

ALReg: Registration of 3D Point Clouds Using Active Learning

Yusuf Huseyin Sahin ^{1,*} , Oguzhan Karabacak ¹, Melih Kandemir ² and Gozde Unal ¹

¹ Department of Computer Engineering, Istanbul Technical University, Maslak, Istanbul 34469, Turkey; karabacako@itu.edu.tr (O.K.); gozde.unal@itu.edu.tr (G.U.)

² Department of Mathematics and Computer Science, University of Southern Denmark, 5230 Odense, Denmark; kandemir@imada.sdu.dk

* Correspondence: sahinyu@itu.edu.tr

Abstract: After the success of deep learning in point cloud segmentation and classification tasks, it has also been adopted as common practice in point cloud registration applications. State-of-the-art point cloud registration methods generally deal with this problem as a regression task to find the underlying rotation and translation between two point clouds. However, given two point clouds, the transformation between them could be calculated using only definitive point subsets from each cloud. Furthermore, training time is still a major problem among the current registration networks, whereas using a selective approach to define the informative point subsets can lead to reduced network training times. To that end, we developed ALReg, an active learning procedure to select a limited subset of point clouds to train the network. Each of the point clouds in the training set is divided into superpoints (small pieces of each cloud) and the training process is started with a small amount of them. By actively selecting new superpoints and including them in the training process, only a prescribed amount of data is used, hence the time needed to converge drastically decreases. We used DeepBBS, FMR, and DCP methods as our baselines to prove our proposed ALReg method. We trained DeepBBS and DCP on the ModelNet40 dataset and FMR on the 7Scenes dataset. Using 25% of the training data for ModelNet and 4% for the 7Scenes, better or similar accuracy scores are obtained in less than 20% of their original training times. The trained models are also tested on the 3DMatch dataset and better results are obtained than the original FMR training procedure.

Keywords: point cloud registration; active learning



Citation: Sahin, Y.H.; Karabacak, O.; Kandemir, M.; Unal, G. ALReg: Registration of 3D Point Clouds Using Active Learning. *Appl. Sci.* **2023**, *13*, 7422. <https://doi.org/10.3390/app13137422>

Academic Editor: Douglas O'Shaughnessy

Received: 20 April 2023

Revised: 10 June 2023

Accepted: 14 June 2023

Published: 22 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, point cloud classification and segmentation methods using deep learning have dominated the field. The pioneer work in the field is the PointNet [1], in which the point positions are fed directly to the network. After PointNet's success, many other methods handled the problem using a deep learning approach [2–4]. Despite the mentioned studies feeding a whole point cloud to the network in the training process, it is shown that not all points in a point cloud have the same contribution to the output of the network [5], and using a relatively smaller amount of points, similar results could be obtained [6]. Thus, especially for large point cloud sets, an active learning-based approach could be adopted to only use a certain count of patches in the training procedure [7–10]. A keypoint-based approach is followed, which also utilizes a relatively smaller number of points while matching the point clouds [3,11].

Point cloud registration, which could be defined as the calculation of the rigid transformation between multiple point clouds and aligning them, is an important problem for many computer vision tasks in robotics, autonomous driving, and 3D reconstruction [12]. The traditional methods, like Iterative Closest Point (ICP) [13] and other methods considering point-to-point or point-to-plane matching using only mathematical constraints [14–16], have been replaced by learning-based methods [3,17,18]. Using the deep feature representation of each point or local patch of the point cloud, feature-based metrics are involved to

gain more robust matching against outliers and noises. Even though point cloud registration has achieved significant progress using the previously mentioned methods, most of them (even in the partial-to-partial point cloud registration [19–21]) are focused on the use of the whole shape to train the network for registration. However, similarly, this problem could also be handled using active learning which was not been addressed in the literature.

Active learning is substantially useful where data labeling is costly, and hence, only a subset of the data ought to be used in the training process. Thus, the most discriminative and diverse subset of the data should be selected. Initially, the training is started with a very small subset of the training data and, using the active learning pipeline, this subset is expanded. A simplified active learning pipeline could be considered as a three-part system: A *predictor*, an *oracle*, and a *guide* [22]. In the common approach, the predictor is a parametric machine learning model (e.g., a deep neural network) which is trained for the task using the labeled data (despite the traditional denomination in segmentation and classification problems, the term labeled data here refers to the data engaged in the training procedure). The guide calculates an acquisition score for each unlabeled data using a heuristic function and/or the feedback obtained from the predictor. This feedback could be simply the uncertainty/discrepancy amount of each unlabeled sample, since a sample for which the network could not reach a decision certainly, will have an impact on the construction of the network when included in the training procedure. Finally, the oracle uses the Guide's scores and labels some of the unlabeled data according to a policy.

Instead of selecting samples from the training set, in the 2D image segmentation methods utilizing active learning [23–25], some parts of each image could also be selected. Considering that the labeling cost of 3D segmentation is higher than its 2D counterparts, some efforts have been devoted to using active learning in point cloud segmentation [26–28]. These studies adopted the superpoint [29] representation where each point cloud is divided into many superpoints using an over-segmentation approach and labeled superpoints in the active learning pipeline.

A similar approach could be applied to point cloud registration. Instead of decreasing the time budget allocated to manually label the dataset, our main aim is to decrease the training time of the network model. Since learning the rigid transform between two point clouds is equivalent to learning the transform between patches of these point clouds, the learning procedure can be altered to its superpoint variant. It is unnecessary to use similarly structured point cloud patches that have resembling features. In this study, we present ALReg, an active learning approach to efficiently use the informative regions for point cloud registration to decrease the training time. As the predictor, the baseline registration networks are modified to use Monte Carlo DropOut (MCDO) [30] to efficiently calculate uncertainty. The guide works on each point cloud separately and calculates acquisition scores for a given point cloud's unlabeled superpoints. The oracle provides a superpoint chosen by the guide.

Our main motivation is that, by using fewer point clouds or point cloud parts in the training phase of any point cloud registration network, a similar accuracy score could be obtained. However, it depends to the efficient selection of these training samples/parts. By the usage of ALReg, we aim the involve the most effective point cloud parts to the training procedure.

To demonstrate the effect of the ALReg, we took three deep learning-based methods as our baselines: Deep Best Buddies (DeepBBS) [31], Feature Metric Registration (FMR) [32] and Deep Closest Point (DCP) [33]. As in the official papers, we evaluated FMR on 7Scenes [34] and the other methods on ModelNet40 [35]. To further investigate our method we also evaluated our trained FMR networks on 3DMatch [18] as unseen data. To demonstrate the performance of ALReg extensively, we used datasets from different domains and structures. We observe in our experiments, that our method matches or outperforms other existing methods, but with significantly reduced computation time.

The three main contributions of this study are as follows:

- We propose ALReg, an active learning pipeline that can be used to drastically decrease training time for registration networks with either a similar performance or even an increase in terms of accuracy. Regarding the drawbacks of the overcalculations for network training for whole-to-whole point cloud registration, ALReg focuses on using only a relevant and adequate subset of superpoints during the process.
- A novel uncertainty-based acquisition function that could be used to calculate superpoint uncertainties is presented. In the previous studies focusing on active learning for point cloud data, class labels were used for uncertainty calculations.
- ALReg is tested on three popular registration methods (DCP, FMR, DeepBBS) for both real (7Scenes, 3DMatch) and synthetic (ModelNet) point cloud datasets. Overall, an improvement over the existing methods in terms of accuracy scores is obtained.

2. Related Work

2.1. Point Cloud Registration

Iterative Closest Point (ICP) [13] is the pioneer method in point cloud registration which is an iterative process aiming to find the closest point correspondences in each transformation iteration and to calculate the current optima using SVD (Singular Value Decomposition). However, ICP is very sensitive to outliers. Despite various extensions to ICP that focus on enhancing the method exists [36–38], learning-based methods which are easy to generalize are commonly used.

As the performance boost of deep neural networks in point cloud segmentation and classification tasks became prominent, deep approaches also dominated the registration task. Using deep features from the point clouds, these methods demonstrate increased robustness and give better performance results. PointNetLK (PointNet Lucas-Kanade) [39] is a hybrid method that extracts global features from the PointNet and iteratively minimizes the distance between feature spaces between the source and target point clouds with the help of the Lukas–Kanade algorithm. SpinNet [40] is an approach that produces rotation-invariant local features from the source and target, and ensures an $SO(2)$ (special orthogonal) equivariance by mapping them onto a cylindrical volume. RGM (Robust Graph Matching) [41] proposes a model to create graphs using node features extracted from the points of each cloud and transforms the problem to graph matching. DeepVCP (Virtual Corresponding Points) [42] generates virtual corresponding points and predicts a relative pose using candidate correspondence probabilities. PointDSC (Deep Spatial Consistency) [43] uses SC-Nonlocal blocks to check spatial consistency between matches. RPM-Net (Robust Point Matching) [44] uses a Sinkhorn layer for coarser assignments to reduce the effect of outliers and enhances the results step-by-step taking advantage of its recurrent neural network structure. PCRNet (Point Cloud Registration Network) [45] uses PointNet-style multi-layer perceptrons for both source and target point clouds in a Siamese architecture model. PPF-FoldNet [46] uses Point Pair Features from PPFNet inside a FoldingNet [47] based autoencoder network to obtain features for local patches and uses those patches to match the clouds. DDRNet (Deep Direct Registration Net) [48] uses a spatially aware encoder to find features for each local region, then applies attention-weighting to weight the local features according to their similarities between each other and finally uses a pyramid-shaped decoder which maps the features to rotation and translation vectors. Fore-Net [49] uses a dual space attention module to map source and target features to a dynamic feature space and a fixed Cartesian space. In PREDATOR (Pointcloud REGistration with Deep ATtention to the Overlap Region) [50], a cross-attention block is presented where each superpoint from the source is connected to all superpoints of the target to form a bipartite graph and utilized message-passing formulation for graph neural networks [51] to control information flow between source and target features. In [52], a Hough voting mechanism is introduced to find a consensus on each correspondence. TriVoC (Triple-layered Voting with Consensus maximization) [53] is inspired by this approach and repeats the voting mechanism in three consecutive layers. In HDRNet (High-Dimensional Regression Network) [54], after feature extraction and point correspondence matrix estimation, a

parametric regression is used to find the rotation quaternion and translation vector. In [55], a reinforcement learning approach is applied where the source point cloud is rotated and translated in many steps and for each step, a reward is defined. In [56], the real-life problem of pallet pose estimation is considered and the registration is done using sample consensus initial alignment and iterative closest point algorithms.

Since we adopt DCP, DeepBBS and FMR as baselines, detailed explanations for these methods will be given in the Method section.

2.2. Active Learning

Active learning [57] is a method to use fewer training labels for a machine learning approach. Although it has been heavily associated with deep learning, it was studied before the deep learning era as well. In [58], an active learning method to select samples from a class-imbalance dataset for Support Vector Machines (SVM) [59] is presented. Similar data-selection methods for SVM have been successfully applied to areas like text classification [60,61] and image retrieval [62].

Here, we will mainly focus on active learning strategies for deep learning and we follow the taxonomy from [27], discuss the active learning methods according to the acquisition functions they use: uncertainty-based, diversity-based and hybrid methods.

In general, maximum entropy sampling and its variances are widely used as the acquisition functions to score uncertainty. BALD (Bayesian Active Learning by Disagreement) [63], using an information-theoretic approach, in addition to uncertainty, measures a selected sample's impact on the network's uncertainty. AL-DL (Active Learning in Deep Learning) [64] uses confidence scoring, margin sampling, and entropy scoring for image classification. In [65], new acquisition functions focusing on samples with maximum variation ratio and mean standard deviation are presented. In CoreGCN (Core Graph Convolutional Network) [66], a graph convolutional network is formed where labeled and unlabeled samples are represented as nodes, and the cross-entropy between each labeled and unlabeled node is minimized. Then, samples are selected according to their uncertainty values. In QUIRE (QUerying Informative and Representative Examples) [67], representativeness and informativeness measures are calculated using network uncertainties.

In [68], a diversity-based active learning mechanism that is inspired by a core-set selection is presented. Using the k-center algorithm on training samples, most representative points are selected according to their position in the feature space.

Hybrid approaches, like BADGE (Batch Active learning by Diverse Gradient Embeddings) [69], create hypothetical labels for unlabeled data using the predictions of neural network and represent their representation in gradient space where large gradients indicate uncertain samples, then they cluster the feature representations of those samples to select a diverse set. In ALBL (Active learning by learning) [70], a multi-armed bandit selects one of the active learning algorithms according to an important-weighted accuracy score. For many general vision tasks like semantic segmentation [71–73], object detection [74] and classification [75] hybrid approaches are used.

2.3. Active Learning for Point Clouds

Recently, active learning gained prominence in 3D point cloud research. In [28], an active learning framework for point cloud segmentation is presented. In this study, three selection methods are investigated (shape-level, superpoint-level and point-level) using a hybrid acquisition function. While a feature-based distance metric is introduced to evaluate the diversity of unlabeled data, entropy is used as an uncertainty score. In ReDAL (Region-Based and Diversity-Aware Active Learning) [26], a super-point level active selection is done for training. Although entropy is used as an uncertainty metric as in the previous study, feature diversity is calculated using color intensity differences and surface variances. Also, a diversity-aware selection is presented to assure diversity while selecting multiple items. In SSDR-AL (Spatial-Structural Diversity Reasoning for Active Learning) [27], the uncertainty score is weighted according to class distributions. Then, Chamfer distance

on the feature-graph space is used to select the most diverse points. In [7], a conditional Markov random field for segmentation is trained using classification errors as scores. In [10], a classification method for terrestrial laser scanning data is presented and a classification confidence metric to select points is used. In [8], four different active learning strategies are evaluated on airborne laser scan (ALS) data: Random sampling, point entropy, segment [76] entropy, and mutual information where segment entropy metric has the best scores. In [9], an ensemble of classifiers is used, and disagreement between each classifier’s predictions is scored for segmenting ALS point clouds. In [77], a multi-granularity active learning pipeline is developed.

As can be seen from the related works, studies on active learning for point clouds entirely focus on the segmentation task. Generally, the main purpose is using fewer point clouds or point cloud parts while training. However, none of these studies have considered active learning usage in point cloud registration. Thus, to the best of our knowledge, ALReg is the first study on point cloud registration with active learning.

3. Method

3.1. Problem Definition

Given two point clouds $X = \{x_1, \dots, x_i, \dots, x_N\} \in R^{N \times 3}$ and $Y = \{y_1, \dots, y_i, \dots, y_N\} \in R^{N \times 3}$, our aim is to find the rigid transformation $[R, t]$ with six degrees of freedom where $R \in SO(3)$ is the rotation matrix and $t \in R^3$ is the translation vector according to the error function:

$$\operatorname{argmin}_{R \in SO(3), t \in R^3} \sum_i^N \|Rx_i + t - y_i\|. \tag{1}$$

The baseline studies have different architectures and methods to minimize error. However, they use all points from a training sample. To train the network using an active learning approach, every sample from the training set $D = \{X_0, \dots, X_i, \dots, X_N\}$ should be decomposed into superpoints $X_i = \{S_{i0}, \dots, S_{ij}, \dots, S_{iM}\} = S_L(X_i) \cup S_U(X_i)$ where M represents superpoint count. S_L and S_U represent sets of labeled and unlabeled superpoints respectively. In ALReg, an active learning approach is followed to efficiently select superpoints from S_U to be used in the training process.

3.2. Baseline Methods

As the baseline models, DCP, DeepBBS and FMR are utilized, whose details are summarized below. The architectural details of these methods are depicted in Figure 1.

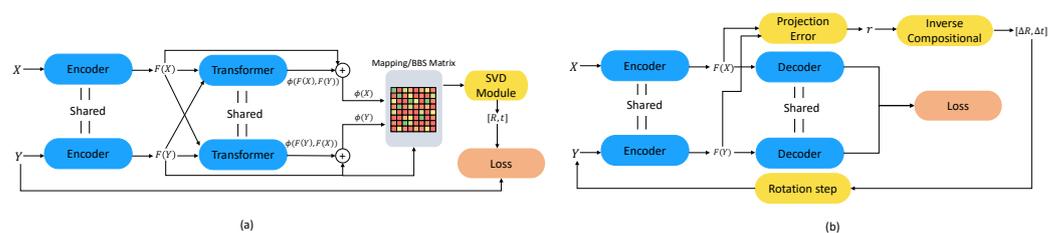


Figure 1. Registration frameworks for (a) DCP [33] and DeepBBS [31], (b) FMR [32]. To ensure simplicity, some details given in the method section are not shown.

DCP: Deep Closest Point framework uses a dynamic graph-based encoder to obtain point cloud features $F(X)$ and $F(Y)$ from source X and target Y . Then, using a transformer [78] with conditional attention, integrated features $\phi(F(X), F(Y))$ and $\phi(F(Y), F(X))$ are obtained. To benefit from both local and global information, new point embeddings are created as

$$\begin{aligned} \phi(X) &= F(X) + \phi(F(X), F(Y)) \\ \phi(Y) &= F(Y) + \phi(F(Y), F(X)). \end{aligned} \tag{2}$$

Then, a probabilistic point-to-point mapping matrix is formed as

$$m(x_i, Y) = \text{softmax}(\phi(Y)\phi(x_i)^T)$$

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, 2, \dots, K \quad (3)$$

where $m(x_i, Y)$ represents a row of the matrix indicating the probabilistic distribution of x_i 's corresponding point among the points of Y . For every point x_i , the predicted corresponding point \hat{y}_i is found, and the cross-covariance matrix $H(X, Y)$ is filled as

$$\hat{y}_i = Y^T m(x_i, Y), \quad H(X, y) = \sum_i^N (x_i - \bar{x})(\hat{y}_i - \bar{y}). \quad (4)$$

Using singular value decomposition (SVD) on $H(X, y) = USV^T$, R and t are found by $R = VU^T$ and $t = -R\bar{x} + \bar{y}$.

DeepBBS: Deep Best Buddies could be considered as an extension to DCP where x_i and y_j are called best buddies if they are the closest points to each other according to their features $\phi(x_i)$ and $\phi(y_j)$. Given the feature distance matrix $D_{i,j}$, the BBS score between two points is calculated as

$$B(x_i, y_j) = \frac{e^{-\frac{D_{ij}}{\alpha}}}{\sum_{i'} e^{-\frac{D_{i'j}}{\alpha}}} \cdot \frac{e^{-\frac{D_{ij}}{\alpha}}}{\sum_{j'} e^{-\frac{D_{ij'}}{\alpha}}}. \quad (5)$$

Similarly to the approach in (6), predicted correspondences and cross-covariance matrix are calculated as

$$\hat{y}_i = Y^T \frac{B_i}{\sum_j B_{ij}}, \quad H(X, y) = \sum_i^N \gamma_i (x_i - \bar{x})(\hat{y}_i - \bar{y}) \quad (6)$$

where γ_i indicates correspondence weights. For the remaining part of the method, steps in DCP are applied.

FMR: Contrary to the other two studies used as baselines for this work, the feature metric registration framework does not find correspondences between X and Y . Instead, it focuses on minimizing the feature distance between the source and the target after the alignment is done. To obtain the point features, a PointNet-like encoder is used without a transformation. Due to the nature of the PointNet, the obtained features are rotation-attentive. A decoder is also trained to obtain meaningful features $F(X)$ and $F(Y)$ in an unsupervised way. Here, the projection error was defined as

$$e = \|F(X) - F(RY + t)\|, \quad (7)$$

where e is fed into the inverse compositional algorithm [79] to stepwise increment the transformation as

$$[\Delta R, \Delta t] = (J^T J)^{-1} (J^T e), \quad (8)$$

where J represents Jacobian of e with respect to R and t .

3.3. Active Selection

For the training set \mathcal{D} , very few labeled superpoints $\mathcal{S}_{\mathcal{L}}$ are initially selected randomly, and the selected registration network is trained with these labeled superpoints. At certain points of the training process, an acquisition score is calculated for each superpoint in \mathcal{D}_U and according to their sorted scores, some of them are included in \mathcal{D}_L . It should be remarked that, for every selection phase, only one unlabeled superpoint from each training sample is selected. Otherwise, the algorithm may select the newly labeled superpoints so that there could be an unbalance among point counts. Considering that our main focus for ALReg is to decrease training time, creating batches for variable-sized samples will be more time-consuming than the mentioned setup. The full framework is depicted in Figure 2.

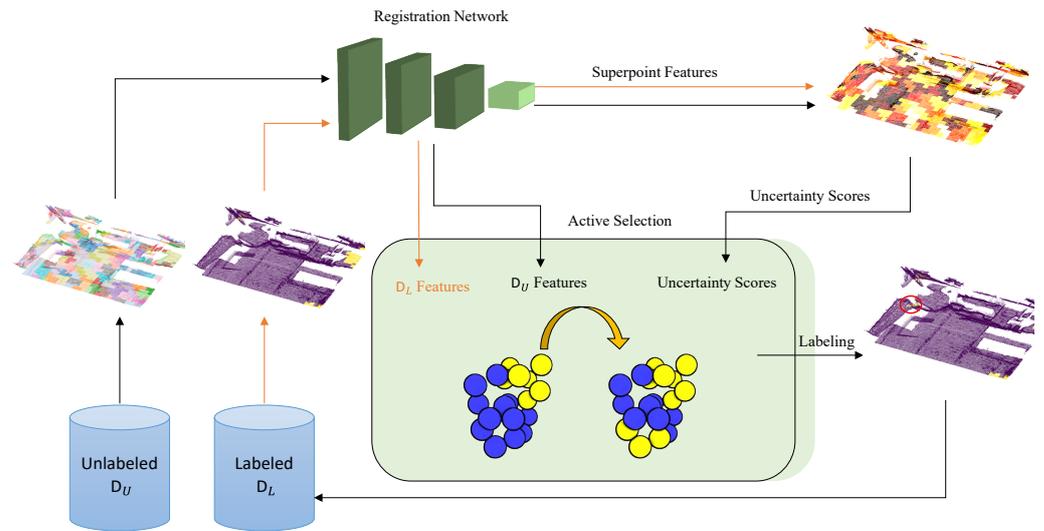


Figure 2. The ALReg Framework. In training time, only the labeled superpoints D_L are used. In the active selection phase, uncertainty scores and network features are obtained to select new superpoints for labeling.

In this study, we present two different acquisition scoring mechanisms: $ALReg_{DIV}$ and $ALReg_{UNC}$ focusing on diversity and uncertainty respectively.

ALReg_{DIV}: To calculate the diversity of a superpoint, the acquisition functions in ReDAL and SSDR-AL are unsuitable here since ReDAL benefits from color discontinuity and SSDR-AL uses class distributions. Thus, to obtain diversity scores, following a similar approach to [28], a feature diversity metric is calculated. Diversity score of an unlabeled superpoint $S_i \in S_U(X_j)$ is calculated as

$$D(s) = \min_{s_k \in S_L(X_j)} \|F(s_i) - F(s_k)\| \tag{9}$$

where F denotes the encoder features from the selected network.

ALReg_{UNC}: To calculate the uncertainty of a superpoint, a Monte Carlo Dropout (MCDO) [80] approach is followed. Since dropout is a specific type of Bayesian neural network (BNN), after each layer of the encoder layer, a dropout layer is placed to obtain varying outputs for the same points. Since it is not a classification task, the uncertainty metrics presented before are unusable. Thus, a variance-based uncertainty metric used for BNNs with latent variables [81,82] could be applied to our problem.

For the data point x and the network output y , total uncertainty in terms of variance could be written as

$$\sigma^2(y|x) = \sigma_{q(W)}^2(E[y|W,x]) + E_{q(W)}[\sigma^2(y|W,x)] \tag{10}$$

where $q(W)$ represents the posterior approximation according to weights W . Here, the first term represents epistemic uncertainty and the second term represents aleatoric uncertainty. The aleatoric uncertainty term could be calculated by $M \times N$ network passes. The network weights W are sampled M times (corresponding to using M different MCDO setups) and the variable is sampled N times (corresponds to use N different point sets from a selected superpoint). For a feature vector with size K , means of variances are calculated for each of the K features and their average will indicate the uncertainty of the selected superpoint $U(s)$.

The procedure is explained in Algorithm 1.

Algorithm 1: Acquisition function

Data: Labeled superpoints of X_i : $S_L(X_i)$, An unlabeled superpoint from X_i : s ,
 Feature extractor: f_θ , a set of fixed DropOuts: $\{\delta_1, \dots, \delta_M\}$

Result: Acquisition scores: $D(s), U(s)$

$features[0 \dots M, 0 \dots N] = 0;$

for $k \leftarrow 1$ to N **do**

$s_{sampled} = sample(s);$

for $l \leftarrow 1$ to M **do**

$f_{\theta, DO} = setDropOut(F_\theta, \delta_l);$

// Apply the predefined DropOut δ_l to the network.

$features[l, k] = f_{\theta, DO}(S_{sampled});$

end

end

$U(s) = var(mean(features));$

$features_s = f_\theta(s);$

$features_{S_L(X_i)} = f_\theta(S_L(X_i));$

$D(s) = min(features_{S_L(X_i)} features_s^T);$

4. Experiments

To evaluate our method, we have done experiments using ModelNet40 (for DCP and DeepBBS) and 7Scenes (for FMR) datasets. We test the ALReg method against random selection (RAND) and also report the original results (FULL) obtained using the official source code. We performed all the experiments on a computer with an NVIDIA RTX A5000 GPU and Intel(R) Core(TM) i7-7700K CPU @ 4.20 GHz.

We used the default hyperparameters of the mentioned methods. In FMR we used Chamfer loss and geometric loss. In DCP we used an MSE based loss as

$$L_{DCP} = \|R_{gt}^T R_{pred} - I\|^2 + \|t_{gt} - t_{pred}\|^2 + \lambda \|\theta\|^2 \quad (11)$$

where R_{gt} and t_{gt} refer to ground truth rotation matrix and translation vector and R_{pred} and t_{pred} refer to their predictions. The last part is a network regularization. For DeepBBS, the used loss function is

$$L_{DeepBBS} = \|R_{gt}^T R_{pred} - I\|^2 + \|t_{gt} - t_{pred}\|^2 + \beta^n \frac{1}{N} \sum_{i=1}^N \gamma_{gt,i} \|q_i - (R_{gt} p_i + t_{gt})\|^2 \quad (12)$$

where $\gamma_{gt,i}$ is an indicator showing whether p_i and q_i are matching points in the source and target point clouds, β is a multiplier decreasing according to epoch number n . Here β is selected as 0.95. For all experiments, the Adam [83] optimizer is used with a learning rate of 0.001 and the batch size is selected as 32.

The only difference between our setup and the baselines is placing dropouts with $p = 0.25$ after each layer of the encoders. After the last active selection phase, the dropouts are removed from the networks for better convergence. Active selection is done after epochs [20, 50, 70] for DCP/DeepBBS and after epochs [40, 60, 80] for FMR.

For DCP (the second version containing a transformer) and FMR, the official weights are directly used. However, DeepBBS which originally uses 786 sampled points in both train and test time, is retrained using 1024 points to ensure a fair comparison. For both networks, using the common procedure, a rotation of $[0, \frac{\pi}{4}]$ and translation of $[-0.5, 0.5]$ is used. For FMR, a rotation of $[0, \frac{\pi}{3}]$ and translation of $[-0.5, 0.5]$ is used following the original setup.

7Scenes is an indoor 3D scan dataset that contains 296 objects having an average of points and it has 57 test point cloud objects. Since 7Scenes is a dense dataset, in full FMR, randomly sampled 10,000 points are used during training and testing. In our active

learning setup, we used VCSS [84] algorithm to divide each object into ~ 150 superpoints. For each superpoint, 20 sampled points from the superpoint are fed to the network. Initially, 3 superpoints (60 points) of each object are used for training. After the last active selection phase, only 4.16% of the training set is selected.

ModelNet40 is a CAD dataset of daily life objects containing 9840 training samples and 2468 test samples from 40 different categories. In the aforementioned registration methods, it is a rule of thumb to use 1024 points from CAD model for each object. Using random sampling, ~ 100 superpoints are created for each object as given in Figure 3.

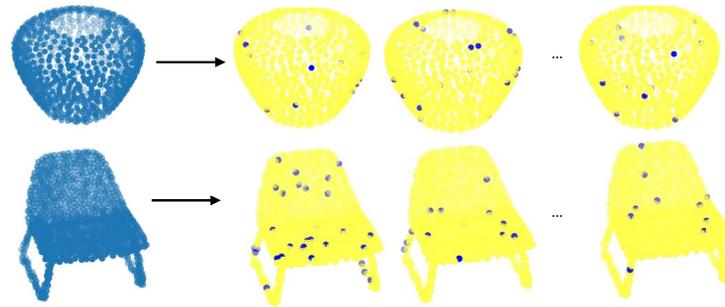


Figure 3. Randomly created superpoints for ModelNet40 dataset.

After training the networks with the mentioned datasets, we also evaluated their accuracy on 3DMatch to show their consistency on unseen data. 3DMatch is a real-life scan dataset containing a total of 520 samples in its test set. Since we evaluated the dataset with the FMR setup, color information is not used and each fragment is subsampled to contain 10,000 points.

To evaluate our method, adopting the metrics from DCP [33], mean square error (MSE), root MSE (RMSE) and mean absolute error (MAE) for rotation and transformation are used.

For the FMR training setup, the results on 7Scenes and 3DMatch datasets are given in Table 1. As can be seen from the results, using only a limited amount of training data with both $ALReg_{DIV}$ and $ALReg_{UNC}$, better results are obtained than random selection for 7Scenes. Also, all methods outperformed the full train scenario, which demonstrates that using all points leads to confusion for the network. Evaluating the network on 3DMatch, $ALReg_{UNC}$ method still outperforms random selection.

Table 1. FMR training results for 7scene and 3DMatch datasets. ALReg method is also compared with original full-size registration.

FMR-7Scenes						
Method	MSE (R)	RMSE (R)	MAE (R)	MSE (t)	RMSE (t)	MAE (t)
FULL	0.39343	0.62724	0.25740	0.00614	0.07835	0.03035
RAND	0.2110 ± 0.0308	0.4581 ± 0.0344	0.1461 ± 0.0211	0.0036 ± 0.0009	0.0594 ± 0.0080	0.0181 ± 0.0038
$ALReg_{DIV}$	0.1524 ± 0.0551	0.3842 ± 0.0692	0.1196 ± 0.0308	0.0131 ± 0.0162	0.0471 ± 0.0099	0.0152 ± 0.0041
$ALReg_{UNC}$	0.1463 ± 0.0203	0.3816 ± 0.0266	0.1167 ± 0.0143	0.0021 ± 0.0002	0.0453 ± 0.0018	0.0139 ± 0.0011
FMR-3DMatch						
Method	MSE (R)	RMSE (R)	MAE (R)	MSE (t)	RMSE (t)	MAE (t)
FULL	0.35491	0.59574	0.26926	0.01624	0.12745	0.04916
RAND	0.1834 ± 0.0019	0.4283 ± 0.0022	0.1605 ± 0.0017	0.0070 ± 0.0004	0.0839 ± 0.0022	0.0275 ± 0.0002
$ALReg_{DIV}$	0.1969 ± 0.0252	0.4428 ± 0.0281	0.1726 ± 0.0178	0.0078 ± 0.0005	0.0885 ± 0.0030	0.0298 ± 0.0021
$ALReg_{UNC}$	0.1681 ± 0.0054	0.4099 ± 0.0065	0.1481 ± 0.0028	0.0066 ± 0.0002	0.0814 ± 0.0010	0.0256 ± 0.0006

In Figure 4, the superpoints actively selected by the different methods are visualized. Qualitatively, both $ALReg_{DIV}$ and $ALReg_{UNC}$ selected a more representative set compared to the random selection. In fact, there are many common superpoints. Considering the locations of these superpoints, it could be asserted that, selecting superpoints from the borders is a better approach than selecting randomly.

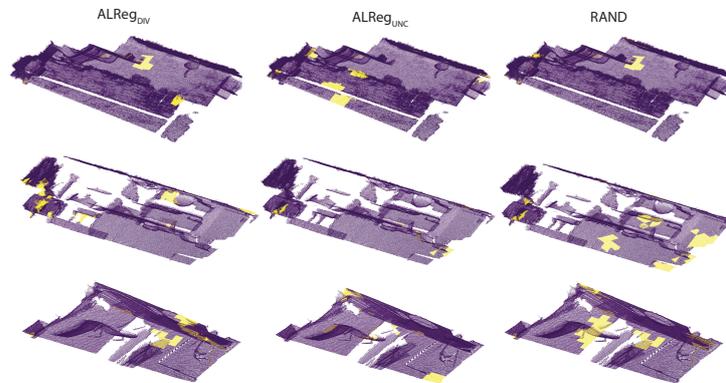


Figure 4. Selected superpoints for 7Scenes dataset with different mechanisms.

We also evaluated ALReg’s performance on FMR according to the amount of rotation starting from $[0, 0, 0]$ to $[\frac{\pi}{2}, \frac{\pi}{2}, \frac{\pi}{2}]$. According to Figure 5, even though their metric performances are worse, it is observed that for larger rotations on 3DMatch, both ALReg methods provide reduced errors than the full point cloud training.

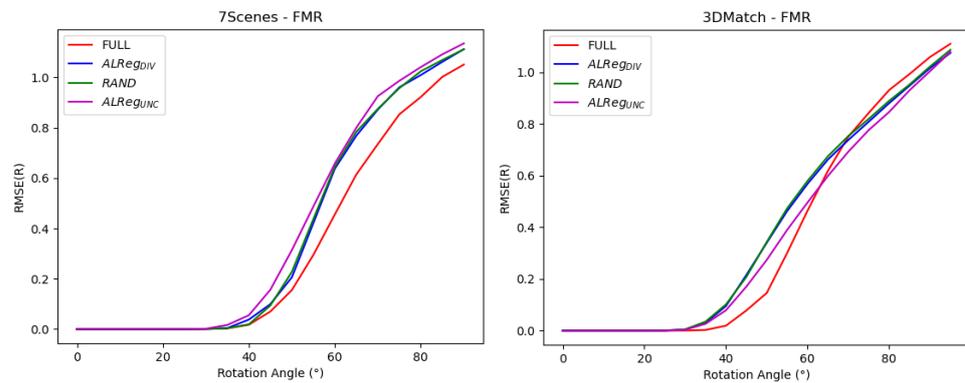


Figure 5. Results for FMR according to different rotation amounts.

For DCP and DeepBBS methods, the numerical results are given in Table 2. According to the results, our $ALReg_{UNC}$ performs slightly behind performance than the full train procedure in all cases.

Table 2. ModelNet40 results for DeepBBS and DCP methods. ALReg method is also compared with original full-size registration.

DCP-ModelNet40						
Method	MSE (R)	RMSE (R)	MAE (R)	MSE (t)	RMSE (t)	MAE (t)
FULL	1.3073	1.1433	0.7705	0.0000	0.0017	0.0011
RAND	2.3273 ± 0.119	1.5255 ± 0.024	1.0506 ± 0.012	$0.0000 \pm 5.61 \times 10^{-9}$	$0.0033 \pm 3.28 \times 10^{-5}$	$0.0023 \pm 2.31 \times 10^{-5}$
$ALReg_{DIV}$	$1.9677 \pm 7.78 \times 10^{-4}$	$1.4027 \pm 2.75 \times 10^{-4}$	$1.4027 \pm 5.11 \times 10^{-3}$	$0.0000 \pm 1.44 \times 10^{-7}$	$0.0034 \pm 8.97 \times 10^{-9}$	$0.0023 \pm 1.31 \times 10^{-6}$
$ALReg_{UNC}$	1.8217 ± 0.127	1.3497 ± 0.026	0.9474 ± 0.0133	$0.0000 \pm 4.32 \times 10^{-9}$	$0.0032 \pm 3.26 \times 10^{-5}$	$0.0022 \pm 2.27 \times 10^{-5}$
DeepBBS-ModelNet40						
Method	MSE (R)	RMSE (R)	MAE (R)	MSE (t)	RMSE (t)	MAE (t)
FULL	1.67×10^{-6}	1.29×10^{-3}	5.18×10^{-4}	1.16×10^{-10}	1.07×10^{-5}	6.55×10^{-6}
RAND	$2.39 \times 10^{-6} \pm 1.27 \times 10^{-6}$	$1.54 \times 10^{-3} \pm 3.74 \times 10^{-4}$	$7.63 \times 10^{-4} \pm 7.56 \times 10^{-5}$	$1.45 \times 10^{-10} \pm 5.51 \times 10^{-11}$	$1.20 \times 10^{-5} \pm 2.43 \times 10^{-6}$	$8.45 \times 10^{-6} \pm 1.09 \times 10^{-6}$
$ALReg_{DIV}$	$4.32 \times 10^{-6} \pm 2.51 \times 10^{-6}$	$2.07 \times 10^{-3} \pm 5.89 \times 10^{-4}$	$7.78 \times 10^{-4} \pm 5.69 \times 10^{-5}$	$1.89 \times 10^{-10} \pm 1.51 \times 10^{-10}$	$1.37 \times 10^{-5} \pm 5.09 \times 10^{-6}$	$8.18 \times 10^{-6} \pm 4.23 \times 10^{-7}$
$ALReg_{UNC}$	$2.36 \times 10^{-6} \pm 4.90 \times 10^{-8}$	$1.53 \times 10^{-3} \pm 1.59 \times 10^{-5}$	$6.87 \times 10^{-4} \pm 8.99 \times 10^{-6}$	$9.61 \times 10^{-11} \pm 1.12 \times 10^{-13}$	$9.80 \times 10^{-6} \pm 4.67 \times 10^{-9}$	$6.34 \times 10^{-6} \pm 1.00 \times 10^{-8}$

We also evaluated ALReg’s robustness against Gaussian noise as well as incomplete point clouds and used 7Scenes and 3DMatch as testbeds. For Gaussian noise, we jittered each point cloud with a noise of $\mathcal{N}(0, 0.1)$. For incomplete point clouds, we randomly select 90% of each point cloud. The results for each study are as given in Tables 3 and 4. In both cases, ALReg’s performance is practically similar to other methods.

Table 3. Gaussian Noise results for FMR.

FMR-7Scene						
Method	MSE (R)	RMSE (R)	MAE (R)	MSE (t)	RMSE (t)	MAE (t)
FULL	0.097413	0.312111	0.124632	0.002955	0.054367	0.022471
RAND	0.095495	0.309023	0.115626	0.006207	0.078784	0.026926
<i>ALReg_{DIV}</i>	0.177675	0.421516	0.144604	0.004147	0.064401	0.022502
<i>ALReg_{UNC}</i>	0.092122	0.303516	0.112774	0.004568	0.067588	0.024137
FMR-3Dmatch						
Method	MSE (R)	RMSE (R)	MAE (R)	MSE (t)	RMSE (t)	MAE (t)
FULL	0.128283	0.358167	0.145229	0.005049	0.071057	0.027608
RAND	0.216643	0.465449	0.196329	0.006501	0.080629	0.029618
<i>ALReg_{DIV}</i>	0.220736	0.469826	0.201843	0.007248	0.085136	0.031431
<i>ALReg_{UNC}</i>	0.190331	0.436269	0.165844	0.007457	0.086357	0.029284

Table 4. Sampled point cloud registration results for DeepBBS and DCP methods.

FMR-7Scene						
Method	MSE (R)	RMSE (R)	MAE (R)	MSE (t)	RMSE (t)	MAE (t)
FULL	0.031967	0.178795	0.053182	0.001211	0.034813	0.012048
RAND	0.048775	0.220852	0.057202	0.001240	0.035215	0.008836
<i>ALReg_{DIV}</i>	0.083325	0.288661	0.059582	0.003184	0.056435	0.010117
<i>ALReg_{UNC}</i>	0.025892	0.160910	0.037307	0.000107	0.010366	0.003278
FMR-3Dmatch						
Method	MSE (R)	RMSE (R)	MAE (R)	MSE (t)	RMSE (t)	MAE (t)
FULL	0.136838	0.369917	0.150323	0.005441	0.073767	0.026348
RAND	0.184270	0.429267	0.176583	0.007063	0.084046	0.029235
<i>ALReg_{DIV}</i>	0.176006	0.419531	0.173010	0.006890	0.083011	0.028737
<i>ALReg_{UNC}</i>	0.136912	0.370016	0.130666	0.005077	0.071257	0.021303

In Table 5, a comparison of training times for the full train and ALReg is given. Using limited data, ALReg achieves to decrease in the training time by 25%, 18% and 20% for FMR, DCP and DeepBBS respectively.

Table 5. Train time of the studied methods in hours for full train mode and ALReg.

	Full Train (h)	ALReg (h)
FMR	0.20	0.05
DCP	8.51	1.56
DeepBBS	12.53	2.56

5. Discussion

Our experimental results demonstrate that ALReg achieves a very close performance in 3D point cloud registration tasks by using only a small portion of the data. The difference of error between full point cloud training or training using only actively selected point cloud parts is relatively small considering the active areas to use. This provides evidence to our hypothesis that following either an uncertainty-based or a diversity-based

selection mechanism, highly beneficial point cloud parts could be selected compared to random selection.

Moreover, it can be thought that using only a few point cloud parts hinders the robustness of the model. However, after applying Gaussian noise or using lesser corresponding points, a drastic change is not observed, which indicates that models trained with ALReg procedure is robust against point cloud malformations.

The acquisition functions for both *ALReg_{DIV}* and *ALReg_{UNC}* methods are easy to calculate during the training time and they could be adapted to other point cloud registration methods using deep learning.

6. Conclusions

We have presented ALReg, an active learning point cloud registration framework that requires less training time and memory relative to existing point cloud registration methods in the literature. According to the experimental results, we demonstrated that using ALReg, whole-to-whole point cloud registration networks could be trained to obtain similar or better results while facilitating speed-ups.

ALReg is the first study to integrate point cloud registration in an active learning approach and introduces baselines. It will stimulate future registration studies with active learning toward efficient model training. As a future work, an adaptation of ALReg to partial registration problems will be investigated.

Author Contributions: Conceptualization, G.U. and M.K.; methodology, Y.H.S. and O.K.; software, Y.H.S. and O.K.; validation, Y.H.S. and O.K.; formal analysis, G.U. and Y.H.S.; investigation, Y.H.S. and O.K.; resources, Y.H.S.; data curation, Y.H.S.; writing—Original draft preparation, Y.H.S.; writing—Review and editing, Y.H.S.; visualization, Y.H.S.; supervision, G.U. and M.K.; project administration, G.U.; funding acquisition, G.U. and M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in the official repositories for [18,34,35].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
2. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5099–5108.
3. Deng, H.; Birdal, T.; Ilic, S. Ppfnet: Global context aware local features for robust 3d point matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 195–205.
4. Liu, Y.; Fan, B.; Meng, G.; Lu, J.; Xiang, S.; Pan, C. DensePoint: Learning Densely Contextual Representation for Efficient Point Cloud Processing. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5239–5248.
5. Sahin, Y.H.; Mertan, A.; Unal, G. ODFNet: Using orientation distribution functions to characterize 3D point clouds. *Comput. Graph.* **2022**, *102*, 610–618. [[CrossRef](#)]
6. Xu, X.; Lee, G.H. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13706–13715.
7. Jiang, T.; Wang, Y.; Tao, S.; Li, Y.; Liu, S. Integrating active learning and contextually guide for semantic labeling of LiDAR point cloud. In Proceedings of the 2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS), Beijing, China, 19–20 August 2018; pp. 1–7.
8. Lin, Y.; Vosselman, G.; Cao, Y.; Yang, M.Y. Active and incremental learning for semantic ALS point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 73–92. [[CrossRef](#)]

9. Kölle, M.; Walter, V.; Schmohl, S.; Soergel, U. Remembering both the machine and the crowd when sampling points: Active learning for semantic segmentation of ALS point clouds. In *Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 505–520.
10. Weidner, L.; Walton, G.; Kromer, R. Generalization considerations and solutions for point cloud hillslope classifiers. *Geomorphology* **2020**, *354*, 107039. [[CrossRef](#)]
11. Li, J.; Lee, G.H. Usip: Unsupervised stable interest point detection from 3d point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 361–370.
12. Zhang, Z.; Dai, Y.; Sun, J. Deep learning based point cloud registration: An overview. *Virtual Real. Intell. Hardw.* **2020**, *2*, 222–246. [[CrossRef](#)]
13. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In Proceedings of the Sensor Fusion IV: Control Paradigms and Data Structures, Boston, MA, USA, 12–15 November 1992 ; Volume 1611, pp. 586–606.
14. Segal, A.; Haehnel, D.; Thrun, S. Generalized-icp. In Proceedings of the Robotics: Science and Systems, Seattle, WA, USA, 28 June–1 July 2009; Volume 2, p. 435.
15. Low, K.L. Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill Univ. North Carol.* **2004**, *4*, 1–3.
16. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2241–2254. [[CrossRef](#)] [[PubMed](#)]
17. Yew, Z.J.; Lee, G.H. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 17–24 May 2018 ; pp. 607–623.
18. Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; Funkhouser, T. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
19. Xu, H.; Liu, S.; Wang, G.; Liu, G.; Zeng, B. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 3132–3141.
20. Lee, D.; Hamsici, O.C.; Feng, S.; Sharma, P.; Gernoth, T. DeepPRO: Deep Partial Point Cloud Registration of Objects. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 5683–5692.
21. Qin, H.; Zhang, Y.; Liu, Z.; Chen, B. Rigid Registration of Point Clouds Based on Partial Optimal Transport. In *Computer Graphics Forum*; Wiley Online Library: New York, NY, USA, 2022.
22. Haußmann, M.; Hamprecht, F.A.; Kandemir, M. Deep active learning with adaptive acquisition. *arXiv* **2019**, arXiv:1906.11471.
23. Cai, L.; Xu, X.; Liew, J.H.; Foo, C.S. Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 10988–10997.
24. Xu, M.; Zhao, Q.; Jia, S. Multiview Spatial-Spectral Active Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–15. [[CrossRef](#)]
25. Qiao, Y.; Zhu, J.; Long, C.; Zhang, Z.; Wang, Y.; Du, Z.; Yang, X. Cpral: Collaborative panoptic-regional active learning for semantic segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 2108–2116.
26. Wu, T.H.; Liu, Y.C.; Huang, Y.K.; Lee, H.Y.; Su, H.T.; Huang, P.C.; Hsu, W.H. ReDAL: Region-based and Diversity-aware Active Learning for Point Cloud Semantic Segmentation. In Proceedings of the IEEE/CVF INTERNATIONAL Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15510–15519.
27. Shao, F.; Luo, Y.; Liu, P.; Chen, J.; Yang, Y.; Lu, Y.; Xiao, J. Active Learning for Point Cloud Semantic Segmentation via Spatial-Structural Diversity Reasoning. *arXiv* **2022**, arXiv:2202.12588.
28. Shi, X.; Xu, X.; Chen, K.; Cai, L.; Foo, C.S.; Jia, K. Label-efficient point cloud semantic segmentation: An active learning approach. *arXiv* **2021**, arXiv:2101.06931.
29. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4558–4567.
30. Kendall, A.; Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach City, CA, USA, 4–9 December 2017.
31. Hezroni, I.; Drory, A.; Giryas, R.; Avidan, S. DeepBBS: Deep Best Buddies for Point Cloud Registration. In Proceedings of the 2021 International Conference on 3D Vision (3DV), Online, 1–3 December 2021; pp. 342–351.
32. Huang, X.; Mei, G.; Zhang, J. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11366–11374.
33. Wang, Y.; Solomon, J.M. Deep closest point: Learning representations for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seattle, WA, USA, 14–19 June 2020; pp. 3523–3532.
34. Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; Fitzgibbon, A. Scene coordinate regression forests for camera relocalization in RGB-D images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2930–2937.

35. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1912–1920.
36. Li, J.; Hu, Q.; Zhang, Y.; Ai, M. Robust symmetric iterative closest point. *ISPRS J. Photogramm. Remote Sens.* **2022**, *185*, 219–231. [[CrossRef](#)]
37. Rusinkiewicz, S. A symmetric objective function for ICP. *Acm Trans. Graph. (TOG)* **2019**, *38*, 1–7. [[CrossRef](#)]
38. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 28 May–1 June 2001; pp. 145–152.
39. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. Pointnetlk: Robust & efficient point cloud registration using pointnet. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7163–7172.
40. Ao, S.; Hu, Q.; Yang, B.; Markham, A.; Guo, Y. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 11753–11762.
41. Fu, K.; Liu, S.; Luo, X.; Wang, M. Robust point cloud registration framework based on deep graph matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2021; pp. 8893–8902.
42. Lu, W.; Wan, G.; Zhou, Y.; Fu, X.; Yuan, P.; Song, S. Deepvcv: An end-to-end deep neural network for point cloud registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 12–21.
43. Bai, X.; Luo, Z.; Zhou, L.; Chen, H.; Li, L.; Hu, Z.; Fu, H.; Tai, C.L. Pointdsc: Robust point cloud registration using deep spatial consistency. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–25 June 2021; pp. 15859–15869.
44. Yew, Z.J.; Lee, G.H. Rpm-net: Robust point matching using learned features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11824–11833.
45. Sarode, V.; Li, X.; Goforth, H.; Aoki, Y.; Srivatsan, R.A.; Lucey, S.; Choset, H. Pcnnet: Point cloud registration network using pointnet encoding. *arXiv* **2019**, arXiv:1908.07906.
46. Deng, H.; Birdal, T.; Ilic, S. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 602–618.
47. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 206–215.
48. Zhang, Z.; Chen, G.; Wang, X.; Shu, M. DDRNet: Fast point cloud registration network for large-scale scenes. *ISPRS J. Photogramm. Remote Sens.* **2021**, *175*, 184–198. [[CrossRef](#)]
49. Zhang, Z.; Chen, G.; Wang, X.; Shu, M. Fore-Net: Efficient inlier estimation network for large-scale indoor scenario. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 165–176. [[CrossRef](#)]
50. Huang, S.; Gojicic, Z.; Usvyatsov, M.; Wieser, A.; Schindler, K. Predator: Registration of 3d point clouds with low overlap. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4267–4276.
51. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
52. Lee, J.; Kim, S.; Cho, M.; Park, J. Deep hough voting for robust global registration. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15994–16003.
53. Sun, L.; Deng, L. TriVoC: Efficient Voting-based Consensus Maximization for Robust Point Cloud Registration with Extreme Outlier Ratios. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4654–4661. [[CrossRef](#)]
54. Gao, J.; Zhang, Y.; Liu, Z.; Li, S. HDRNet: High-Dimensional Regression Network for Point Cloud Registration. In *Computer Graphics Forum*; Wiley Online Library: New York, NY, USA, 2022.
55. Chen, B. Point Cloud Registration via Heuristic Reward Reinforcement Learning. *Stats* **2023**, *6*, 268–278. [[CrossRef](#)]
56. Shao, Y.; Fan, Z.; Zhu, B.; Lu, J.; Lang, Y. A Point Cloud Data-Driven Pallet Pose Estimation Method Using an Active Binocular Vision Sensor. *Sensors* **2023**, *23*, 1217. [[CrossRef](#)]
57. Settles, B. Active Learning Literature Survey. In *Computer Sciences Technical Report 1648*; University of Wisconsin–Madison: Madison, WI, USA, 2009.
58. Ertekin, S.; Huang, J.; Bottou, L.; Giles, L. Learning on the border: Active learning in imbalanced data classification. In Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, Lisbon, Portugal, 6–10 November 2007; pp. 127–136.
59. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
60. Tong, S.; Koller, D. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* **2001**, *2*, 45–66.
61. Lewis, D.D.; Gale, W.A. A sequential algorithm for training text classifiers. In *SIGIR'94*; Springer: Berlin/Heidelberg, Germany, 1994; pp. 3–12.

62. Vijayanarasimhan, S.; Grauman, K. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 2262–2269.
63. Hounsby, N.; Huszár, F.; Ghahramani, Z.; Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv* **2011**, arXiv:1112.5745.
64. Wang, D.; Shang, Y. A new active labeling method for deep learning. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 112–119.
65. Gal, Y.; Islam, R.; Ghahramani, Z. Deep bayesian active learning with image data. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 1183–1192.
66. Caramalau, R.; Bhattarai, B.; Kim, T.K. Sequential graph convolutional network for active learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 9583–9592.
67. Huang, S.J.; Jin, R.; Zhou, Z.H. Active learning by querying informative and representative examples. *Adv. Neural Inf. Process. Syst.* **2010**, *23*, 1–9. [[CrossRef](#)] [[PubMed](#)]
68. Sener, O.; Savarese, S. Active learning for convolutional neural networks: A core-set approach. *arXiv* **2017**, arXiv:1708.00489.
69. Ash, J.T.; Zhang, C.; Krishnamurthy, A.; Langford, J.; Agarwal, A. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv* **2019**, arXiv:1906.03671.
70. Hsu, W.N.; Lin, H.T. Active learning by learning. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
71. Cai, L.; Xu, X.; Zhang, L.; Foo, C.S. Exploring Spatial Diversity for Region-Based Active Learning. *IEEE Trans. Image Process.* **2021**, *30*, 8702–8712. [[CrossRef](#)]
72. Sreenivasaiah, D.; Otterbach, J.; Wollmann, T. MEAL: Manifold Embedding-based Active Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 1029–1037.
73. Stilla, U.; Xu, Y. Change detection of urban objects using 3D point clouds: A review. *ISPRS J. Photogramm. Remote Sens.* **2023**, *197*, 228–255. [[CrossRef](#)]
74. Kellenberger, B.; Marcos, D.; Lobry, S.; Tuia, D. Half a percent of labels is enough: Efficient animal detection in UAV imagery using deep CNNs and active learning. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9524–9533. [[CrossRef](#)]
75. Bengar, J.Z.; van de Weijer, J.; Fuentes, L.L.; Raducanu, B. Class-Balanced Active Learning for Image Classification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 4–8 January 2022; pp. 1536–1545.
76. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [[CrossRef](#)]
77. Ye, S.; Yin, Z.; Fu, Y.; Lin, H.; Pan, Z. A multi-granularity semisupervised active learning for point cloud semantic segmentation. *Neural Comput. Appl.* **2023**, *35*, 15629–15645. [[CrossRef](#)]
78. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach City, CA, USA, 4–9 December 2017.
79. Baker, S.; Matthews, I. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.* **2004**, *56*, 221–255. [[CrossRef](#)]
80. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016; pp. 1050–1059.
81. Depeweg, S.; Hernández-Lobato, J.M.; Doshi-Velez, F.; Udluft, S. Uncertainty decomposition in bayesian neural networks with latent variables. *arXiv* **2017**, arXiv:1706.08495.
82. Depeweg, S.; Hernandez-Lobato, J.M.; Doshi-Velez, F.; Udluft, S. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1184–1193.
83. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
84. Papon, J.; Abramov, A.; Schoeler, M.; Worgotter, F. Voxel cloud connectivity segmentation-supervoxels for point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2027–2034.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.