DeepUME: Learning the Universal Manifold Embedding for Robust Point Cloud Registration [1]

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 sparsely and differently sampled observations in the presence of noise is still lacking. We approach registration in this scenario with a fusion of the closed-form Universal Manifold Embedding (UME) method and a deep neural network. The two are combined into a single unified framework, named DeepUME, trained end-to-end and in an unsupervised manner. To successfully provide a global solution in the presence of large transformations, we employ an $SO(3)$-invariant coordinate system to learn both a joint-resampling strategy of the point clouds and $SO(3)$-invariant features. These features are then utilized by the geometric UME method for transformation estimation. The parameters of DeepUME are optimized using a metric designed to overcome an ambiguity problem emerging in the registration of symmetric shapes, when noisy scenarios are considered. We show that our hybrid method outperforms state-of-the-art registration methods in various scenarios, and generalizes well to unseen data sets. Our code is publicly available[2].

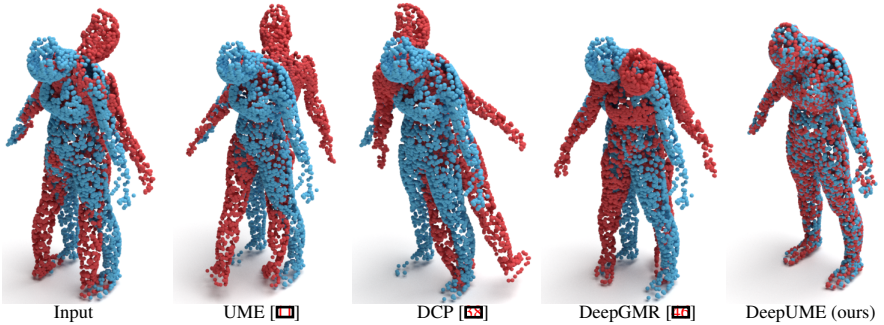


Figure 1: Registration results on an unseen data set, where the observations are subject to a large relative rotation and sampling noise (zero-intersection model). While UME [11] and DCP [33] fail to align the objects, and DeepGMR [40] results with a substantial registration error, the proposed method successfully aligns the shapes.

[1] This research was supported by NSF-BSF Computing and Communication Foundations (CCF) grants, CCF-2016667, and BSF-2016667 and by the Israeli Ministry of Innovation, Science and Technology grant 3-16583.

[2] https://github.com/langnatalie/DeepUME

1 Introduction

The massive development of 3D range sensors [2, 33] 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 [6] 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 [26].

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, and especially when sampling is sparse, 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 when performing registration, and we therefore refer to it as sampling noise. Registration of point clouds in the presence of noise has been extensively studied by both closed-form [2, 30, 43, 48] and learning-based [1, 15, 38, 46] methods. In most of these works, the noise is modeled as an Additive White Gaussian Noise (AWGN) on the coordinates. However, in many registration applications the point clouds are sampled differently and sparsely. In such applications, these sampling effects are dominant and adversely affect registration performance, yet they cannot be modeled by an AWGN.

In this work we address the global registration of 3D under-sampled point clouds, where the point clouds are differently sampled, and the samples are subject to the presence of 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 non-linearly 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 (features) need to be invariant to the transformation. While in the original UME framework, the invariant features are *hand-crafted functions*, in this work we *learn* those from data using an unsupervised deep neural network architecture.

The proposed framework is a cascade of three blocks: The first is a pre-processing step that employs an SO(3)-invariant coordinate system, constructed using PCA, to enable estimation of large transformations. The second block is the neural network, designed to implement joint-resampling and embedding of the raw point clouds. The final block implements the UME. Our trained model is tested on both seen and unseen data sets, to demonstrate generalization capabilities. Inference performance is evaluated for different noise scenarios using metrics that are invariant to the ambiguity arising in symmetric shapes registration when noisy scenarios are considered.

Our main contributions are as follows:

- We integrate, for the first time, the closed-form UME registration methodology and a data-driven approach by both adapting the UME method to the DNN framework and designing the DNN architecture to optimize the UME performance. We address the highly practical yet less studied case of registering point clouds that are sparsely and

randomly sampled in the presence of large transformations (full range of rotations). Our hybrid model is trained end-to-end, labels free and results with substantial performance gains compared to competing state-of-the-art methods.

- To enhance the DeepUME performance in the presence of large deformations and since the point clouds are sparsely and differently sampled, we present a learned joint resampling of both point clouds to be registered, using a novel approach for integrating a PCA module, a Transformer module and a DGCNN module. By mapping the input data to an $SO(3)$-invariant coordinates system, we overcome DGCNN inability to learn invariant features for registration in the case of large rotations. The Transformer is used for implementing a joint-resampling strategy of both point clouds to be registered. However, while learned joint-resampling procedures are usually applied in a *high-dimensional* feature space, in our framework, the joint-resampling is aimed at "equalizing" the differences in the sampling patterns of the observations, and is therefore implemented in the coordinate *low-dimensional* space.

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, 21, 31, 42, 44, 45]. Many of the existing methods are 3D adaptations of 2D image processing solutions, such as variants of 3D-SIFT [24] and the 3D Harris key-point detector [32]. In 3D, with the absence of a regular sampling grid, artifacts, and sampling noise, key-point matching is prone to high outlier rates and localization errors. Hence, global alignment estimated by key-point matching usually employs outlier rejection methods such as RANSAC [13] followed by a refinement stage using local optimization algorithms [3, 25, 27, 47]. DGR [8] follows a similar paradigm, but inlier detection is learnable. Numerous works have been proposed for handling outliers and noise [7], formulating robust minimizers [14], or proposing more suitable distance metrics. The standard algorithm in the category of refinement algorithms, also known as local registration, is the Iterative Closest Point algorithm (ICP) [3, 47]. 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 key-points. In [20], for example, an initial alignment is found by employing a matched filter in the frequency space of local orientation histograms. 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 this work, a global closed-form solution that employs the UME [11] representation of the shapes to be registered, is being integrated. As a result, an efficient and accurate registration scheme is achieved where no initial alignment is required.

Other types of registration methods adopt learning-based techniques. Pioneered by PointNet [28] and subsequently by DGCNN [40], point cloud representations are learned from data in a task-specific manner. These can, in turn, be leveraged for robust point cloud registration *e.g.*, [29, 35, 38, 39]. PointNetLK [1] minimizes learned feature distance by a differentiable Lucas-Kanade algorithm [23]. DCP [38] addressed feature matching via attention-based module and differentiable SVD modules for point-to-point registration. The recently proposed RGM [15] transforms point clouds into graphs and perform deep graph matching for extracting deep features soft correspondence matrix. In section 5 we show that

registration performance using the aforementioned architectures deteriorates when observations are related by large rotations. We relax that limitation by employing an SO-(3)invariant coordinate system, and therefore provide a global registration solution. DeepGMR [46] also addresses this limitation 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 $\mathbb{R}^3$. Usually, 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 objective 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 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 $T(\mathcal{O})$, respectively) satisfies the relation $\mathcal{P}_2 = T(\mathcal{P}_1)$. Unfortunately, when point clouds are sparsely sampled at two different poses of some object, it is unlikely that the same set of object points is sampled, in both: If we assume a uniformly distributed sampling pattern on a continuous surface, it may be easily proved that the probability of having such a relation is null. As we show in our experiments, once sparse and differently-sampled point clouds are considered, the sampling differences result in substantial registration errors, having different characteristics from those of AWGN on the coordinates.

Under-sampled point cloud registration is considered 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. The proposed framework is a cascade of three blocks: The first is a pre-processing step that employs an SO(3)-invariant coordinate system, constructed using PCA, that enables estimation of large transformations. The second block is the neural network, designed to implement joint-resampling and embedding of the raw point clouds aimed at "equalizing" the differences in the sampling patterns of the observations, and consequently boost the performance of the third block, implementing the UME. The framework is illustrated in Figure 2 (see supplementary for detailed illustrations). The details about its building blocks, and in particular the Transformer and DGCNN architectures adaptations into the UME framework are provided in the following.

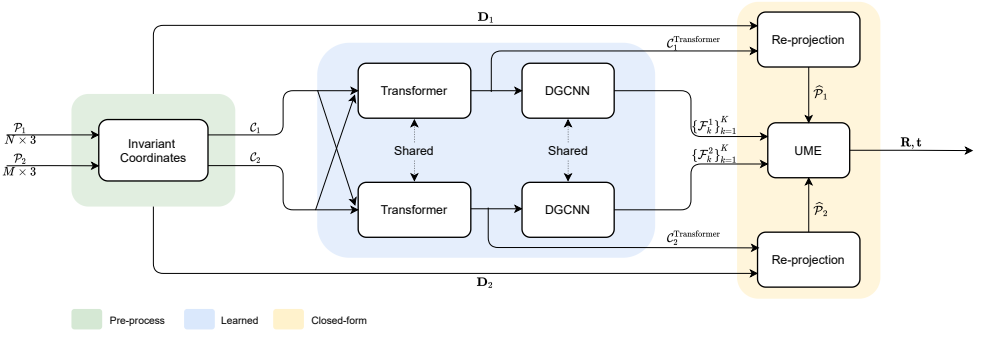


Figure 2: DeepUME network architecture.

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 $\mathcal{F}(\mathbf{p}) = \mathcal{F}(\mathbf{R} \cdot \mathbf{p} + \mathbf{t})$ 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 us 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 adopt the joint usage of DGCNN and Transformer blocks [28], which has been proven to be very efficient in creating high-dimensional embedding for point clouds, and adapt it to the UME framework in order to *learn* multiple high-quality $\mathrm{SO}(3)$ invariant features.

UME registration The UME framework [11, 17] is 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 UME matrix of dimension $(n+1) \times D$ (where $D > n+1$). The UME matrices of f and g satisfy the relation: $\mathrm{UME}_f = \mathbf{A} \cdot \mathrm{UME}_g$. Following the principles of the UME, 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 $\mathbf{T}$. 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 $\mathbf{T}(\mathbf{p}) \in \mathcal{P}_2$ 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 $\mathbf{T}(\mathbf{p}) = \mathbf{R}\mathbf{p} + \mathbf{t}$ to a rotation-only problem, *i.e.*, $\mathbf{t} = 0$, we next show that the moment integral calculations involved in evaluating the UME operator, may be replaced by computing moments of the invariant functions using summations:

Theorem 4.1 *Let* $\mathbf{R}$ *be a rotation matrix and* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *be two point clouds satisfying the relation* $\mathcal{P}_2 = \mathbf{R} \cdot \mathcal{P}_1$. *Let* $\mathcal{F}$ *be an* $SO(3)$ *invariant function on* $\mathcal{P}_1$ *and* $\mathcal{P}_2$. *Then,*

$$\mathbf{M}_{\mathcal{P}_2}(\mathcal{F}) = \mathbf{R}\cdot\mathbf{M}_{\mathcal{P}_1}(\mathcal{F}), \quad \textit{where } \mathbf{M}_{\mathcal{P}_i}(\mathcal{F}) = \frac{1}{|\mathcal{P}_i|}\begin{bmatrix}\sum_{\mathbf{p}\in\mathcal{P}_i} p_1\mathcal{F}(\mathbf{p})\\ \sum_{\mathbf{p}\in\mathcal{P}_i} p_2\mathcal{F}(\mathbf{p})\\ \sum_{\mathbf{p}\in\mathcal{P}_i} p_3\mathcal{F}(\mathbf{p})\end{bmatrix},\ \mathbf{p}=\begin{bmatrix}p_1\\p_2\\p_3\end{bmatrix}. \tag{1}$$

(See the supplementary material for the proof). 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]. 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.

Feature Extraction Towards obtaining noise resilient $\mathrm{SO}(3)$-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). We adopt the joint usage of DGCNN and Transformer blocks [33], which has been proven to be very efficient in creating high-dimensional embedding for point clouds, and adapt it to the UME framework in order to *learn* multiple high-quality $\mathrm{SO}(3)$ invariant features.

The DGCNN block is designed to preform a per-point embedding, such that information on neighboring points is well incorporated. Each input point in the 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. DGCNN processes 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 joint-resampling 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 (demonstrated in Section 5). Therefore DGCNN architecture in its original design, cannot be employed towards constructing invariant features when rotations by large angles are considered.

We overcome this inherent difficulty by mapping the input point clouds to an alternative coordinates system, which is $\mathrm{SO}(3)$ 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

$$\mathbf{H}^{\mathcal{P}'} = \sum_{\mathbf{p}\in\mathcal{P}'} \mathbf{p}\mathbf{p}^T \tag{2}$$

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.

However, in our setting, where the point clouds to be registered are sparse, differently sampled and noisy, the relation $\mathcal{C}_1 = \mathcal{C}_2$ does not hold anymore. The loss of information caused by low sampling rate makes the resulting representations of the clouds significantly different yielding axes that are no longer co-variant and thus projections that are no longer invariant. Nonetheless, employing the SO(3) invariant representation of the point clouds, the difference between $\mathcal{C}_1$ and $\mathcal{C}_2$ is sufficiently small to enable a learning process of multiple invariant features using a DGCNN block.

While DCP strategy,[38], is to jointly-resample the embedded point clouds in a high-dimensional feature space via a Transformer [37]. In DeepUME we adopt the joint-resampling strategy, but DeepUME resamples the low-dimensional coordinate space, rather than the high-dimensional feature space. We note that resampling point clouds in the coordinate space has a complexity advantage since the dimension of the coordinate space is 3, while the dimension of the embedding space is much higher (512 in [38] and 32 in our case). In terms of architecture blocks, with this sampling approach the Transformer is leading the embedding network (and not the other way around, as in [38]).

We therefore employ the Transformer 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

$$\mathcal{C}_1^{\text{Transformer}} = \mathcal{C}_1 + \phi(\mathcal{C}_1,\mathcal{C}_2), \quad \mathcal{C}_2^{\text{Transformer}} = \mathcal{C}_2 + \phi(\mathcal{C}_2,\mathcal{C}_1). \tag{3}$$

The additive terms, $\phi(\mathcal{C}_1,\mathcal{C}_2)$ and $\phi(\mathcal{C}_2,\mathcal{C}_1)$ are optimized to improve registration, by jointly "equalizing" the sampling patterns of both point clouds.

The resampled point clouds produced using the Transformer are employed for evaluating the UME moments, by re-projecting them on the corresponding principle axes to obtain

$$\widehat{\mathcal{P}}_1 = \mathbf{D}_1 \cdot \mathcal{C}_1^{\text{Transformer}} + \mathbf{m}_{\mathcal{P}_1}, \quad \widehat{\mathcal{P}}_2 = \mathbf{D}_2 \cdot \mathcal{C}_2^{\text{Transformer}} + \mathbf{m}_{\mathcal{P}_2}. \tag{4}$$

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 SO(3)-invariant for differently and sparsely sampled point clouds corrupted by noise. As demonstrated by the experimental results, applying UME registration on the re-projected re-sampled point clouds, provides with an improved performance.

Loss In order to overcome the ambiguity problem in registering symmetric objects, discussed in Section 5, we adopt the Chamfer distance [2] as our loss function. That is, if $\hat{\mathbf{T}}$ is the estimated transformation,

$$\mathcal{L}(\hat{\mathbf{T}}) = d_C(\hat{\mathbf{T}}(\mathcal{P}_1),\mathcal{P}_2). \tag{5}$$

Using this loss, ambiguous symmetric objects 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, which makes it unsupervised.

5 Experiments

We conduct experiments on three data sets: ModelNet40 [41], FAUST [5] and Stanford 3D Scanning Repository [34] where the latter two are used only for testing. We train our network using ModelNet40, which consists of 12,311 CAD models in 40 categories where 80% samples 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 choose a rotation drawn uniformly from the full range of Euler angles and a translation vector in $[-0.5, 0.5]$ in each axis. 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 in the scenario of Bernoulli noise (see bellow), in the specific case where $p_1 = p_2 = 0.5$, and test all scenarios using the trained configuration.

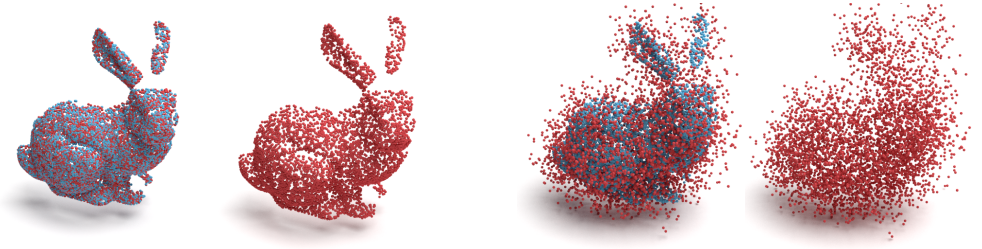
Each experiment is evaluated using four metrics: The RMSE metric, both for the rotation and the translation as well as the Chamfer distance and the Hausdorff distance [22, 36], proposed next as alternative metrics to resolve ambiguity issues. We compare our performances with the basic implementation of the UME method (coded by ourselves), ICP (implemented in Intel Open3D [49]), and four learned methods; PointNetLK and DCP (benchmarks point cloud registration networks) as well as the recently proposed DeepGMR, [46] and RGM, [15]. We retrain the baselines, adapting the code released by the authors. The experimental results are detailed below and summarized in Tables 1, 2; further experimental results are provided in the supplementary.

The ambiguity problem in the registration of symmetric objects 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 guarantees 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 $\mathbf{T}(\mathcal{P}_1)$, 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 man-made objects, the rotation angles RMSE metric may assign large errors to successful registrations. To resolve this ambiguity we replace this metric by the Chamfer and Hausdorff distances defined by

$$\begin{aligned} d_{\mathrm{C}}(\mathcal{P}_1,\mathcal{P}_2) &= \frac{1}{|\mathcal{P}_1|}\sum_{\mathbf{p}\in\mathcal{P}_1}\min_{\mathbf{p}'\in\mathcal{P}_2}\|\mathbf{p}-\mathbf{p}'\|_2 + \frac{1}{|\mathcal{P}_2|}\sum_{\mathbf{p}'\in\mathcal{P}_2}\min_{\mathbf{p}\in\mathcal{P}_1}\|\mathbf{p}'-\mathbf{p}\|_2 \\ d_{\mathrm{H}}(\mathcal{P}_1,\mathcal{P}_2) &= \max_{\mathbf{p}\in\mathcal{P}_1}\min_{\mathbf{p}'\in\mathcal{P}_2}\|\mathbf{p}-\mathbf{p}'\|_2 + \max_{\mathbf{p}'\in\mathcal{P}_2}\min_{\mathbf{p}\in\mathcal{P}_1}\|\mathbf{p}'-\mathbf{p}\|_2. \end{aligned} \tag{6}$$

Using ModelNet40, we test performance in four scenarios: noise free model, sampling noise (Bernoulli noise and zero-intersection noise) and AWGN noise. We note that in the



Zero-intersection noise model Gaussian noise model

Figure 3: Sampling-noise vs. AWGN. Red point clouds are noisy observations of the blue ones, for zero-intersection (left) and AWGN with variance 0.09 (right) noise models. While both achieve the same registration error under DeepUME, the former preserves well the original shape and the latter sever distortion makes it barely recognizable.

presence of sampling noise we indeed observe large errors in RMSE($\mathbf{R}$) due to the symmetry of objects, although the symmetric shapes are well aligned. Moreover, in the unseen data sets, 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.

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 of 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 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. This scenario is visualized in Figure 1 that shows registration results of several baseline methods and our proposed method on a representative example from the unseen data set FAUST.

ModelNet40: AWGN 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. We find that for sufficiently large additive noise variance, the rotation RMSE in both AWGN and sampling-noise models are comparable, yet these different types of noise have very different impact on registration, as illustrated in Figure 3. In order to have the same registration error as in the zero-intersection model on the Stanford data set, an AWGN with variance of approximately 0.09 is required. As shown in Figure 3, such a

Model	Noise free				Bernoulli noise				Gaussian noise			
	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	0.193	<1e-05	0.0581	0.394	74.164	0.015	0.019	0.151	27.684	0.002
ICP [0]	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 [0]	0.027	0.142	80.360	1.0105	0.029	0.157	83.280	1.073	0.026	0.153	81.843	1.037
DCP [33]	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 [46]	<7e-06	<9e-05	0.193	<5e-05	0.033	0.224	72.447	0.018	0.011	0.085	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 stand for Chamfer and Hausdorff distances respectively. The best results are **bold** and the second best are underlined.

Model	ModelNet40 [11]				FAUST [8]				Stanford 3D Scanning Repository [32]			
	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	0.010	0.007	0.085	35.983	0.044	0.033	0.267	48.716	0.010
ICP [0]	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 [0]	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 [33]	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 [46]	0.026	0.117	**67.282**	0.010	0.003	0.027	27.941	0.020	0.005	0.119	39.402	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**	70.818	**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 data sets (FAUST and Stanford 3D Scanning Repository). Our method outperforms the competing techniques in all scenarios for all metrics, only except RMSE(**R**) for ModelNet40.

noise causes a sever distortion, which practically makes the original shape unrecognizable. On the other hand, the original shape of the bunny in the sampling noise scenario, is well preserved.

Unseen data sets 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 data set 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 data set. 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 sparsely and differently sampled point clouds in the presence of noise and large transformations. Our model, named DeepUME, integrates the closed-form UME registration method into a DNN framework. The two are combined into a single unified framework, trained end-to-end and in an unsupervised manner. DeepUME employs an $SO(3)$-invariant coordinate system to learn both a joint-resampling strategy of the point clouds and $SO(3)$-invariant features. The resampling is performed in the low-dimensional coordinate space (rather than in the high-dimensional feature space), to minimize the effect of the sampling noise on the registration performance. The constructed features are utilized by the geometric UME method for transformation estimation. The parameters of DeepUME are optimized using a metric designed to overcome the ambiguity emerging in the registration of symmetric shapes, when noisy scenarios are considered. We show that our hybrid method outperforms state-of-the-art registration methods in various scenarios, and generalizes well to unseen datasets. Future work will extend the proposed method for registration of sub-parts and key-points, incorporating both local and global information to allow the registration of partially overlapping scenes.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[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] Phillip Isola, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. What makes an image memorable? In *CVPR 2011*, pages 145–152. IEEE, 2011.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

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

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

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