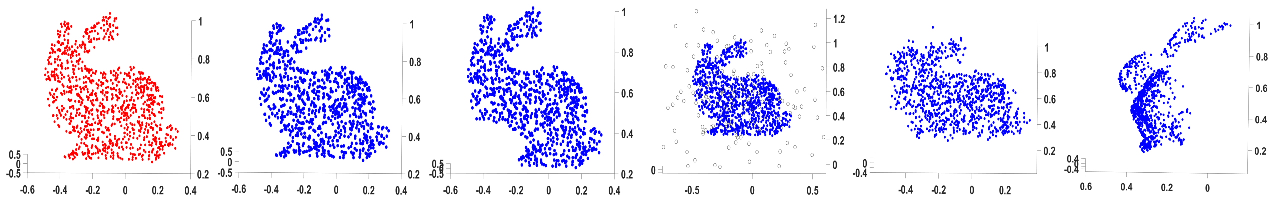
We implemented our algorithm using Matlab and Cuda C++. We ran all the algorithms on a laptop with Intel Core i7-7820HK processor (quad-core, 8MB cache, up to 4.4GHZ) and NVidia Geforce GTX 1080 with 8GB GDDR5X. To test the accuracy and robustness of our algorithm, our proposed method is compared with relevant recent algorithms from the top journals and conferences: CPD [19], GMMREG [15], BDICP [32], GOICP [29], GOGMA [4], 3DMATCH [30][1], FDCP [16][2]. All the code is directly from the authors. We did not compare ours with [9] because we could not get our re-implemented algorithm to work well based on their partial released code. The Stanford 3D Scanning Repository [17] and our new 3D dataset have been used to do performance comparison of the algorithms. After that, 30 real scenes with ground truth from multiple Kinect sensors in our new dataset[3] have been used to show the approach works on par or better in a real application compared with the rest.

[1]We only compared ours with 3DMATCH in the Kinect data application with their pre-trained weights.
[2]FDCP has compared their results with [24] so we neglected [24].
[3]https://github.com/Canpu999/DUGMA



(a) Bunny (b) Armadillo (c) Drill (d) Pole (e) Court (f) Garden

Figure 2: six example 3D models, (a)(b)(c) from Stanford 3D Scanning Repository, (d)(e)(f) from our new dataset



(a) Model (b) Sampling rate 85% (c) Occlusion 10% (d) Outliers 200 (e) Noise standard deviation=0.1 (f) Initial rotation angle in x,y,z=(0°,30°,30°)

Figure 3: Different influences from various factors.

4.1. Simulation

To synthesize the two point clouds to register, we randomly choose a model from the datasets above for two point clouds firstly. Then a different random large segment of each point cloud is removed completely to simulate occlusion. After that, the two occluded models are sampled differently, which simulates the resolution of different sensors in real scenarios. Also, different anisotropic Gaussian noise with random standard deviations and zero mean has been added to each point to simulate the complex noise in real environments resulting from known and unknown factors. The variances of all the noise on each axis have been stored in the covariances accurately. Next, outliers have been added into both point clouds to simulate outliers acquired by the sensors. Finally, an initial rigid transformation is applied to the moving point cloud.

The experiments are divided into four groups given the four influence factors or variables: noise, outliers, occlusion, and initial rotation. In each group of experiments, one controlled variable will be changed and the values of the other variables will be picked randomly from a default range. The experiment is conducted 3 times at each controlled value for each of 100 shapes with a random perturbation each time, see Algorithm 2. The maximum iteration value for all is 100. For FDCP, we set `gridStep`=1.5 and `Rho`=0.1 to make it robust to different densities. For GOGMA we set the scale parameter for SVM (0.5,0.5) to limit GOGMA's running time to around 100 seconds per registration. For GOICP, Mean Squared Error (MSE) convergence threshold `MSEThresh`=0.2. The rest of the parameters share default values in their open code.

We use $||\mathbf{t_{gt}} - \mathbf{t_{est}}||_F$, $||\mathbf{I} - \mathbf{R_{gt}}\mathbf{R_{est}^{-1}}||_F$ [14] to estimate the quality of the registration, where $\mathbf{R_{gt}}, \mathbf{t_{gt}}$ are the ground truth and $\mathbf{R_{est}}, \mathbf{t_{est}}$ are estimated results respectively and $|| \bullet ||_F$ is the Frobenius norm.

Algorithm 2 Controlled and random variables process. For each method, 14700 trials have been done.

```
1: for $controlled\_variable := start$ by $step$ to $end$ do
2:     for $shape := 1$ to 100 do
3:         for $instance := 1$ to 3 do
4:             Produce data with controlled and random
5:               variable
6:             Do registeration (different algorithms)
7:             Calculate the registration error
```

From the Stanford 3D Scanning Repository (50) and our new dataset (50) we got 100 models from various views of different objects and scenes. Each was downsampled to about 1000 points with different densities. Figure 2 shows 6 example models from different scenes and objects.

We apply different effects to simulate the real factors in the real environment. Figure 3 shows examples of the effects. In our experiment, the sampling rate is set to 90% and 85% for the fixed and moving point cloud, respectively. Table 2 gives specific information about the parameters.

Table 2: Range for random and controlled factors

Factor	*random range*	*controlled range*
Initial rotation	[-20°, 20°]	[-60°, 60°]; step=8°
Outliers	[0, 500] that is, [0, ≈33%]	[0, 2000] that is, [0, ≈67%]; step=200
Noise standard deviation	[0, 0.2] × radius of point cloud	[0, 0.3] × radius of point cloud; step=0.03
Occlusion	[0, 15%]	[0, 30%]; step=0.03

Figure 4 shows one successful registration in a real garden. After registration, we could see the hedges and trees overlap well although there is a big patch of occlusion in both two point clouds, many outliers and noise.

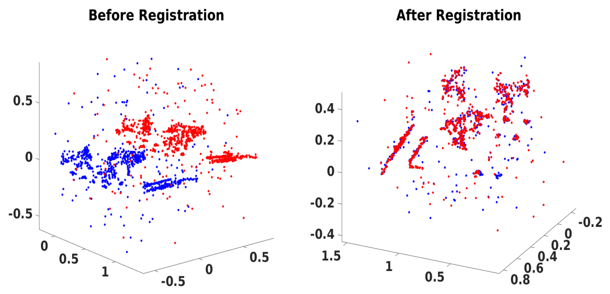


Figure 4: A successful registration in a real garden.

When the initial rotation angle value is the controlled variable, it ranges from [-60°, 60°], with an 8° step. In the experiments, the specific rotation angle around each axis is chosen as 0 or the initial rotation angle value randomly. Figure 5 shows that beyond -40°or 40°, the proposed algorithm breaks down because the iteration count exceeds the maximum. But within [-40°, 40°], our algorithm is much more stable and accurate compared with the rest. In Figures 5,6,7,8, 'Time' refers to the average running time per registration. If rotation and translation error is below 0.2 and 0.1 respectively in a trial, the trial is a success (third plot). When occlusion rate is the controlled variable, Figure 6 shows within 25%, the proposed algorithm performs well.

Judging that GOGMA needs much more time (about 1000 sec for a trial) to achieve a much better performance and behaved poorly in the experiments above, we will neglect GOGMA in the remaining noise and outlier experiments but later compare the proposed algorithm with it in the small dataset registration experiment.

When outliers are the controlled variable, we use covariances generated in the same manner as the true data points. Figure 7 shows the proposed algorithm has superior performance again. When the noise level is the controlled variable, Figure 8 shows robust and accurate performance compared with the rest.

4.2. Real data from multiple Kinect sensors

The simulation experiments above show our algorithm works well with very accurate covariance estimates. In the real case, it is hard to get very accurate covariance estimates. In this real application, we estimate an inaccurate covariance for each 3D point from a Kinect sensor to test our algorithm. We design the uncertainty of each valid 3D point acquired by the

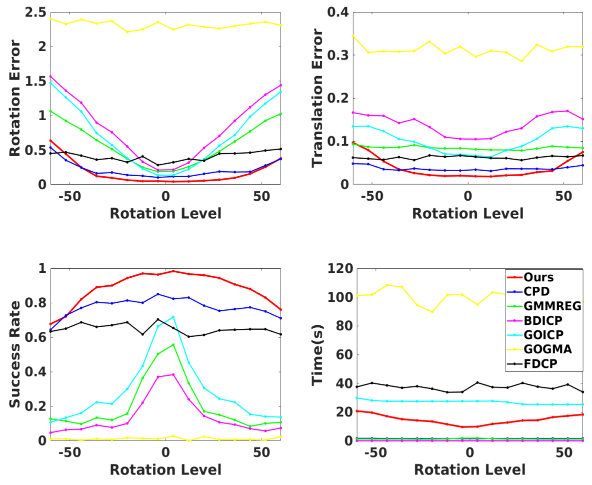


Figure 5: Rotation Experiment

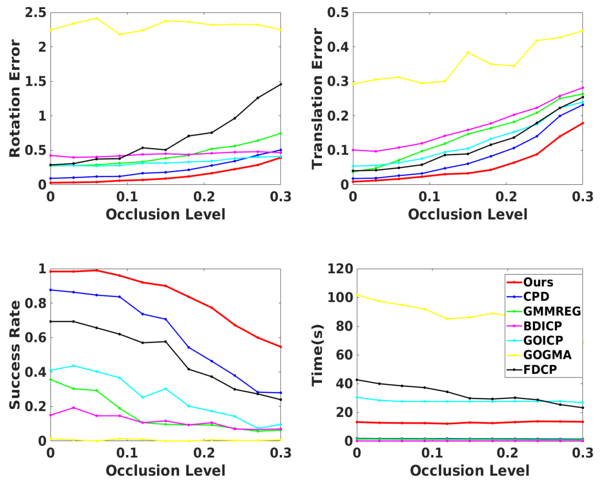


Figure 6: Occlusion Experiment

Kinect sensor based on the depth value d and the angle α between the camera and the normal of the surface [20].

$$U(\alpha, d) = exp[w_1(1 - \cos\alpha) + w_2 d] \tag{19}$$

We use $w_1 = 1.6658$ and $w_2 = 0.2776$ by letting $U(\pi/3, 0) = U(0, 3) = 2.2$. The number 2.2 is set manually and the algorithm works well if that number is in the range [1,10] (known by our experiments). Then we simply multiplied the uncertainty and the identity matrix to estimate a coarse covariance for each point. Future work will explore more accurate real covariances to represent the 3D uncertainty distribution.

We tested our algorithm using two point clouds from two Kinect sensors. The ground truth of the rotation and translation between the two Kinect sensors is known by calibration. Figure 1 (a), (b) show the scene before and after registration using our algorithm. Figure 1 (c) adds the colour texture information into the two registered point clouds.

In the experiment, the two point clouds have been downsampled to ∼4K points or so using the grid average method. The downsampling was small from ∼20K to ∼4K (rather than 640*480 to ∼4K): the Kinect scan was cropped to the fixed scenes. Then we applied the same initial rotation to all the algorithms and reduced the scale parameter for SVM (0.08, 0.08) in GOGMA and `MSEThresh`=0.001 in GOICP to make them get their best performance. Here we present results from 30 scenes in our new dataset. We calculated the error and running time based on only successful registrations (rotation and

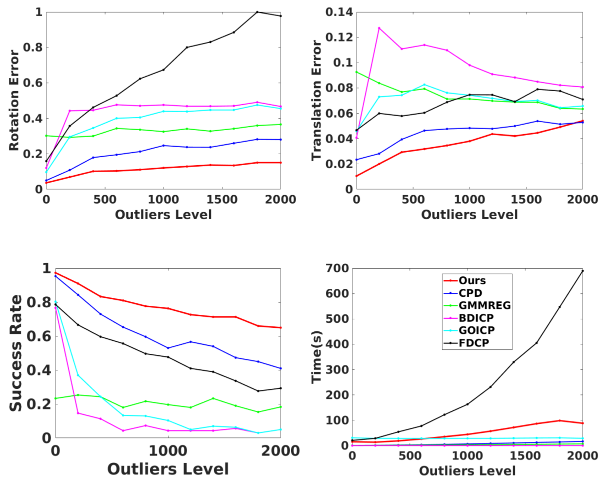


Figure 7: Outlier Experiment

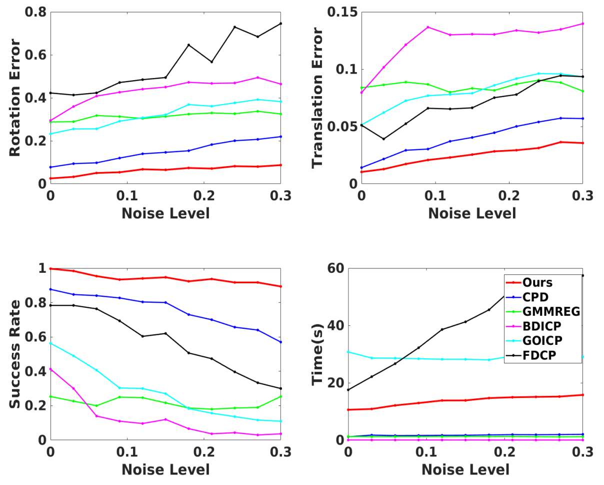


Figure 8: Noise Experiment

translation error is below 0.2 and 0.1). After all the algorithms have converged, our successful rate (most important) ranks first (96%). The estimated mean rotation of our algorithm (0.04) is lowest, see Table 3. Otherwise, our algorithm is about 7 times faster than GOGMA whose success rate (93%) ranks second.

4.3. Additional Experiments

We have also done 14700 similar trials using 100 2D fish models from the Gatorbait100[4] database and have used the other 2D point clouds with 100 different shapes (face, umbrella, computer etc.) to test our sensitivity to shapes. We have used 100*5 2D models from Gatorbait100 database with 5 different densities to test our sensitivity to density. All the results are equally robust. We have also tested our method without any uncertainty information and replaced each covariance matrix with an identity matrix. The results show our method with uncertainty information is better than that without uncertainty and both are better than the other comparison algorithms. For more details, see our supplementary materials `https://github.com/Canpu999/DUGMA`.

[4]http://www.rvg.ua.es/graphs/dataset01.html

Table 3: Experiment Results for the Real Application

Method	Error Mean (R)	Error Std (R)	Error Mean (t)	Error Std (t)	Suc. rate	Time s/trial
CPD	0.05	0.03	**0.03**	**0.01**	50%	42.0
Gmmreg	-	-	-	-	0	-
BDICP	-	-	-	-	0	-
GOICP	-	-	-	-	0	-
GOGMA	**0.04**	0.03	**0.03**	0.02	93%	1125
3dmatch	0.08	0.04	0.04	0.02	23%	**6.6**
FDCP	0.06	**0.02**	0.04	**0.01**	40%	8.2
Ours	**0.04**	0.03	0.04	0.02	**96%**	163

5. Conclusion

The proposed algorithm is simpler and more effective than the previous algorithms. We incorporated the 3D uncertainty distribution into a simple dynamic Gaussian mixture alignment. The obvious difference between our algorithm and the previous ones is that it needs covariances at each point as input, which requires error models of how to estimate the real covariance for each kind of sensor. All the experiments we have done show that the proposed method is very robust and accurate and works well. In the future, we will only use the points from set $\mathbf{Y}$ in the neighbourhood of x_i to approximate all the points in set $\mathbf{Y}$ in Equation (17) to reduce the time complexity to $\mathcal{O}(N)$.

6. ACKNOWLEDGMENTS

The research is funded by TrimBot2020 project [Grant Agreement No. 688007] in the European Union Horizon 2020 programme. We thank Dylan Campbell ([4, 3]), Andy Zeng [30], Huan Lei [16], Jihua Zhu [32] for providing help when we did the comparison experiments. We thank Dylan Campbell, Marcelo Saval-Calvo for giving us good advice on this paper.

References

[1] P. J. Besl, N. D. McKay, et al. A method for registration of 3-d shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992. 1, 3

[2] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995. 2, 5

[3] D. Campbell and L. Petersson. An adaptive data representation for robust point-set registration and merging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4292–4300, 2015. 1, 3, 11

[4] D. Campbell and L. Petersson. Gogma: Globally-optimal gaussian mixture alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5685–5694, 2016. 2, 5, 6, 11

[5] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 545–548. IEEE, 2002. 1

[6] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003. 3

[7] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo. Probabilistic tof and stereo data fusion based on mixed pixels measurement models. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2260–2272, 2015. 1, 3

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. 2, 5

[9] G. Elbaz, T. Avraham, and A. Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *CVPR*, 2017. 3, 6

[10] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1456, 2013. 1, 3

[11] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[12] A. W. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13):1145–1153, 2003. 1

[13] S. Gold, C.-P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. In *Advances in Neural Information Processing Systems*, pages 957–964, 1995. 4

[14] D. Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009. 7

[15] B. Jian and B. C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, 2011. 1, 2, 3, 5, 6

[16] H. Lei, G. Jiang, and L. Quan. Fast descriptors and correspondence propagation for robust global point cloud registration. *IEEE Transactions on Image Processing*, 2017. 1, 3, 6, 11

[17] M. Levoy, J. Gerth, B. Curless, and K. Pull. The stanford 3d scanning repository. *URL http://www-graphics. stanford. edu/data/3dscanrep*, 2005. 6

[18] G. Marin, P. Zanuttigh, and S. Mattoccia. Reliable fusion of tof and stereo depth driven by confidence measures. In *European Conference on Computer Vision*, pages 386–401. Springer, 2016. 1, 3

[19] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010. 1, 3, 5, 6

[20] C. V. Nguyen, S. Izadi, and D. Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 524–530. IEEE, 2012. 1, 3, 9

[21] 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, 3

[22] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009. 1, 3

[23] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435, 2009. 1, 3

[24] J. Straub, T. Campbell, J. P. How, and J. W. Fisher, III. Efficient global point cloud alignment using bayesian nonparametric mixtures. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3, 6

[25] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010. 1, 3

[26] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *European conference on computer vision*, pages 558–569. Springer, 2004. 1, 3

[27] O. J. Woodford, M.-T. Pham, A. Maki, F. Perbet, and B. Stenger. Demisting the hough transform for 3d shape recognition and registration. *International Journal of Computer Vision*, 106(3):332–341, 2014. 3

[28] S. J. Wright. *Primal-dual interior-point methods*. SIAM, 1997. 6

[29] J. Yang, H. Li, D. Campbell, and Y. Jia. Go-ICP: a globally optimal solution to 3D ICP point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2016. 1, 2, 3, 6

[30] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017. 1, 3, 6, 11

[31] Z. Zhou, J. Zheng, Y. Dai, Z. Zhou, and S. Chen. Robust non-rigid point set registration using student's-t mixture model. *PloS one*, 9(3):e91381, 2014. 1, 3

[32] J. Zhu, D. Wang, X. Bai, H. Lu, C. Jin, and Z. Li. Registration of point clouds based on the ratio of bidirectional distances. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 102–107. IEEE, 2016. 1, 3, 6, 11