MLMD: Maximum Likelihood Mixture Decoupling for Fast and Accurate Point Cloud Registration

Ben Eckart[1,2] Kihwan Kim[2] Alejandro Troccoli[2] Alonzo Kelly[1] Jan Kautz[2]
[1]The Robotics Institute, Carnegie Mellon University [2]NVIDIA Research

Abstract

Registration of Point Cloud Data (PCD) forms a core component of many 3D vision algorithms such as object matching and environment reconstruction. In this paper, we introduce a PCD registration algorithm that utilizes Gaussian Mixture Models (GMM) and a novel dual-mode parameter optimization technique which we call mixture decoupling. We show how this decoupling technique facilitates both faster and more robust registration by first optimizing over the mixture parameters (decoupling the mixture weights, means, and covariances from the points) before optimizing over the $6DOF$ registration parameters. Furthermore, we frame both the decoupling and registration process inside a unified, dual-mode Expectation Maximization (EM) framework, for which we derive a Maximum Likelihood Estimation (MLE) solution along with a parallel implementation on the GPU. We evaluate our MLE-based mixture decoupling (MLMD) registration method over both synthetic and real data, showing better convergence for a wider range of initial conditions and higher speeds than previous state of the art methods.

1. Introduction

Estimation of the relative pose between two point clouds forms the basis of many algorithms in 3D vision applications such as 3D object matching, SLAM, body/head pose estimation, and medical imaging. Typically, many of these applications use some form of the Iterative Closest Point (ICP) algorithm [1, 4], an iterative procedure that cycles between trying to find correspondences from one cloud (or mesh) to the other, and minimizing the summed distance between all previously established correspondences. Since the correspondences may not be correct, the algorithm iterates between matching and minimizing, with the hope that successive iterations will provide more and more accurate correspondences and therefore the minimization of the summed distance will provide the correct pose displacement.

However, in practice, the original ICP algorithms tend to converge poorly when subjected to severe noise and large pose displacements without a good initial guess. To make the ICP algorithm robust to noise, outliers, and sampling differences, many variants of the classic ICP algorithm have been introduced over the last decade [22].

The Softassign algorithm [10] was the first in a new class of methods that framed ICP into a statistical framework. Developed as an effort to replace the *ad hoc* procedure of correspondence matching and distance minimization with a theoretical footing in statistics, algorithms in this class are derived from general assumptions about the nature of point clouds, their underlying noise properties, and how the minimization step relates to some type of energy function or probabilistic estimate. Thus, these methods facilitate robust outlier rejection, proofs of convergence, and various continuous optimization procedures with differentiable objective functions.

Under a statistical framework, the iteration of correspondences and minimization of summed distances can be seen as a type of Expectation Maximization (EM) procedure [6]. By establishing the connection between ICP's point matching step with calculating expectations over latent correspondence variables (E Step), and ICP's distance minimization step with the maximization of some bound on the data likelihood (M Step), many of the EM-based formulations actually generalize back to ICP under various basic statistical assumptions about the input data [23, 11]. Additionally, if each point is given a Gaussian noise parameter, the total point cloud can be interpreted as a type of Gaussian Mixture Model (GMM). Thus, most robust registration techniques explicitly utilize a GMM representation for point cloud data (PCD) to derive claims and proofs about robustness and convergence [5, 8, 11, 12, 20]. Unfortunately, these algorithms tend to be much more complex than ICP, and their robustness often comes at a high computational and representational cost.

Our main contribution is a novel statistical algorithm that provides a fast, compact, and accurate solution for 3D PCD registration. We build on the idea of using Gaussian Mixtures to represent point clouds and EM to find our registration solution. However, to reduce the computational and representational cost of our method, we include an addi-

tional optimization step before the actual registration occurs to decouple the mixtures from the points ("*mixture decoupling*"). At the same time, we estimate non-uniform mixture weights and anisotropic covariances, according to the principle of maximum data likelihood. This is in contrast to previous techniques that utilize per-point Gaussians, uniformly weight each mixture, and restrict covariances to be the same isotropic value per point, which we show to be ill-posed in the case of range data exhibiting high degrees of anisotropic uncertainty. Finally, we show how the dual-mode optimizations (mixture decoupling and registration) can be unified under a common EM framework and how exploiting this relation can both improve robustness and drive a highly parallel implementation on the GPU.

1.1. Background and Motivation

For most GMM-based methods, given a point cloud with N points $\mathbf{z}_i \in \mathbb{R}^3$, the probability of an arbitrary point in space $x \in \mathbb{R}^3$ is defined to be,

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}(x|\mathbf{z}_i, \sigma^2\mathbf{I}) \tag{1}$$

where $\mathcal{N}(x|\mathbf{z}_i, \sigma^2\mathbf{I})$ is the multivariate Gaussian with mean and covariance $\{\mathbf{z}_i, \sigma^2\mathbf{I}\}$. Thus, the means are the point locations of the modeled point cloud, and all covariances are controlled by a single isotropic bandwidth parameter, σ^2. Implicit in this construction is that the mixture vector, π_i is uniformly constant for every point ($\pi_i = \frac{1}{N}$). In other words, if each point has Gaussian uncertainty, then the collection of these points as a PDF (the linear combination of their individual Gaussian PDF's) forms a GMM. The GMM, being a continuous and valid PDF, thus forms the foundation for robust probability estimates, noise handling, and EM-based energy minimization or likelihood maximization methods for registration.

Though registration algorithms designed around the GMM/EM concept have been established in the literature to be much more robust than ICP [11], ICP still remains the industry standard for 3D PCD alignment in vision algorithms due to its simplicity and speed. One problematic aspect of existing GMM/EM registration methods stems from the choice of representation. Though it is theoretically desirable to assign a covariance around each point, this representation is now much more complex than the original point cloud and now has a number of means equal to the number of points. Furthermore, point-based GMM approaches operate under the assumption that each point should have a corresponding point (match) in the other cloud, when, in fact, sampling differences make this unlikely. The fundamental assumption that a point is generated by a single Gaussian source doesn't actually correspond to the reality of non-uniformly scanned geometry, which is that points belong to surfaces, not other points.

Intuitively, it should be noted that point samples representing pieces of the same local geometry could be compressed into smaller clusters, with the local geometry encoded inside the covariance of that cluster. Thus, if the point cloud is of size N, it is possible to adequately describe it by a smaller set of J means and covariances.

1.2. Mixture Decoupling

The former discussion motivates the need to add a preprocessing step by first finding a compact but representative model before attempting to establish correspondences or minimize log-likelihood. This is because, in terms of computation, the complexity of EM is dependent on the size of the model. Furthermore, the accuracy and descriptive power of the model directly affects the convergence rate and robustness of the algorithm. Therefore, we propose to *optimize with respect to the representation first* for the express purpose of aiding registration.

We do this by modifying the normal EM procedure to first maximize the model data over the set of all possible GMMs of a given size J, denoted as $\mathbf{\Theta}$, without restriction to the structure of the covariance, the placement of the means, and the mixture weighting. Given point clouds $\mathcal{Z}_1, \mathcal{Z}_2$ and some unknown transformation $T(\cdot)$ representing the spatial relationship between them, this forms a dual-step optimization problem for the form,

$$\textbf{Step 1: } \hat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}}{\operatorname{argmax}}\, p(\mathcal{Z}_1|\mathbf{\Theta}) \tag{2}$$

$$\textbf{Step 2: } \hat{T} = \underset{T}{\operatorname{argmax}}\, p(T(\mathcal{Z}_2)|\hat{\mathbf{\Theta}}) \tag{3}$$

After mixture decoupling (Eq. 2-Step 1), we end up with a more descriptive, compact representation of the original data. Though this preprocessing step does not come for free (unlike the traditional approaches described in Section 1.1), as long as the model is sufficiently compact and well-suited to the data, the entire process can actually be sped up considerably. Fig. 1 shows this situation pictorially. If each point cloud is of size N, a typical GMM/EM registration algorithm will contain N^2 potential connections and thus scale poorly. If we first compress our cloud to size J, where $J \ll N$, then we have two different optimization procedures of size NJ. Given that $J \ll N$, we will be able to produce a method that is much more efficient overall ($O(N^2)$ to $O(N)$). For the experiments in this paper we have used $J = 16$, which is empirically chosen for the best performance.

2. Related Work

Since the advent of ICP in 1992 [1, 4], there have been hundreds of papers published on the topic of point cloud registration. Much early work focused on the direct improvement of ICP itself [22] through more robust, nonlinear optimization steps. Some notable works include point

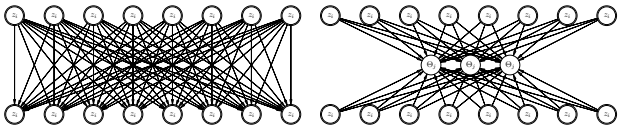

Figure 1: **Mixture Decoupling** *Left:* Given two similarly sized point clouds of size N, point-based GMMs produce N^2 potential matches. *Right:* Decoupling points from mixtures into a smaller, more descriptive set of latent $\mathbf{\Theta}$ produces two separate procedures of size $O(JN)$. If $J \ll N$, this effectively linearizes the registration process with respect to N.

to quadratic surface approximations [19] and robust M-Estimators for Euclidean distance [9] solved by Levenberg-Marquadt optimization. Since this paper is focused on GMM and EM-based techniques, we will not cover the vast amount of the ICP literature, but will focus instead only on those techniques that use statistical models based on GMMs and/or solved via EM, which we call GMM/EM techniques as shorthand.

The first statistical algorithm to multiply link correspondences was Softassign [10]. Softassign used a simulated annealing scheme, borrowed from statistical physics to add additional robustness. Shortly afterwards Mixture Points Matching (MPM) [5] and EM-ICP [11] derived multiple links from Gaussian noise assumptions. Both authors noted how point sets with Gaussian noise were equivalent to a Gaussian Mixtures if each point was interpreted as being generated by some Gaussian with a set isotropic covariance. Thus, the probability measure of another point cloud having been generated by the same Gaussian Mixture set is well-defined and its rigid transformation can be solved in closed form for isotropic covariances. Also, using annealing for robustness, these methods established the process of interpreting PCD as a Gaussian Mixture with each point representing a mixture mean with an associated (isotropic) covariance. It should be noted that, similar to our work, EM-ICP also recognized the inherent scalability problems of point-level GMMs and thus devised a very primitive form of mixture decoupling using a sphere decimation technique.

Tsin and Kanade derived a correlation-based approach called Kernel Correlation (KC) [27], followed by Jian and Vemuri with GMMReg [15, 14]. The latter method minimizes GMM-to-GMM L_2 distance, a metric that is heavily related to Tsin and Kanade's correlation-based approach. Instead of utilizing an EM-based framework, however, these methods directly optimize over a cost function containing all N^2 point pairs. Since there is no explicit correspondence step (as in EM), these methods have difficulty scaling. Tsin and Kanade attempted to get around the scalability problems by numerical differentiation over grids, while GMMReg is restricted to small or subsampled point sets. Maintaining all point-point correspondences, however, results in a very robust algorithm since Gaussian point-pair "distances" act as robust loss functions.

Later, the Coherent Point Drift (CPD) [20, 21] algorithm was introduced as a method similar to EM-ICP, having both explicit E and M Steps, but with the addition of isotropic covariance estimation into the M Step in lieu of the annealing technique used by SoftAssign, MPM, EM-ICP, and GMMReg. ECMPR [12] would later extend CPD's covariance estimation to fully anisotropic covariances, solving the optimization via Expectation Conditional Maximization or ECM [18], an EM-like framework that performs several M-Steps in succession after each E Step, holding all parameters but the variable to be optimized constant.

Like CPD and ECMPR, Generalized ICP (G-ICP) [23] extended ICP to include covariance estimation. Functioning as a bridge between methods like ICP and EM-ICP, G-ICP uses the nearest neighbor criterion of ICP along with a generalized M Step that, depending on the choice of the type of covariance estimation, behaves like point-to-point ICP, point-to-plane ICP, or a plane-to-plane ICP. However, instead of incorporating covariance estimation into their M Step, they estimated it separately by looking at local (20-nn) neighborhoods around each point.

Building off the work of [3, 2] for Normal Distance Transforms (NDT), a 3D-NDT method for registration was introduced, and later further refined in [24, 25]. Similar to G-ICP's data-driven covariance estimation apart from the registration optimization itself, the NDT is a way to produce a smaller set of means and covariances through voxelization. Instead of neighborhoods around each point producing a covariance estimate, as in G-ICP, NDT methods simply voxelize and then record the mean and covariance of points that fall within that voxel. As opposed to point-based GMM constructions, the NDT is a voxel-based GMM construction and thus can provide large time savings, which is especially important for many time-critical applications. The 3D-NDT shares similarities to our proposed work in that it first establishes a compact model from the data that is then operated over for the purpose of registration. In contrast to our work, however, they do not employ EM, but instead use the voxel occupation of the projected scene or Euclidean distance to voxels to establish correspondence (most versions therefore are singly linked, though [25] uses the 4-nearest neighbors), and then the implicit M-Step is solved through Newton's method over a modified GMM-to-GMM L2 distance. Thus, one can then interpret the 3D-NDT methods in terms of a GMMReg variant that first performs a type of mixture decoupling through voxelization.

More recently, REM-Seg [7] and JRMPC [8] were established as algorithms that decouple the means from the points and do so without resorting to voxelization like the NDT methods. The former uses a small set of anisotropic covari-

Method	Year	Mult. Link	Prob. Corr.	Cov. Est.	Aniso-tropic	Com-pact
ICP [1]	1992					
SoftAssign [10]	1998	✓	✓			
MPM [5]	2000	✓	✓			
EM-ICP [11]	2002	✓	✓			
KC [27]	2004	✓				
GMMReg [14]	2005	✓				
CPD [21]	2006	✓	✓	✓		
3D-NDT [17]	2007			✓	✓	✓
G-ICP [23]	2009			✓	✓	
ECMPR [12]	2011	✓	✓	✓	✓	
NDT-D2D [24]	2012	✓		✓	✓	✓
REM-Seg [7]	2013	✓		✓	✓	✓
JRMPC [8]	2014	✓	✓	✓		
Ours (MLMD)		✓	✓	✓	✓	✓

Table 1: **A Comparison of Probabilistic Registration Methods.** *Multiply Linked*: many-to-one or many-to-many correspondences (robustness under non-uniform sampling), *Probabilistic Correspondences*: fully probabilistic correspondences or not (as opposed to kernel-based or nearest neighbor approaches). Allows for application of EM, GEM, or ECM. *Covariance Estimation*: Improves convergence as opposed to statically set covariances or simulated annealing methods. *Anisotropic*: non-spherically shaped mixture covariance. Most earlier methods restrict covariances to be uniformly isotropic across all mixtures. This improves robustness through better local shape alignment. *Compactness*: Given PCD of size N, compact models are those that have a number of mixtures $J \ll N$. This addresses scalability problems with point-based GMM representations.

ances (as we do), and the latter uses a much larger number of isotropic mixtures (J=0.6N). REM-Seg follows a similar preprocessing structure, using EM to derive a small set of anisotropic mixtures, but unlike our method, does not utilize the EM framework for registration, instead minimizing the cost function directly using gradient descent (similar to NDT, GMMReg, and KC). JRMPC finds the GMM jointly from all available data in batch, interleaving the optimization of the model with the calculation of the transformation parameters under an ECM framework. In contrast, we perform the optimization of the model to completion first before recovering the pose transformation. We view our construction to be more robust since, before the geometry has been fully defined, the transformation calculation could be ill-posed. Furthermore, JRMPC's lack of anisotropy precludes compactness, a necessary component for scalability.

3. Approach

The GMM/EM formulation allows one to recast the correspondence problem of ICP into probabilistic terms, opening the door to soft correspondences based on relative likelihoods, as well as maximum likelihood optimization procedures or minimum energy functions. By explicitly introducing latent correspondences, the EM framework factorizes the joint likelihood so that the log likelihood has no exponentials in it at all, thus greatly simplifying the optimization procedure over methods that do not operate under the EM framework [7, 24, 27, 14].

Our method builds on the GMM/EM concept, and through mixture decoupling, offers large reductions in computational expense for both representation and registration. Furthermore, as we later show in Section 4, the added descriptive power of using general anisotropic covariances, mean-point decoupling, and floating mixture weights adds robustness to pose displacements, noise, and offers a more compact description of point cloud geometry.

In this Section, we discuss how mixture decoupling and registration can be unified under a dual-mode EM framework. Then, we derive several accurate MLE approximation techniques for models with general anisotropic covariance parameters since the exact MLE equation with anisotropic covariances has no closed form solution. These approximations along with our technique of mixture decoupling form a highly efficient Generalized EM (GEM) procedure [6] for maximum likelihood point cloud registration.

3.1. Overview

In general, directly optimizing over the GMM parameters, $\boldsymbol{\Theta}$, and the registration parameters, T is intractable. To solve this problem we introduce latent correspondence variables, enabling the EM algorithm as a means to iteratively produce estimates of these parameters. The E Step finds the maximum likelihood estimate of points-to-cluster correspondences. The correspondences are then used to drive a tractable optimization procedure over the both the model and registration parameters in succession. We denote the M Step over GMM parameters $\boldsymbol{\Theta}$ as $M_{\boldsymbol{\Theta}}$ and the M Step over registration parameters T as M_T. Fig. 2 shows the diagrammatic depiction of the proposed algorithm.

The process of decoupling cluster means from the points is intimately related to the process of finding the pose transformation for registration. Fig. 2 shows our dual-step process: we perform the same E Step during both model optimization and registration, but utilize two different M Steps according to the state of convergence of the algorithm. We first iterate over $\hat{\boldsymbol{\Theta}}$ using $M_{\boldsymbol{\Theta}}$. Once this process has converged, we introduce the second point cloud, but now we iterate over jointly coupled M Step that solves an MLE approximation for the registration parameters. For 3D PCD, previous work has shown that $M_{\boldsymbol{\Theta}}$ can be solved efficiently in parallel and in closed form ([7]). Since M_T is jointly coupled to $M_{\boldsymbol{\Theta}}$, as we will show, it too can be efficiently computed in parallel. Note that for additional computational efficiency, we transform $\boldsymbol{\Theta}$ and not $\mathcal{Z}$. Upon convergence,

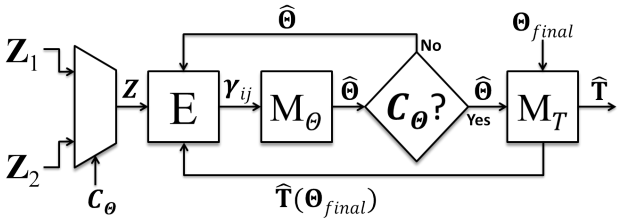

Figure 2: **Algorithm Flow** The algorithm first iterates over E and $M_{\boldsymbol{\Theta}}$ until convergence, a condition denoted by the binary variable $C_{\boldsymbol{\Theta}}$. Once $M_{\boldsymbol{\Theta}}$ converges under $\mathcal{Z}_1$, the algorithm then switches to the second point cloud for input (shown through the one-bit multiplexer). In order to produce an optimization criterion of size J, the $M_{\boldsymbol{\Theta}}$ Step result then feeds into the M_T Step along with the last $\hat{\boldsymbol{\Theta}}$ from $\mathcal{Z}_1$ as $\boldsymbol{\Theta}_{final}$. The algorithm then iterates accordingly, finding new transformation updates $\hat{T}$, which are applied to $\boldsymbol{\Theta}_{final}$.

the concatenation of all $\hat{T}$ produce the correct relative transformation between the two point clouds, $\mathcal{Z}_1$ and $\mathcal{Z}_2$.

3.2. E and $M_{\boldsymbol{\Theta}}$ Steps

The first E and $M_{\boldsymbol{\Theta}}$ Steps iterate over a point cloud $\mathcal{Z} = \{\mathbf{z}_i\}, \forall i \in \{1..N\}$ in order to solve the maximum data likelihood problem over a set of Gaussian Mixture parameters, $\boldsymbol{\Theta} = \{\pi_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}, \forall j \in \{1..J\}$. Since we cannot maximize this probability in closed form, we introduce latent binary correspondences $\mathcal{C} = \{c_{ij}\}$ for each point, cluster tuple $\{\mathbf{z}_i, \boldsymbol{\Theta}_j\}$. We adopt the parallelized EM algorithm of Eckart and Kelly [7] to perform this optimization process efficiently.

In the *E Step*, we calculate in parallel the posterior for all $c_{ij} \in \mathcal{C}$ given $\boldsymbol{\Theta}$:

$$E[c_{ij}] = \frac{\pi_j \mathcal{N}(\mathbf{z}_i|\boldsymbol{\Theta}_j)}{\sum_{j'=1}^{J} \pi_{j'} p(\mathbf{z}_i|\boldsymbol{\Theta}_{j'}) + \frac{\pi_{J+1}}{\eta}} \tag{4}$$

where η and π_{J+1} control a special "noise cluster" to filter outliers.

In the $M_{\boldsymbol{\Theta}}$ *Step*, we maximize the expected log-likelihood with respect to $\boldsymbol{\Theta}$, using our current $E[c_{ij}] \stackrel{\text{def}}{=} \gamma_{ij}$:

$$\max_{\boldsymbol{\Theta}} \sum_{ij} \gamma_{ij} \{\ln \pi_j + \ln p(\mathbf{z}_i|\boldsymbol{\Theta}_j)\} \tag{5}$$

Given a fixed set of expectations, one can solve for the optimal parameters in parallel in closed form at iteration k:

$$\hat{\boldsymbol{\mu}}_j = \frac{\sum_i \gamma_{ij} \mathbf{z}_i}{\sum_i \gamma_{ij}} \tag{6}$$

$$\hat{\boldsymbol{\Sigma}}_j = \frac{\sum_i \gamma_{ij} \mathbf{z}_i \mathbf{z}_i^T}{\sum_i \gamma_{ij}} - \hat{\boldsymbol{\mu}}_j \hat{\boldsymbol{\mu}}_j^T \tag{7}$$

$$\hat{\pi}_j = \sum_i \frac{\gamma_{ij}}{N} \tag{8}$$

Thus, given a set number of mixtures, J, we can iterate these two steps until convergence, leaving us with a compact set of mixtures representing the original PCD, decoupled from the original points.

3.3. M_T Step

In this section we show how the relationship between $M_{\boldsymbol{\Theta}}$ and M_T can provide an efficient technique for recovering the registration parameters. As in the E-$M_{\boldsymbol{\Theta}}$ stage (Sec. 3.2), we introduce correspondences to allow the optimization problem to factorize. For shorthand, we denote our moving point cloud as $\tilde{\mathcal{Z}} \stackrel{\text{def}}{=} T(\mathcal{Z})$. The full joint probability is then,

$$\ln p(\tilde{\mathcal{Z}}, \mathcal{C}|\boldsymbol{\Theta}) = \sum_{i=1}^{N} \sum_{j=1}^{J} c_{ij} \{\ln \pi_j + \ln \mathcal{N}(\tilde{\mathbf{z}}_i|\boldsymbol{\Theta}_j)\} \tag{9}$$

We transform the problem into an iterative algorithm, alternating between finding γ_{ij} (the same E Step as before) and the maximizing the expected likelihood,

$$\hat{T} = \operatorname*{argmax}_T E_{p(\mathcal{C}|\tilde{\mathcal{Z}}, \boldsymbol{\Theta})}[\ln p(\tilde{\mathcal{Z}}, \mathcal{C}|\boldsymbol{\Theta})] \tag{10}$$

$$= \operatorname*{argmax}_T \sum_{ij} \gamma_{ij} \{\ln \pi_j + \ln \mathcal{N}(\tilde{\mathbf{z}}_i|\boldsymbol{\Theta}_j)\} \tag{11}$$

$$= \operatorname*{argmin}_T \sum_{ij} \gamma_{ij} ||\tilde{\mathbf{z}}_i - \boldsymbol{\mu}_j||^2_{\boldsymbol{\Sigma}_j} \tag{12}$$

In other words, this represents that the most likely estimate (MLE) of T is the one that minimizes the weighted squared Mahalanobis distance between points and clusters, where the weights are determined by calculating the expectation of the correspondence w.r.t. the current guess $\hat{T}$.

Note that due to the double sum, this equation has NJ terms. However, we can reduce this problem to an optimization procedure of size J by deriving an explicit relation between $M_{\boldsymbol{\Theta}}$ and M_T.

To see how we can do this, we first make the simplification that $\Sigma_j = \mathrm{I}, \forall j \in \{1...J\}$. Now the Mahalanobis distance reduces to the L_2 norm,

$$\hat{T} = \operatorname*{argmin}_T \sum_{ij} \gamma_{ij} ||\tilde{\mathbf{z}}_i - \boldsymbol{\mu}_j||^2 \tag{13}$$

Note that in the $M_{\boldsymbol{\Theta}}$ step, the solution for the mixing parameters is $\hat{\pi}_j = \frac{\sum_i \gamma_{ij}}{N}$, which we can rewrite as $\sum_i \gamma_{ij} = N\hat{\pi}_j$. Similarly, in the $M_{\boldsymbol{\Theta}}$ step, the optimal mean is $\hat{\boldsymbol{\mu}}_j = \frac{\sum_i \gamma_{ij} \mathbf{z}_i}{\sum_i \gamma_{ij}}$, which we can also rewrite as $\sum_i \gamma_{ij} \mathbf{z}_i = (\sum_i \gamma_{ij}) \hat{\boldsymbol{\mu}}_j = N\hat{\pi}_j \hat{\boldsymbol{\mu}}_j$. Finally, let $\tilde{\boldsymbol{\mu}}_j \stackrel{\text{def}}{=} T(\hat{\boldsymbol{\mu}}_j)$. Given these equations, we can reduce the original quantity to be minimized by first expanding it and then completing the square for only those parts that are vary with T.

We first expand,

$$\sum_i \sum_j \gamma_{ij} ||\tilde{\mathbf{z}}_i - \boldsymbol{\mu}_j||^2 \tag{14}$$
$$= \sum_j \{ \sum_i \gamma_{ij} \tilde{\mathbf{z}}_i^T \tilde{\mathbf{z}}_i - 2\boldsymbol{\mu}_j^T \sum_i \gamma_{ij} \tilde{\mathbf{z}}_i + \boldsymbol{\mu}_j^T \boldsymbol{\mu}_j \sum_i \gamma_{ij} \}$$
$$= \sum_j \{ \sum_i \gamma_{ij} \tilde{\mathbf{z}}_i^T \tilde{\mathbf{z}}_i - 2N\hat{\pi}_j \boldsymbol{\mu}_j^T \tilde{\boldsymbol{\mu}}_j + N\hat{\pi}_j \boldsymbol{\mu}_j^T \boldsymbol{\mu}_j \}$$

We can further break down the first term in order to complete the square,

$$\sum_i \gamma_{ij} \tilde{\mathbf{z}}_i^T \tilde{\mathbf{z}}_i = \sum_i \gamma_{ij} ||\tilde{\mathbf{z}}_i - \tilde{\boldsymbol{\mu}}_j||^2 - N\hat{\pi}_j \tilde{\boldsymbol{\mu}}_j^T \tilde{\boldsymbol{\mu}}_j \tag{15}$$

We then place this result back into Eq. 14:

$$\sum_j \{ \sum_i \gamma_{ij} (\tilde{\mathbf{z}}_i - \tilde{\boldsymbol{\mu}}_j)^T (\tilde{\mathbf{z}}_i - \tilde{\boldsymbol{\mu}}_j) +$$
$$N\hat{\pi}_j \tilde{\boldsymbol{\mu}}_j^T \tilde{\boldsymbol{\mu}}_j - 2N\hat{\pi}_j \boldsymbol{\mu}_j^T \tilde{\boldsymbol{\mu}}_j + N\hat{\pi}_j \boldsymbol{\mu}_j^T \boldsymbol{\mu}_j \}$$
$$= N \sum_j \{ \frac{1}{N} \sum_i \gamma_{ij} (\tilde{\mathbf{z}}_i - \tilde{\boldsymbol{\mu}}_j)^T (\tilde{\mathbf{z}}_i - \tilde{\boldsymbol{\mu}}_j) +$$
$$\hat{\pi}_j ||\tilde{\boldsymbol{\mu}}_j - \boldsymbol{\mu}_j||^2 \} \tag{16}$$

The first term inside the summation over j represents the weighted average of the squared distance of each point to its (weighted) centroid, and is thus invariant under rigid transformations so it can be dropped. Dropping this term then transforms the problem into a weighted optimization criterion of size J.

$$\hat{T} = \underset{T}{\operatorname{argmin}} \sum_j \hat{\pi}_j ||\tilde{\boldsymbol{\mu}}_j - \boldsymbol{\mu}_j||^2 \tag{17}$$

In contrast with Eq. 13, we now have J virtualized point correspondences between the decoupled model means and the maximum likelihood means of the new points with respect to the model. Additionally, each pair is weighted by its expected relative contribution among all J mixtures. Finally, it should be stressed that the calculations of $\hat{\pi}_j$ and $\hat{\boldsymbol{\mu}}_j$ are *exactly* the same as in $M_{\boldsymbol{\Theta}}$, and this property allows us utilize the same optimized parallel computation for registration as for mixture decoupling.

Other GMM/EM methods have used similar transformations for reducing the problem size to a single sum over points to virtualized points [11, 12, 8]. These constructions can be viewed as transformations from a one-to-one point-based criterion into a many-to-one or point-to-model criterion. In contrast, our reduction can be viewed as a *many-to-many* or *model-to-model* optimization criterion.

3.4. Closed Form Approximations

If we restrict T to include only the set of $6DOF$ rigid transformations, parameterized by the set of all $R \in SO(3)$ and $t \in \mathbb{R}^3$, such that $\mathcal{Z}_1 = R\mathcal{Z}_2 + \mathbf{t}$, then the minimization of Eq. 17 can be solved by using a weighted version of Horn's method [13]:

$$a_j = \boldsymbol{\mu}_j - \frac{\sum_j \hat{\pi}_j \boldsymbol{\mu}_j}{\sum_j \hat{\pi}_j} \quad \text{, and} \quad b_j = \hat{\boldsymbol{\mu}}_j - \frac{\sum_j \hat{\pi}_j \hat{\boldsymbol{\mu}}_j}{\sum_j \hat{\pi}_j} \tag{18}$$

and

$$K = \sum_j \hat{\pi}_j a_j b_j^T \tag{19}$$

where K is a 3x3 matrix containing all the elements required to construct a special 4x4 matrix, from which the SVD solution gives the optimal rotation as a unit quaternion. See [13] for details.

The optimal translation is then,

$$\hat{t} = \sum_j \hat{\pi}_j (a_j - R b_j) \tag{20}$$

However, in the preceding derivation we omitted the covariances by setting $\Sigma_j = \mathrm{I}, \forall j \in \{1...J\}$. If we want to retain a closed form solution in M_T but do not want to force our GMM in E and $M_{\boldsymbol{\Theta}}$ to have isotropic covariances, one simple approximation strategy is to use the "closest" isotropic *inverse* covariance in a Frobenius sense for the M_T Step. Note that we are still letting the covariance be fully anisotropic for both the E Step and the $M_{\boldsymbol{\Theta}}$ Step. We can solve this trivial approximation for M_T by inspection,

$$\hat{\sigma}_j = \underset{\sigma_j}{\operatorname{argmin}} ||\boldsymbol{\Sigma}_j^{-1} - \sigma_j^2 I||_F = \sqrt{\frac{\mathrm{Tr}(\boldsymbol{\Sigma}_j^{-1})}{d}} \tag{21}$$

Thus, the new M_T criterion is,

$$\sum_j \hat{\pi}_j \hat{\sigma}_j^2 ||\boldsymbol{\mu}_j - \tilde{\boldsymbol{\mu}}_j||^2 \tag{22}$$

We denote the collection of $\hat{\sigma}_j$ as *shape weights*. To differentiate the shape weighting from isotropic weighting, we define M_T*-points* and M_T*-shape*. M_T*-points* is the M step that does not have shape weights, and the M_T*-shape* is the one with shape weights. Note that the shape weights correspond to the average inverse eigenvalue for each covariance, thus scaling point distances inversely by the average "spread" of the cluster covariance. This formulation marks a significant departure to previous works that use just one sigma for all J clusters. We use full anisotropy in both the E steps for segmentation and registration, as well as for the $M_{\boldsymbol{\Theta}}$ step. The only step that does not use full anisotropy is the M_T step, where it is approximated.

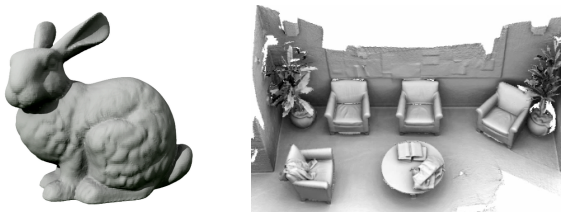

Figure 3: **Datasets** *Left:* Bunny [28], *Right:* Lounge [29]

Method	Recall (≤ 0.01)	Recall (≤ 0.025)	Avg. Time (s)	Std. Time (s)
ICP	0.00	0.00	0.196	0.050
SoftAssign	0.00	0.00	1.832	0.203
GMMReg	0.36	0.91	1.819	1.020
EM-ICP (CPU)	0.14	0.72	1.953	0.353
EM-ICP (GPU)	0.14	0.72	1.136	0.912
M_T-points	**0.53**	**0.92**	**0.107**	0.034
M_T-shape	**0.61**	**0.92**	**0.121**	0.037

Table 2: **Accuracy of random rigid transformations**

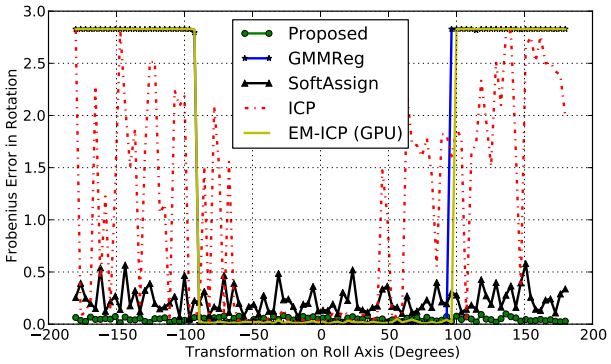

Figure 4: **Robustness to Large Transformations** Pitch axis transformations from -180 to 180 degrees were performed over randomly sampled Stanford Bunnies (N=2000). ICP has the smallest convergence radius, handling transformations of -45 to 45 degrees. EM-ICP and GMMReg approximately double the convergence region. Due to sampling nonuniformity, SoftAssign often produced poor registration accuracy though its worst-case solution was the best of all the algorithms.

4. Results

For our experimental results, we use both synthetic data (Stanford Bunny [28]) and real-world data from the Stanford Scene Datasets [29] (Fig. 3). The bunny contains around 40k points, while the lounge data is taken from a Kinect, and thus consists of 640x480 depth frames taken at 30 Hz. Our hardware testbed is a desktop computer with an i5-3500 CPU and GTX660 GPU.

We compare our method versus other popular publicly available open-source methods. For fairness, we use other parallelized registration methods when available, including optimized multicore implementations of ICP and EM-ICP (using OpenMP), and GPU-accelerated versions of Softassign and EM-ICP [26]. We also test using GMMReg [15], a popular open source package.

4.1. Robustness to Large/Random Rigid Transformations and Noise

To obtain a comparative measure of the convergence properties between our proposed method and the state of the art, we ran each registration algorithm 100 times while varying the amount of rotation (overlap). Using two differently subsampled cross-sections of the Stanford bunny (N=2000), we applied pitch axis transformations from -180 to 180 degrees. Additionally, we added noise in the form of 500 outliers sampled uniformly inside a region twice the extent of the original data. Fig. 4 shows the results. We used the Frobenius norm error between the recovered and real rotation matrices as our measure of rotational error. Both SoftAssign and our method were the only two algorithms that did not diverge over the entire range of pitch transformations, though SoftAssign's error is an order of magnitude higher than our method. As expected, ICP produced the worst convergence, recovering angles of < 45 degrees only. EM-ICP and GMMReg produced high quality results when the angle was less than 90 degrees (about double the range of ICP), but diverged to a local minimum for larger transformations.

To obtain a more general result of robustness to transformations on all axes, we randomly sampled 100 6DOF transformations and applied them to the Stanford Bunny. To sample from the space of rotations in $SO(3)$ uniformly and without bias, we used the method outlined in [16]. The random rotations were bounded by 90 degrees in absolute sum over each axis angle, and the random translations were bounded by the extent of the dataset on each axis. As before, the bunny for both model and scene were randomly subsampled without replacement down to 2000 points and corrupted with 5% noise. In order to derive a simple accuracy measure, we looked at the percentage of solutions (% recall) for which the Frobenius error between the real and calculated rotation was acceptably small. For our method, we set $J = 16$. The results are summarized in Table 2, along with average execution times.

Due to the random subsampling and noise, ICP and SoftAssign could not obtain accurate solutions over the random transformations tested. The GPU optimized EM-ICP was the fastest previous method tested, though ours is roughly an order of magnitude faster. GMMReg produced good results at an error threshold of 0.025, with a recall accuracy of 91%, but generated about a third of the solutions with error under 0.01. In contrast, our methods faired better, recover-

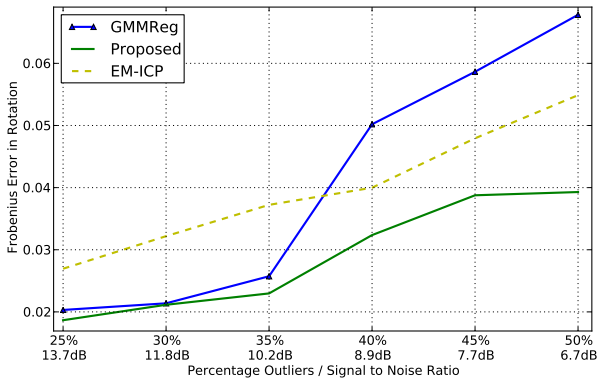

Figure 5: **Robustness to Noise** We injected both outliers and point-level Gaussian noise into two subsampled Bunny point clouds (N=2000) separated by 15 degrees on all roll, pitch, and yaw axes. Each iteration included both more outliers (shown as a percentage on the x-axis) and more point noise (shown as a Signal-to-Noise dB value). We did not include ICP or SoftAssign since any amount of noise and outliers tested corrupted the result significantly.

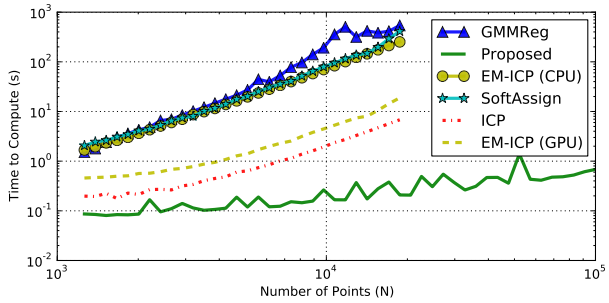

Figure 6: **Computation Times** Scaling the number of points subsampled from each point cloud for registration. Note that the scale is logarithmic on both x and y axes.

ing over half of the transformations at the 0.01 error level, for both point-based and shape-based approximations. The M_T-shape method is about 10% slower in our current GPU implementation than M_T-points due to the need to calculate the shape parameters.

Fig. 5 shows how the proposed approach can handle the point cloud data with additional noises. Notice that our approach outperforms the other methods that can handles noisy inputs.

4.2. Scalability

Many modern sensors generate many thousands to millions of 3D points per second. In general, the registration methods reviewed in this paper greatly benefit from the addition of extra data in terms of accuracy and robustness. Unfortunately, the construction of point-based GMMs provide scaling challenges. When scaling to large numbers of points, even methods that have been optimized to be parallel [26] fail to approach real-time speeds for large point clouds. See Fig. 6 for details. We applied the same random transformations as in Section 4.1, but increased the amount of points sampled from the Stanford Bunny on every iteration. Due to our process of mixture decoupling inside a unified parallelized EM framework, Fig. 6 demonstrates orders of magnitude improvement on large point clouds.

4.3. Real-World Data

To test our algorithm over real-world data, we chose to use the Lounge scene from the Stanford Scene dataset (Fig 3). For ground truth, we used the authors' pose calculations. Below is a table summarizing the frame-to-frame rotation error:

Method	Avg. Err	Std Dev
EM-ICP	0.890	1.109
SoftAssign	0.413	0.753
ICP	0.032	0.110
GMMReg	0.022	0.025
M_T-shape (N=2k)	**0.017**	**0.018**
M_T-shape (N=10k)	**0.010**	**0.005**

We subsampled down to 2000 points for each frame to produce tractable running times. ICP faired somewhat better on this real data since the rigid transformations remain small due to the fast framerate of the Kinect and the small amount of interframe motion. Suprisingly, EM-ICP often diverged, which we attribute to a failure of the multilevel registration process. We ran two versions of our algorithm, one using the same number of points as the others (2k), and the other using a much larger subsampling (10k). As shown in Section 4.2, the parallel nature of our algorithm along with the process of mixture decoupling allows us to incorporate much more data while still keeping running times better than the state of the art. Thus, the 10k subsampled PCD produces a better model and drives down the overall average frame to frame registration error significantly.

5. Conclusion

In this paper, we introduced a novel 3D point cloud registration method that utilizes a preprocessing step to first optimize over the set of models for the purpose of aiding registration, a technique we refer to as mixture decoupling. We first represent the input 3D PCD with a GMM, and decouple the optimization of the model parameters and registration (pose transformation) parameters. This dual-step optimization provides more efficient and accurate registration performance even over large rigid transformations without any good initialization, as well as in conditions of high noise. We also show that the proposed approach can provide a more efficient and scalable solution compared to other state-of-the-art registration algorithms.

References

[1] P. Besl and H. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 1, 2, 4

[2] P. Biber, S. Fleck, W. Strasser, C. Rasmussen, H. Bülthoff, B. Schölkopf, and M. Giese. A probabilistic framework for robust and accurate matching of point clouds. In *Pattern Recognition*, volume 3175, pages 480–487. 2004. 3

[3] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *IEEE Conf. on Intelligent Robots and Systems*, volume 3, pages 2743–2748, 2003. 3

[4] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145 – 155, 1992. Range Image Understanding. 1, 2

[5] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 190–197, 2000. 1, 3, 4

[6] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society.*, pages 1–38, 1977. 1, 4

[7] B. Eckart and A. Kelly. REM-seg: A robust EM algorithm for parallel segmentation and registration of point clouds. In *IEEE Conf. on Intelligent Robots and Systems*, pages 4355–4362, 2013. 3, 4, 5

[8] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis. A generative model for the joint registration of multiple point sets. In *ECCV 2014*, pages 109–122. 2014. 1, 3, 4, 6

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

[10] S. Gold, A. Rangarajan, C. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2d and 3d point matching:: pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998. 1, 3, 4

[11] S. Granger and X. Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. *ECCV 2002*, pages 69–73, 2002. 1, 2, 3, 4, 6

[12] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang. Rigid and articulated point registration with expectation conditional maximization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(3):587–602, 2011. 1, 3, 4, 6

[13] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987. 6

[14] B. Jian and B. C. Vemuri. A robust algorithm for point set registration using mixture of Gaussians. In *IEEE Intern. Conf. on Computer Vision*, pages 1246–1251, 2005. 3, 4

[15] B. Jian and B. C. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1633–1645, 2011. 3, 7

[16] J. J. Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *IEEE Intern. Conf. on Robotics and Automation*, volume 4, pages 3993–3998, 2004. 7

[17] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007. 4

[18] X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993. 3

[19] N. Mitra, N. Gelfand, H. Pottmann, and L. Guibas. Registration of point cloud data from a geometric optimization perspective. In *Eurographics/ACM Symposium on Geometry Processing*, pages 22–31, 2004. 3

[20] 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

[21] A. Myronenko, X. Song, and M. A. Carreira-Perpinán. Non-rigid point set registration: Coherent point drift. In *Advances in Neural Information Processing Systems*, pages 1009–1016, 2006. 3, 4

[22] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001. 1, 2

[23] A. Segal, D. Haehnel, and S. Thrun. Generalized ICP. *Robotics: Science and Systems*, 2(4), 2009. 1, 3, 4

[24] T. Stoyanov, M. Magnusson, and A. J. Lilienthal. Point set registration through minimization of the L2 distance between 3D-NDT models. In *IEEE International Conference on Robotics and Automation*, pages 5196–5201, 2012. 3, 4

[25] T. D. Stoyanov, M. Magnusson, H. Andreasson, and A. Lilienthal. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. *International Journal of Robotics Research*, 2012. 3

[26] T. Tamaki, M. Abe, B. Raytchev, and K. Kaneda. Softassign and EM-ICP on GPU. In *IEEE International Conference on Networking and Computing*, pages 179–183, 2010. 7, 8

[27] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. *ECCV 2004*, pages 558–569, 2004. 3, 4

[28] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH*, pages 311–318, 1994. 7

[29] Q.-Y. Zhou and V. Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics*, 32(4):112, 2013. 7