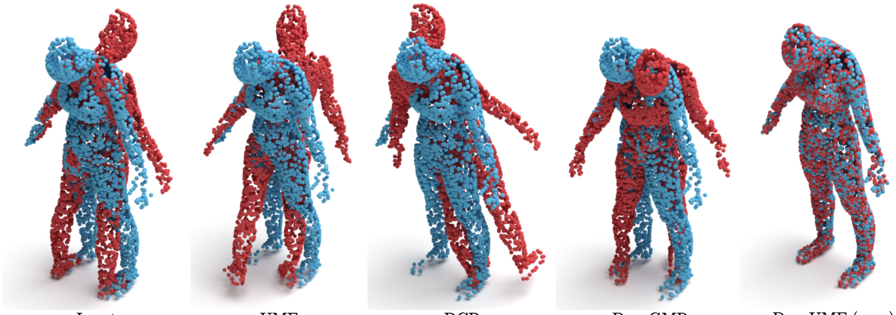
DeepUME: Learning the Universal Manifold Embedding for Robust Point Cloud Registration

Natalie Lang
langn@post.bgu.ac.il
Joseph M.Francos
francos@ee.bgu.ac.il

Ben-Gurion University
Beer-Sheva, Israel

Abstract

Registration of point clouds related by rigid transformations is one of the fundamental problems in computer vision. However, a solution to the practical scenario of aligning differently sampled noisy observations of the point clouds is still lacking. We approach registration in this scenario with a fusion of the Universal Manifold Embedding (UME) method and an unsupervised deep neural network. In order to overcome a major obstacle in the learning process under full rotation range, we employ an SO(3)-invariant coordinate system to learn SO(3)-invariant features, later to be utilized by the closed-form geometric UME method for transformation estimation. We evaluate the performance of the proposed method using the standard RMSE metric as well as using two alternative metrics, designed to overcome the ambiguity problem emerging in ModelNet40 dataset, when noisy scenarios are considered. Finally, we show that our hybrid method outperforms state-of-the-art registration methods in various scenarios, and generalizes well to unseen datasets.



Input UME DCP DeepGMR DeepUME (ours)

Figure 1: Registration results on an unseen dataset, where the observations are subject to a large relative rotation and sampling noise (zero-intersection model). While UME [11] and DCP [40] fail to align the objects, and DeepGMR [48] results with a substantial registration error, the proposed method successfully aligns the shapes.

1 Introduction

The massive development of 3D range sensors [9, 35] led to an intense interest in 3D data analysis. As 3D data is commonly acquired in the form of a point cloud, many related applications have been studied in recent years for that data form. In wide range of applications, specifically in medical imaging [18], autonomous driving [7] and robotics [10], the alignment of 3D objects into a coherent world model is a crucial problem. Point cloud rigid alignment is a deep-rooted problem in computer vision and graphics, and various methods for point cloud registration have been suggested [28].

In general, the point clouds to be registered are sampled from a physical object. When two point clouds are sampled at two different poses of an object, it is unlikely that the same set of object points is sampled in both. The difference between the sampling patterns of the object may result in model mismatch in performing the registration, and we therefore refer to it as sampling noise.

Registration of point clouds in the presence of noise has been extensively studied, both by classical methods [5, 32, 45, 50] and by learning based methods [2, 15, 40, 48]. In most of these works, the noise is modeled as an Additive White Gaussian Noise (AWGN) on the coordinates. However, this type of model is inadequate for modeling sampling noise. As we show in our experiments, sampling noise affects the registration error differently than additive noise on the coordinates of the point cloud and introduces large registration errors.

In this work, we address registration of 3D point clouds in the presence of sampling noise and an additive coordinate noise. Our strategy is to combine the closed-form Universal Manifold Embedding (UME) registration method [11], and a learning based framework. The UME nonlinearly maps functions related by geometric transformations of coordinates (rigid, in our case) to matrices that are linearly related by the transformation parameters. In the UME framework, the embedding of the orbit of possible observations on the object to the space of matrices is based on constructing an operator that evaluates a sequence of low-order geometric moments of some function defined on the point clouds to be registered. This representation is therefore more resilient to noise than local operators, as under reasonable noise, the geometric structure of the point cloud is preserved. Since the UME is an operator defined on functions of the coordinates, in order to enable registration, these functions need to be invariant to the transformation. We generate such invariant features using an unsupervised deep neural network architecture, which is based on DCP [40]. The framework is explained in details in Section 4. We train our framework on ModelNet40 [43] dataset, and test on both seen datasets (ModelNet40), and two unseen dataset (FAUST [6] and Stanford 3D Scanning Repository [36]).

Our main contributions are as follows:

- We address the highly practical, yet less studied problem of point cloud registration in the presence of sampling noise. We point out an ambiguity problem in ModelNet40 dataset that emerges in this scenario and propose alternative error measures that eliminate this ambiguity.
- We integrate the UME registration methodology for the first time into a DNN framework. With the resulting hybrid framework, we show that a successful registration is achievable for the full range of rotation angles and subject to various types of noise, outperforming the compared methods in all evaluated scenarios and metrics.

2 Related Work

There are many approaches to 3D point cloud registration. One of the commonly practiced approaches is to extract and match spatially local features *e.g.*, [16, 20, 33, 44, 46, 47]. Many of the existing methods are adaptations to 2D of image processing solutions, such as variants of 3D-SIFT [23] and the Harris keypoint detector [34]. In 3-D, with the absence of a regular sampling grid, artifacts, and sampling noise, keypoint matching is prone to high outlier rates and localization errors. Hence, the alignment estimated by keypoint matching usually employs outlier rejection methods such as RANSAC [13] and is followed by a refinement stage using local optimization algorithms [4, 24, 29, 49].

Numerous works have been proposed for handling outliers and noise [8], formulating robust minimizers [14], or proposing more suitable distance metrics. Refinement algorithms employ numerical optimization to iteratively minimize an objective function measuring the distance between points in the observation and assumed correspondence points in the reference model [4, 49], or between points in the observation and the surface of the model [4, 24, 29]. The Iterative Closest Point algorithm (ICP) [4, 49] is the standard algorithm in this category. It constructs point correspondences based on spatial proximity followed by a transformation estimation step. Over the years, many variants of the ICP algorithm have been proposed in attempt to improve the convergence rate, robustness, and accuracy of the algorithm.

Registration methods are not restricted only to methods based on the extraction and matching of keypoints. In [12] an initial alignment is found by clustering the orientations of local point cloud descriptors followed by estimating the relative rotation between clusters. In [1] and [25], the algorithm searches congruent sets of four co-planar points between point clouds to create point correspondences.

In this paper we adopt the UME framework [11, 17] designed for registering two functions $f,g:\mathbb{R}^n\to\mathbb{R}$, with compact supports related by a geometric transformation (rigid, affine) parameterized by $\mathbf{A}$. Zero and first order moments (integrals) are evaluated in constructing the $(n+1)\times D$ UME matrix (where $D>n+1$). The UME matrices of f and g satisfy the relation: $\mathrm{UME}_f=\mathbf{A}\cdot\mathrm{UME}_g$.

Learned Registration Pioneered by PointNet [30] and subsequently by DGCNN [42], these methods suggest learning of task-specific point cloud representations. These were then leveraged for robust point cloud registration by [31, 37, 40, 41]. PointNetLK [2] minimizes learned feature distance by a differentiable Lucas-Kanade algorithm [22]. DCP [40] addressed feature matching via an attention-based module, followed by a differentiable SVD for point-to-point registration. The recently proposed RGM [15] transforms point clouds into graphs to perform deep graph matching, extracting soft deep features correspondence matrix. In section 5 we show that registration performance using these architectures deteriorates when the objects are related by large rotations. DeepGMR [48] addresses this problem by extracting pose-invariant correspondences between raw point clouds and Gaussian mixture model (GMM) parameters, and recovers the transformation from the matched mixtures. However, its performance deteriorates in the presence of sampling noise.

3 Problem Definition

A point cloud $\mathcal{P}$ is a finite set of points in $\mathbb{R}^3$. In many applications these points are samples from a physical object, $\mathcal{O}\subseteq\mathbb{R}^3$ (we may think of it as a surface or a manifold). Viewing point clouds as sets of samples, the registration problem may be formulated as follows: Let $\mathcal{O}\subseteq\mathbb{R}^3$ be a physical object and $T(\mathbf{x})=\mathbf{R}\mathbf{x}+\mathbf{t}$ a rigid map ($\mathbf{R}\in\mathrm{SO}(3)$ is a rotation matrix and $\mathbf{t}\in\mathbb{R}^3$ is translation vector). We consider the transformed object $T(\mathcal{O}):=\{T(\mathbf{x})\ :\ \mathbf{x}\in\mathcal{O}\}$. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two point clouds sampled from the object $\mathcal{O}$ and the transformed object $T(\mathcal{O})$, respectively. In the registration problem, the goal is to estimate the transformation parameters $\mathbf{R}$ and $\mathbf{t}$ given only $\mathcal{P}_1$ and $\mathcal{P}_2$.

Since point clouds are generated by a sampling procedure, the effects of sampling must

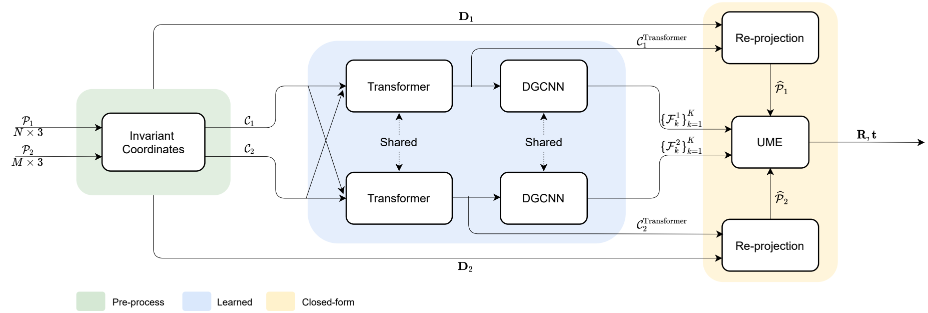


Figure 2: DeepUME network architecture.

be addressed, when solving the registration problem. Ideally, the relation between the two sampled point clouds $\mathcal{P}_1$ and $\mathcal{P}_2$ (sampled from $\mathcal{O}$ and $T(\mathcal{O})$) satisfies the relation $\mathcal{P}_2 = T(\mathcal{P}_1)$. Unfortunately, when point clouds are sampled at two different poses of some object, it is unlikely that the same set of object points is sampled, in both. In fact, if we assume a uniformly distributed sampling pattern on a continuous surface, it may be easily proved that the probability to have such a relation is null. The sampling differences are reflected as a registration error, in a different manner form the AWGN on the coordinates.

We consider the registration problem in the following scenarios:

- Full intersection (Vanilla model) - where $\mathcal{P}_2 = T(\mathcal{P}_1)$.
- Sampling noise - Two cases are considered: Partial intersection, where $\mathcal{P}_2$ and $T(\mathcal{P}_1)$ may intersect, but are not identical; Zero intersection, where $\mathcal{P}_2$ and $T(\mathcal{P}_1)$ have no samples in common.
- Gaussian noise - $\mathcal{P}_2 = T(\mathcal{P}_1) + \mathcal{N}$, where $\mathcal{P}_2$ is a result of a rigid transformation of $\mathcal{P}_1$ with its coordinates perturbed by AWGN.

4 DeepUME

Following the strategy of integrating the UME registration methodology into a deep neural network, we both adjust the UME method [11] adapting it to a DNN framework and design our architecture to optimize the UME performance.

In the UME registration framework the input is composed of two point clouds satisfying the relation $\mathcal{P}_2 = \mathbf{R} \cdot \mathcal{P}_1 + \mathbf{t}$, and an invariant feature (that is, a function $\mathcal{F}$ defined on a point cloud $\mathcal{P}$ such that $f(\mathbf{p}) = f(\mathbf{R} \cdot \mathbf{p} + \mathbf{t})$ for any $\mathbf{p} \in \mathcal{P}$). Applying the UME operator on each of the point clouds, two matrices $\mathbf{M}_{\mathcal{P}_1}$ and $\mathbf{M}_{\mathcal{P}_2}$ are obtained, such that $\mathbf{M}_{\mathcal{P}_2} = \mathbf{R} \cdot \mathbf{M}_{\mathcal{P}_1}$. The geometric nature of the UME motivates as to use it as a basis for our framework. While many registration methods find corresponding points in the reference and the transformed point clouds in order to solve the registration problem, in the UME methodology a new set of “corresponding points" is constructed by evaluating low order geometric moments of the invariant feature. This is a significant advantage in noisy scenarios, where point correspondences between the reference and transformed point clouds, may not exist at all. The use of moments allows us to exploit the geometric structure of the objects to be registered, which is invariant under sampling (as long as the sampling is reliable) resulting in an improved immunity to sampling noise.

The goal of the deep neural network in our framework is to construct multiple high-quality invariant features, in order to maximize the performance of the UME in various noisy scenarios. We adapt DCP architecture [40], which has been proved to be very efficient in creating high-dimensional embedding for point clouds. Conceptually, we may divide our framework into three main parts, each responsible of a different aspect of the registration process. The first part is a pre-process designed to overcome DCP limitation in preforming the learning process over the entire rotation range. The second part is an unsupervised deep neural network responsible for learning features that are subsequently employed for parameter estimation. The final part consists of the UME method for parameter extraction. The framework is illustrated in Figure 2, and explained in details in the following.

UME registration: Following the principles of the UME framework, we derive a new discrete closed-form implementation of the UME for the registration of point clouds undergoing rigid transformations. More specifically, let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two point clouds related by a rigid transformation T. Then, an invariant feature (function) $\mathcal{F}$ on $\mathcal{P}_1$ and $\mathcal{P}_2$ is a function that assigns any point $\mathbf{p} \in \mathcal{P}_1$ and the transformed point $T(\mathbf{p}) \in \mathcal{P}_2$ with the same value. A simple example for such an invariant feature is the one that assigns each point with its distance from the point cloud center of mass.

Since for finite support objects, it is straightforward to reduce the problem of computing the rigid transformation $T(\mathbf{p}) = R\mathbf{p} + \mathbf{t}$ to a rotation-only problem, *i.e.*, $\mathbf{t} = 0$, we show (supplementary material) that the moment integral calculations involved in evaluating the UME operator, may be replaced by computing moments of the invariant functions using summations to yield

$$\mathbf{M}_{\mathcal{P}_2}(\mathcal{F}) = \mathbf{R} \cdot \mathbf{M}_{\mathcal{P}_2}(\mathcal{F}), \quad \text{where } \mathbf{M}_{\mathcal{P}_i}(\mathcal{F}) = \frac{1}{|\mathcal{P}_i|} \begin{bmatrix} \sum_{\mathbf{p} \in \mathcal{P}_i} p_x \mathcal{F}(\mathbf{p}) \\ \sum_{\mathbf{p} \in \mathcal{P}_i} p_y \mathcal{F}(\mathbf{p}) \\ \sum_{\mathbf{p} \in \mathcal{P}_i} p_z \mathcal{F}(\mathbf{p}) \end{bmatrix}. \tag{1}$$

We call $\mathbf{M}_{\mathcal{P}_i}$ the moment vector of the transformation invariant function $\mathcal{F}$ defined on $\mathcal{P}_i$.

Given two sets of points in $\mathbb{R}^3$, $\{\mathbf{v}_i\}_{i=1}^k$ and $\{\mathbf{u}_i\}_{i=1}^k$ $k \geq 3$, satisfying the relation $\mathbf{v}_i = \mathbf{R} \cdot \mathbf{u}_i$ for all i, we may find $\mathbf{R}$ by a standard procedure proposed by Horn *et al.* [19, 26] . Hence, we conclude from (1) that in the absence of noise, finding a set of invariant functions $\mathcal{F}_1, \ldots, \mathcal{F}_k$, such that $k \geq 3$, yields a closed-form solution to the registration problem.

However in the presence of noise (sampling or additive) (1) no longer holds as $\mathcal{P}_2$ and $\mathbf{R} \cdot \mathcal{P}_1$ are not identical anymore and in fact, with a high probability they do not share any point in common. Therefore, the estimated rotation matrix is noisy.

This is the point where a deep neural network comes into play. The registration error obviously depends on the difference between $\mathbf{M}_{\mathcal{P}_2}$ and $\mathbf{R} \cdot \mathbf{M}_{\mathcal{P}_1}$, and on the function $\mathcal{F}$ being rigid transformation invariant despite the noise. To that extent, we employ a deep neural network in order to *learn* how to construct good transformation-invariant functions. These invariant functions are designed to exploit the geometry of the point cloud so that their invariance to the transformation is minimally affected by the noise, resulting in a smaller registration error.

DGCNN: Towards obtaining noise resilient SO(3)-invariant functions for effectively evaluating the UME moments, we aim at learning features capturing the geometric structure of the point cloud. This structure is determined by the point cloud coordinates (global information) and the neighborhood of each point (local information). For that reason, we adopt the same architecture used by the DCP embedding network - DGCNN [42]. DGCNN is designed to preform a per-point embedding such that information on neighboring points is well incorporated.

A key to DGCNN local information extraction into features is it's input structure - instead of a raw point cloud, the input is the point clouds self-looped k-NN graph $\mathcal{G}$. As such, each point in the point cloud is characterized by the coordinates of points in its neighborhood in addition to its own coordinates. This approach results in a new representation for each point using a $6 \times k$ matrix. Each column of that matrix represents one of the k nearest neighbors and consists of the coordinates of the observed point stacked on top of the coordinated of the neighbor point.

Formally, DGCNN is a neural network comprised of a total of L layers. Let h_θ^l denote the nonlinear function parameterized by a multi-layer perceptron in the l-th layer, and let $\mathbf{p}_i^l$, denote the embedding of the point cloud i-th point. The network forward pass is given by

$$\mathbf{p}_i^l = f\left(\left\{h_\theta^l\left(\mathbf{p}_i^{l-1}, \mathbf{p}_j^{l-1}\right) \ \forall j \in \mathcal{N}_i\right\}\right) \tag{2}$$

where $\mathcal{N}_i$ is the i-th point neighbourhood in $\mathcal{G}$ and f represents channel-wise max aggregation.

DGCNN embeds one point cloud at a time, sharing its weights between the two clouds. Ideally, our embedding network should result in identical feature (function value) for two corresponding points in the reference point cloud and in its transformed version.

SO(3)-invariant coordinate system followed by a Transformer: Since DGCNN weights are shared, when the coordinates of corresponding points between two point clouds are significantly different, the learning process fails. Clearly, this situation occurs when transformations of large magnitude, *e.g.*, large rotations, are considered. See Section 5. Therefore DGCNN in its existing design cannot be employed towards constructing invariant features when rotations by large angles are considered.

We overcome this inherent difficulty by transferring the input point clouds to an alternative coordinates system, which is SO(3) invariant: Given a point cloud $\mathcal{P}$, the cloud center of mass (denoted by $\mathbf{m}_\mathcal{P}$) is subtracted from each point coordinates, to obtain a centered representation $\mathcal{P}'$. We then construct a new coordinate system for the point cloud using PCA. That is, the axes of the new coordinate system are the principle vectors of the point cloud covariance matrix given by

$$\mathbf{H}^{\mathcal{P}'} = \sum_{\mathbf{p} \in \mathcal{P}'} \mathbf{p}\mathbf{p}^T \tag{3}$$

The principle vectors form the axes of the new coordinate system, and the new coordinates of each point are the projection coefficients on these axes. Formally, for a point $\mathbf{p} \in \mathcal{P}'$, the new coordinates of $\mathbf{p}$ are defined to be $\mathbf{c_p} = \mathbf{D}^T\mathbf{p}$ where $\mathbf{D}$ is the matrix whose columns are the principle vectors. The resulting point cloud new coordinates are denoted by $\mathcal{C}$.

It is easy to verify (see supplementary) that the new axes (columns of the PCA matrix) are co-variant under a rigid transformation. Therefore if $\mathcal{P}_2$ is obtained by a rigid transformation of $\mathcal{P}_1$, it holds that $\mathcal{C}_1 = \mathcal{C}_2$. Furthermore, since the change of coordinate system is invertible, the original point cloud can be reconstructed.

In the noisy case, the relation $\mathcal{C}_1 = \mathcal{C}_2$ does not hold anymore. Due to the noise, axes are no longer co-variant and thus the projections are no longer invariant. However, the difference between $\mathcal{C}_1$ and $\mathcal{C}_2$ is sufficiently small so that a successful learning process of invariant features may be applied using DGCNN.

In order to minimize the sampling noise impact on the registration results, we employ a strategy inspired by DCP. In [40] it is concluded that more reliable features for point clouds registration can be learned once their embeddings are processed simultaneously, by a Transformer. We therefore employ the DCP Transformer [39] for learning a resampling strategy of the projected samples in $\mathcal{C}_1$ such that the resampling depends on the sampling of $\mathcal{C}_2$, and vice versa. Denoting the asymmetric function of the Transformer by ϕ, we have

$$\mathcal{C}_1^{\text{Transformer}} = \mathcal{C}_1 + \phi(\mathcal{C}_1, \mathcal{C}_2), \quad \mathcal{C}_2^{\text{Transformer}} = \mathcal{C}_2 + \phi(\mathcal{C}_2, \mathcal{C}_1). \tag{4}$$

The dual resampling process executed by the Transformer has two objectives in the UME framework. The first, as mentioned, is to provide DGCNN a good 'starting point' for constructing better invariant features. The second is related to the actual evaluation of the UME moments, where we use the resampled point clouds produced by the Transformer and re-project them on the corresponding principle axes to obtain

$$\widehat{\mathcal{P}}_1 = \mathbf{D}_1 \cdot \mathcal{C}_1^{\text{Transformer}} + \mathbf{m}_{\mathcal{P}_1}, \quad \widehat{\mathcal{P}}_2 = \mathbf{D}_2 \cdot \mathcal{C}_2^{\text{Transformer}} + \mathbf{m}_{\mathcal{P}_2}. \tag{5}$$

The point clouds, $\widehat{\mathcal{P}}_1$ and $\widehat{\mathcal{P}}_2$, are re-sampled versions of the original points clouds, related by the same rigid transformation that relates $\mathcal{P}_1$ and $\mathcal{P}_2$. We therefore apply the UME registration to the resampled point clouds $\widehat{\mathcal{P}}_1$ and $\widehat{\mathcal{P}}_2$, using the DGCNN generated functions designed to be $\mathrm{SO}(3)$ invariant in the presence of observation sampling noise. As demonstrated by the experimental results, applying UME registration on the re-projected re-sampled point clouds provides improved performance.

Loss: In order to overcome the possible ambiguity problem of symmetric objects, discussed in Section 5, we adopt the Chamfer distance [3] as our loss function. That is, if $\hat{T}$ is the estimated transformation,

$$\mathcal{L}(\hat{T}) = d_C(\hat{T}(\mathcal{P}_1), \mathcal{P}_2). \tag{6}$$

Using this loss, the ambiguous examples such as those in ModelNet40 [43] do not damage the learning process, as even if an ambiguity exists and the registration is successful, the Chamfer distance will be small. The Chamfer loss function has another advantage as no labels are required for the learning process, what makes it unsupervised.

5 Experiments

We conduct experiments on three datasets: ModelNet40, FAUST and Stanford 3D Scanning Repository where the last two are used only for testing. We train our network using ModelNet40, which consists of 12,311 CAD models, in 40 categories where (80%) are used for training and the rest for testing.

Following previous works experimental settings, we uniformly sample 1,024 points from each model's outer surface and further center and rescale the model into the unit sphere. In our training and testing procedure, for each point cloud, we randomly (uniformly) choose a rotation drawn uniformly from the full range of Euler angles and a translation vector in $[-0.5, 0.5]$ in each axes. We apply the rigid transformation obtained from the resulting parameters on $\mathcal{P}_1$, followed by a random shuffling of the points order, and get $\mathcal{P}_2$. In noisy scenarios, a suitable noise is applied to $\mathcal{P}_2$. We train our framework using NVIDIA Quadro RTX 6000 in the scenario of Bernulli noise (see bellow), in the specific case where $p_1 = p_2 = 0.5$, and test all scenarios using the trained configuration.

Each experiment is evaluated using four metrics: The RMSE metric, both for the rotation and the translation as well as the Chamfer distance and the Hausdorff distance [21, 38], proposed next as alternative metrics where ambiguity issues are resolved. We compare our performances with the basic implementation of the UME method (coded by ourselves), ICP (implemented in Intel Open3D [51]), and four learned methods; PointNetLK and DCP (benchmarks point cloud registration networks) as well as the recently proposed DeepGMR, [48] and RGM, [15]. We retrain the baselines, adapting the code released by the authors, and test all compered methods in exactly the same setting.

The ambiguity problem in ModelNet40 An ambiguity problem arises in point cloud registration whenever symmetric objects are considered, as more than a single rigid transformation (depending on the degree of symmetry of the object) can correctly align the two point clouds. In the noise free scenario such an ambiguity is trivially handled since a fixed point constellation undergoes a rigid transformation which in the case of nonuniform sampling guaranties the uniqueness of the solution. However, when observations are noisy, the ambiguity is harder to resolve as the constellation structure breaks. In that scenario, $\mathcal{P}_2$ is a noisy version of $T(\mathcal{P}_1)$, and possibly no rigid transformation can perfectly align the point clouds. In that case, if the point clouds to be aligned represent a symmetric shape, there are multiple transformations that approximately align the two clouds together.

For symmetric shapes, which are very common in ModelNet40 dataset, the rotation angles RMSE metric may assign large errors to successful registrations. To resolve this ambiguity we suggest to replace this metric by the well known Chamfer and Hausdorff distances defined by

$$\begin{aligned} d_{\text{C}}(\mathcal{P}_1,\mathcal{P}_2) &= \frac{1}{|\mathcal{P}_1|}\sum_{\mathbf{p}\in\mathcal{P}_1}\min_{\mathbf{p}'\in\mathcal{P}_2}\|\mathbf{p}-\mathbf{p}'\|_2 + \frac{1}{|\mathcal{P}_2|}\sum_{\mathbf{p}'\in\mathcal{P}_2}\min_{\mathbf{p}\in\mathcal{P}_1}\|\mathbf{p}'-\mathbf{p}\|_2 \\ d_{\text{H}}(\mathcal{P}_1,\mathcal{P}_2) &= \max_{\mathbf{p}\in\mathcal{P}_1}\min_{\mathbf{p}'\in\mathcal{P}_2}\|\mathbf{p}-\mathbf{p}'\|_2 + \max_{\mathbf{p}'\in\mathcal{P}_2}\min_{\mathbf{p}\in\mathcal{P}_1}\|\mathbf{p}'-\mathbf{p}\|_2. \end{aligned} \tag{7}$$

Using ModelNet40, we test performance, in four scenarios: noise free model, sampling noise (Bernoulli noise and zero-intersection noise) and additive white Gaussian coordinate noise. We note that in the presence of sampling noise we indeed observe large errors in RMSE(R) due to the symmetry of objects although the symmetric shapes are well aligned. Moreover, in the unseen datasets, where real world data is used, which is naturally asymmetric, the rotation RMSE is indeed small and indicates successful registration. Therefore we consider the Chamfer and Hausdorff distances to be more reliable metrics for registration in the presence of symmetries. Figure 1 shows registration results of several baseline methods and our proposed method on a representative example from the unseen dataset FAUST (point cloud rendering is made using Mitsuba [27]). All experimental results are summarized in Tables 1 and 2.

ModelNet40: Noise free model We examine the case where no noise is applied to the measurements. In that case, we see that the estimation error is practically null. DeepGMR is shown to be the second best learned method, while all other tested methods yield large errors, meaning that the registration fails when large range of rotation angles is considered.

ModelNet40: Bernoulli noise The Bernoulli noise case is related to the scenario of sampling noise, in which the point clouds contain different number of points. In that case, we choose randomly (uniformly) two numbers p_1 and p_2 in $[0.2, 1]$. Then, each point in $\mathcal{P}_i$ is removed with probability $1 - p_i$, independently from the rest of the points. We perform registration on the resulting point clouds $\mathcal{P}_1^B$ and $\mathcal{P}_2^B$. We note that the number of points in $\mathcal{P}_i^B$ averages to $p_i \cdot 2048$, and is likely to be different in the two clouds. The number of corresponding points between the resulting clouds averages to $p_1 p_2 \cdot 2048$ (as the probability for the two point clouds to share a specific point is $p_1 p_2$). We note that we were not able to evaluate RGM performance in that scenario.

ModelNet40: Zero-intersection noise Zero intersection noise is the extreme (and most realistic) case of sampling noise. In that case, we randomly choose 1024 points from $\mathcal{P}_1$ to

	Noise free				Bernoulli noise				Gaussian noise			
Model	d_{C}	d_{H}	**RMSE**(R)	**RMSE**(t)	d_{C}	d_{H}	**RMSE**(R)	**RMSE**(t)	d_{C}	d_{H}	**RMSE**(R)	**RMSE**(t)
UME [11]	<1e-04	<7e-04	<u>0.193</u>	<u><1e-05</u>	0.0581	0.394	74.164	<u>0.015</u>	0.019	0.151	<u>27.684</u>	<u>0.002</u>
ICP [5]	0.275	1.446	83.039	0.276	0.297	1.480	86.381	0.286	0.266	1.436	83.073	0.277
PointNetLK [2]	0.027	0.142	80.360	1.0105	<u>0.029</u>	<u>0.157</u>	83.280	1.073	0.026	0.153	81.843	1.037
DCP [40]	0.059	0.470	92.285	0.014	0.067	0.483	92.818	0.020	0.055	0.456	90.715	0.014
DeepGMR [48]	<u><7e-06</u>	<u><9e-05</u>	<u>0.193</u>	<5e-05	0.033	0.224	<u>72.447</u>	0.018	<u>0.011</u>	<u>0.085</u>	42.515	0.004
RGM [15]	0.255	1.333	99.937	0.388	N/A	N/A	N/A	N/A	0.254	1.331	100.117	0.389
DeepUME (ours)	**<1e-07**	**<1e-07**	**<3e-04**	**<1e-07**	**0.010**	**0.083**	**40.357**	**0.015**	**0.002**	**0.012**	**2.425**	**0.001**

Table 1: ModelNet40 experimental results. Our method achieves substantial performance gains compared to the competing techniques for all metrics in all the examined scenarios. d_{C} and d_{H} stands for Chamfer and Hausdorff distances respectively. Best results are **bold** and second best are <u>underlined</u>.

	ModelNet40 [43]				FAUST [6]				Stanford 3D Scanning Repository [36]			
Model	d_{C}	d_{H}	**RMSE**(R)	**RMSE**(t)	d_{C}	d_{H}	**RMSE**(R)	**RMSE**(t)	d_{C}	d_{H}	**RMSE**(R)	**RMSE**(t)
UME [11]	0.051	0.373	80.331	<u>0.010</u>	0.007	0.085	35.983	0.044	0.033	0.267	48.716	<u>0.010</u>
ICP [5]	0.276	1.448	82.948	0.277	0.376	1.643	84.544	0.279	0.288	1.337	87.292	0.277
PointNetLK [2]	0.028	0.147	80.858	1.023	0.018	0.170	90.512	1.120	0.040	0.289	84.520	1.147
DCP [40]	0.059	0.475	93.221	0.014	0.046	0.516	94.315	0.137	0.072	0.522	99.328	0.011
DeepGMR [48]	<u>0.026</u>	<u>0.117</u>	**67.282**	<u>0.010</u>	<u>0.003</u>	<u>0.027</u>	<u>27.941</u>	<u>0.020</u>	<u>0.005</u>	<u>0.119</u>	<u>39.402</u>	0.012
RGM [15]	0.254	1.335	100.970	0.388	0.385	1.677	114.496	0.418	0.278	1.257	104.872	0.368
DeepUME (ours)	**0.011**	**0.094**	<u>70.818</u>	**0.009**	**0.002**	**0.024**	**8.630**	**0.019**	**0.002**	**0.110**	**5.625**	**0.010**

Table 2: Zero-intersection noise results on seen (ModelNet40) and unseen datasets (FAUST and Stanford 3D Scanning Repository). Our method outperforms the competing techniques in all scenarios for all metrics, only except RMSE(**R**) for ModelNet40.

be removed. Next, we remove the 1024 points from $\mathcal{P}_2$ that correspond to the points in $\mathcal{P}_1$. Thus, by construction, no point in one cloud is the result of applying a rigid transformation to a point in the other cloud.

ModelNet40: Additive Gaussian Coordinate Noise In these set of experiments each coordinate of every point in $\mathcal{P}_2$ is perturbed by an additive white Gaussian random variable drawn from $\mathcal{N}(0,\sigma)$ where σ is chosen randomly in $[0, 0.04]$. All noise components are independent and no value clipping is performed. The results in Table 1 indicate that since the UME is an integral (summation) operator the registration error of both the UME and the DeepUME is lower than the error of the other tested methods.

Unseen dataset The generalization ability of a model to unseen data is an important aspect for any learning based framework. In order to demonstrate that our framework indeed generalizes well, we test it on two unseen datasets. The first is the FAUST dataset that contains human scans of 10 different subjects in 30 different poses each with about 80,000 points per shape, and the other is the Stanford 3D Scanning Repository. We generate the objects to be registered using a similar methodology to that employed for the ModelNet40 dataset. Our framework achieves accurate registration results in all scenarios checked, and shows superior performance over the compared methods.

6 Conclusions

We derived a novel solution to the highly practical problem of aligning differently sampled and noisy point clouds. The solution integrates the closed form UME registration into a DNN framework. Since the UME is an operator defined on functions of the coordinates, in order to enable registration, these functions need to be invariant to the transformation. We generate such invariant features using an unsupervised deep neural network architecture designed to jointly resample the two point clouds to minimize the effect of the sampling noise. The derived new discrete version of the UME operator and its integration into the DNN framework enable us to achieve accurate registration in various noisy scenarios. In addition, projecting the point clouds on transformation invariant coordinate system, removes a major obstacle in the learning process of DGCNN-based networks when registration under large rotations is considered.

References

[1] D. Aiger, N. J. Mitra, and D. Cohen-Or. 4-points congruent sets for robust surface registration. *ACM Transactions on Graphics*, 27(3):#85, 1–10, 2008.

[2] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019.

[3] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical report, SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER, 1977.

[4] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992. ISSN 0162-8828.

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014.

[7] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.

[8] Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and vision computing*, 23(3):299–309, 2005.

[9] RTH Collis. Lidar. *Applied optics*, 9(8):1782–1788, 1970.

[10] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[11] Amit Efraim and Joseph M Francos. The universal manifold embedding for estimating rigid transformations of point clouds. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5157–5161. IEEE, 2019.

[12] I. H. Ferencz and I. Shimshoni. Registration of 3d point clouds using mean shift clustering on rotations and translations. *2017 Int. Conf. 3D Vis. (3DV)*, pages 374–382, 2017.

[13] Martin A Fischler and Robert 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.

[14] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and vision computing*, 21(13-14):1145–1153, 2003.

[15] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. *Internaltional Conference on Computer Vision and Pattern Recogintion (CVPR)*, 2021.

[16] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, Jan 2016.

[17] Rami R Hagege and Joseph M Francos. Universal manifold embedding for geometrically deformed functions. *IEEE Transactions on Information Theory*, 62(6):3676–3684, 2016.

[18] Derek LG Hill, Philipp G Batchelor, Mark Holden, and David J Hawkes. Medical image registration. *Physics in medicine & biology*, 46(3):R1, 2001.

[19] Berthold KP Horn, Hugh M Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*, 5(7):1127–1135, 1988.

[20] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999. ISSN 0162-8828.

[21] Yongshan Liu, Dehan Kong, Dandan Zhao, Xiang Gong, and Guichun Han. A point cloud registration algorithm based on feature extraction and matching. *Mathematical Problems in Engineering*, 2018, 2018.

[22] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proc. of the Intl. Joint Conference on Artificial Intelligence*. Vancouver, British Columbia, 1981.

[23] Chris Maes, Thomas Fabry, Johannes Keustermans, Dirk Smeets, Paul Suetens, and Dirk Vandermeulen. Feature detection on 3d face surfaces for pose normalisation and recognition. In *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6. IEEE, 2010.

[24] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24(10):803–827, 2007. doi: https://doi.org/10.1002/rob.20204.

[25] Nicolas Mellado, Dror Aiger, and Niloy J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33(5):205–215, 2014. ISSN 1467-8659.

[26] Andriy Myronenko and Xubo Song. On the closed-form solution of the rotation matrix arising in computer vision problems. *arXiv preprint arXiv:0904.1613*, 2009.

[27] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)*, 38 (6):1–17, 2019.

[28] François Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1): 1–104, 2015.

[29] Helmut Pottmann, Stefan Leopoldseder, and Michael Hofer. Registration without icp. *Computer Vision and Image Understanding*, 95(1):54 – 71, 2004. ISSN 1077-3142.

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[31] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[32] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001.

[33] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391, Sep. 2008.

[34] Ivan Sipiran and Benjamin Bustos. Harris 3d: A robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27:963–976, 11 2011.

[35] Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3d with kinect. In *Consumer depth cameras for computer vision*, pages 3–25. Springer, 2013.

[36] Stanford Scanning Repository. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/, 1994.

[37] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2530–2539, 2018.

[38] Dahlia Urbach, Yizhak Ben-Shabat, and Michael Lindenbaum. Dpdist: Comparing point clouds using deep point cloud distance. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, volume 12356 of *Lecture Notes in Computer Science*, pages 545–560. Springer, 2020. doi: 10.1007/978-3-030-58621-8_32. URL https://doi.org/10.1007/978-3-030-58621-8_32.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[40] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019.

[41] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *arXiv preprint arXiv:1910.12240*, 2019.

[42] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[43] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[44] J. Yang, Y. Xiao, and Z. Cao. Toward the repeatability and robustness of the local reference frame for 3d shape matching: An evaluation. *IEEE Transactions on Image Processing*, 27(8):3766–3781, Aug 2018. ISSN 1057-7149.

[45] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde 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, 2015.

[46] Jiaqi Yang, Zhiguo Cao, and Qian Zhang. A fast and robust local descriptor for 3d point cloud registration. *Information Sciences*, 346-347:163 – 179, 2016. ISSN 0020-0255.

[47] Jiaqi Yang, Qian Zhang, Ke Xian, Yang Xiao, and Zhiguo Cao. Rotational contour signatures for both real-valued and binary feature representations of 3d local shape. *Computer Vision and Image Understanding*, 160:133 – 147, 2017. ISSN 1077-3142.

[48] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *European Conference on Computer Vision*, pages 733–750. Springer, 2020.

[49] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994.

[50] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016.

[51] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.