USIP: Unsupervised Stable Interest Point Detection from 3D Point Clouds

Jiaxin Li* Gim Hee Lee
Department of Computer Science, National University of Singapore

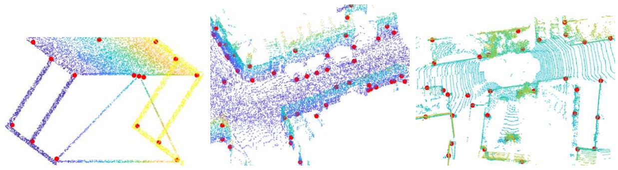
Abstract

In this paper, we propose the USIP detector: an Unsupervised Stable Interest Point detector that can detect highly repeatable and accurately localized keypoints from 3D point clouds under arbitrary transformations without the need for any ground truth training data. Our USIP detector consists of a feature proposal network that learns stable keypoints from input 3D point clouds and their respective transformed pairs from randomly generated transformations. We provide degeneracy analysis and suggest solutions to prevent it. We encourage high repeatability and accurate localization of the keypoints with a probabilistic chamfer loss that minimizes the distances between the detected keypoints from the training point cloud pairs. Extensive experimental results of repeatability tests on several simulated and real-world 3D point cloud datasets from Lidar, RGB-D and CAD models show that our USIP detector significantly outperforms existing hand-crafted and deep learning-based 3D keypoint detectors. Our code is available at the project website.[1]

1. Introduction

3D interest point or keypoint detection refers to the problem of finding stable points with well-defined positions that are highly repeatable on 3D point clouds under arbitrary SE(3) transformations. These detected keypoints play important roles in many computer vision and robotics tasks, where 3D point clouds are widely adopted as the data structure to represent objects and scenes in the 3D space. Examples include geometric registration for 3D object modeling [1] or point cloud-based SLAM [20], and 3D object [12, 16] or place recognition [30]. In these tasks, the detected keypoints are respectively used as correspondences to compute rigid transformations, and locations to extract representative signatures for efficient retrievals.

Despite the high number of successful hand-crafted detectors proposed for 2D images [22, 17, 11], significantly lesser hand-crafted detectors [28] with limited success are proposed for hand-crafted detectors on 3D point clouds. This difference can be largely attributed to the difficulty in hand-crafting powerful algorithms to extract meaningful information solely from the Euclidean coordinates of the point cloud in comparison to images that contain richer information from the additional RGB channels. The problem is further aggravated by the fact that it is difficult to hand-craft 3D detectors to handle 3D point clouds in arbitrary transformations, *i.e*., different reference coordinate frames.



(a) (b) (c)

Figure 1. Examples of keypoints detected by our USIP detector on four datasets: (a) ModelNet40 [31], object model. (b) Oxford RobotCar [18], outdoor SICK LiDAR. (c) KITTI [9] (Trained on Oxford), outdoor Velodyne LiDAR.

Very few deep learning-based 3D keypoint detectors exist (only one deep learning-based approach [32] exists to date) in contrast to its increasing success on learning 3D keypoint descriptors [6, 5, 34, 13]. This is due to the lack of ground truth training datasets to supervise deep learning-based detectors on 3D point clouds. Unlike 3D descriptors that are supervised by easily available ground truth registered overlapping 3D point clouds [6, 5, 13, 34, 32, 10], it is impossible for anyone to identify and label the "ground truth" keypoints on 3D point clouds.

We propose the USIP detector: an **U**nsupervised **S**table **I**nterest **P**oint deep learning-based detector that can detect highly repeatable, and accurately localized keypoints from 3D point clouds under arbitrary transformations *without* the need for any ground truth training data. To this end, we design a Feature Proposal Network (FPN) that outputs a set of keypoints and their respective saliency uncertainties from an input 3D point cloud. Our FPN improves keypoint localization by estimating their positions on contrary to existing 3D detectors [26, 32, 35] that select existing points in the point cloud as keypoints, which causes quantization errors. During training, we apply randomly generated SE(3) trans-

*Jiaxin Li now works at nuTonomy, an APTIV company.
[1] https://github.com/lijx10/USIP

formations on each point cloud to get a set of corresponding pairs of transformed point clouds as inputs to the FPN. Furthermore, we identify and prevent the degeneracy of our USIP detector. We encourage high repeatability and accurate localization of the keypoints with a probabilistic chamfer loss that minimizes the distances between the detected keypoints from the training point cloud pairs. Additionally, we introduce a point-to-point loss to enforce the constraint of getting keypoints that lie close to the point cloud. We verify our USIP detector by performing extensive experiments on several simulated and real-world 3D point cloud datasets from Lidar, RGB-D and CAD models. Some qualitative results are shown in Fig 1. Our key contributions:

- Our USIP detector is fully unsupervised, thus avoids the need for ground truth that are impossible to obtain.
- We provide degeneracy analysis of our USIP detector and suggest solutions to prevent it.
- Our FPN improves keypoint localization by estimating the keypoint position instead of choosing it from an existing point in the point cloud.
- We introduce the probabilistic chamfer loss and point-to-point loss to encourage high repeatability and accurate keypoint localization.
- The use of randomly generated transformations on point clouds during training inherently allows our network to achieve good performance under rotations.

2. Related Work

Unlike the recent success of deep learning-based 3D keypoint descriptors [6, 5, 13, 34, 32, 10], most existing 3D keypoint detectors remain hand-crafted. A comprehensive review and evaluation of existing hand-crafted 3D keypoint detectors can be found in [28]. Local Surface Patches (LSP) [3] and Shape Index (SI) [7] are based on the maximum and minimum principal curvatures of a point, and consider the point as a keypoint if it is a global extremum in a predefined neighborhood. Intrinsic Shape Signatures (ISS) [35] and KeyPoint Quality (KPG) [19] select salient points that has a local neighborhood with large variations along each principal axis. MeshDoG [33] and Salient Points (SP) [2] construct a scale-space of the curvature with the Difference-of-Gaussian (DoG) operator similar to SIFT [17]. Points with local extrema values over an one-ring neighborhood are selected as keypoints. Laplace-Beltrami Scale-sapce (LBSS) [29] computes the saliency by applying a Laplace-Beltrami operator on increasing supports for each point.

More recently, LORAX [8] proposes the method of projecting the point set into a depth map and use Principal Component Analysis (PCA) to select keypoints with commonly found geometric characteristics. All hand-crafted approaches share the common trait of relying on the local geometric properties of the points to select keypoints. Hence, the performances of these detectors deteriorate under disturbances such as noise, density variations and/or arbitrary transformations. To the best of our knowledge, the only existing deep learning-based 3D keypoint detector is the weakly supervised 3DFeatNet [32], which is trained with GPS/INS tagged point clouds. However, the training of 3Dfeat-Net is largely focused on learning discriminative descriptors using the Siamese architecture with an attention score map that estimates the saliency of each point as its by-product. It does not ensure good performance of the keypoint detection. In comparison, our USIP is designed to encourage high repeatability and accurate localization of the keypoints. Furthermore, our method is fully unsupervised and does not rely on any form of ground truth datasets.

3. Our USIP Detector

Fig. 2(a) shows the illustration of the pipeline to train our USIP detector. We denote a point cloud from the training dataset as $\mathbf{X} = [X_0, \cdots, X_N] \in \mathbb{R}^{3\times N}$. A set of transformation matrices $\{T_1, \cdots, T_L\}$, where $T_l \in \mathrm{SE}(3)$ is randomly generated and applied to the point cloud $\mathbf{X}$ to form L pairs of training inputs denoted as $\{\{\mathbf{X}, \tilde{\mathbf{X}}_1\}, \cdots, \{\mathbf{X}, \tilde{\mathbf{X}}_L\}\}$, where $\tilde{\mathbf{X}}_l = T_l \circ \mathbf{X} \in \mathbb{R}^{3\times N}$. Here, we use the operator $\circ$ to denote matrix multiplication under homogeneous coordinate with a slight abuse of notation. We drop the indices l for brevity and refer to a triplet of training pair of point clouds and their corresponding transformation matrix as $\{\mathbf{X}, \tilde{\mathbf{X}}, T\}$. During training, $\mathbf{X}$ and $\tilde{\mathbf{X}}$ are respectively fed into the FPN, which outputs M proposal keypoints and its saliency uncertainties denoted as $\{\mathbf{Q} = [Q_1, \cdots, Q_M], \Sigma = [\sigma_1, \cdots, \sigma_M]^T\}$ and $\{\tilde{\mathbf{Q}} = [\tilde{Q}_1, \cdots, \tilde{Q}_M], \tilde{\Sigma} = [\tilde{\sigma}_1, \cdots, \tilde{\sigma}_M]^T\}$ for the respective point cloud. $Q_m \in \mathbb{R}^3$, $\tilde{Q}_m \in \mathbb{R}^3$, $\sigma_m \in \mathbb{R}^+$ and $\tilde{\sigma}_m \in \mathbb{R}^+$. We enforce $\sigma_m \in \mathbb{R}^+$ and $\tilde{\sigma}_m \in \mathbb{R}^+$ so that it is a valid rate parameter in our probabilistic chamfer loss (see later paragraph). To improve keypoint localization, it is not necessary for all $Q_m \in \mathbf{Q}$ to be any of the points in $\mathbf{X}$. Similar condition applies to all $\tilde{Q}_m \in \tilde{\mathbf{Q}}$.

We undo the transformation on $\tilde{\mathbf{Q}}$ with a slight abuse of notation to get $\mathbf{Q}' = T^{-1} \circ \tilde{\mathbf{Q}} \in \mathbb{R}^{3\times M}$, so that $\mathbf{Q}'$ can be compared directly to $\mathbf{Q}$. Here, we made an assumption that the saliency uncertainties remain unaffected after the transformation, *i.e.*, $\Sigma' = \tilde{\Sigma}$. The objectives of detecting keypoints that are highly repeatable and accurately localized from 3D point clouds under arbitrary transformations can now be achieved by formulating a loss function that minimizes the difference between $\mathbf{Q}$ and $\mathbf{Q}'$. To this end, we propose the loss function: $\mathcal{L} = \mathcal{L}_c + \lambda\mathcal{L}_p$, where $\mathcal{L}_c$ is the probabilistic chamfer loss that minimizes the probabilistic distances between all correspondence pairs of keypoints in $\mathbf{Q}$ and $\mathbf{Q}'$. $\mathcal{L}_p$ is the point-to-point loss that minimizes the

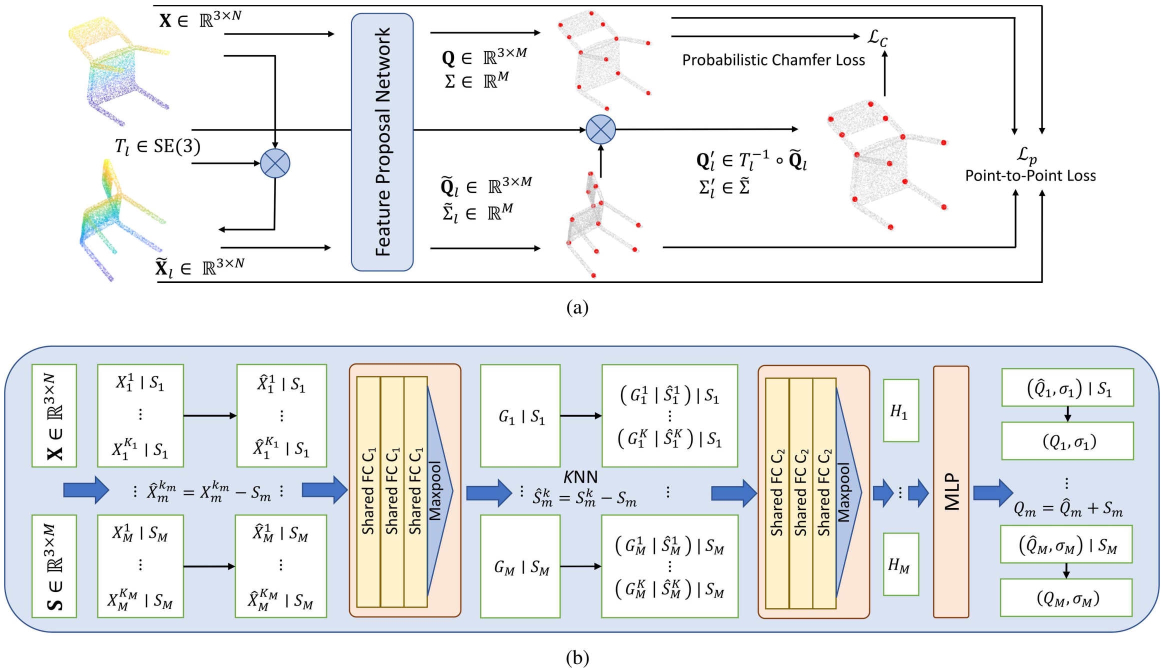


Figure 2. (a) The training pipeline of USIP detector. (b) The architecture of our Feature Proposal Network (FPN). See text for more detail.

distance of the estimated keypoints to their respective nearest neighbor in the point cloud. λ is a hyperparameter that adjust the relative contribution of $\mathcal{L}_c$ and $\mathcal{L}_p$ to the total loss.

Probabilistic Chamfer Loss A simple way to minimize the distance between $\mathbf{Q}$ and $\mathbf{Q}'$ is to use the chamfer loss:

$$\sum_{i=1}^{M} \min_{Q'_j \in \mathbf{Q}'} \|Q_i - Q'_j\|_2^2 + \sum_{j=1}^{M} \min_{Q_i \in \mathbf{Q}} \|Q_i - Q'_j\|_2^2, \tag{1}$$

that minimizes the distance of each point in one point cloud with its nearest neighbor in the other point cloud. However, the M proposals are not equally salient. The receptive field of a point Q_i can be a featureless surface since the receptive field is limited to a small volume. In this case, it is detrimental to force the FPN to minimize the distance between Q_i and Q'_j, where Q'_j is the nearest neighbor of Q_i in $\mathbf{Q}'$.

To mitigate the above problem, we design our FPN to learn the saliency uncertainties Σ and Σ' of the proposal keypoints $\mathbf{Q}$ and $\mathbf{Q}'$ with a probabilistic chamfer loss $\mathcal{L}_c$. In particular, we propose to formulate $\mathcal{L}_c$ with an exponential distribution that measures the probabilistic distances between $\mathbf{Q}$ and $\mathbf{Q}'$ with the saliency uncertainties Σ and Σ'. More formally, the probability distribution between Q_i and Q'_j for $i = 1, \cdots, M$ is given by:

$$p(d_{ij} \mid \sigma_{ij}) = \frac{1}{\sigma_{ij}} \exp\left(-\frac{d_{ij}}{\sigma_{ij}}\right), \qquad \text{where}$$
$$\sigma_{ij} = \frac{\sigma_i + \sigma'_j}{2} > 0, \quad d_{ij} = \min_{Q'_j \in \mathbf{Q}'} \|Q_i - Q'_j\|_2 \geq 0. \tag{2}$$

$p(d_{ij} \mid \sigma_{ij})$ is a valid probability distribution since it integrates to 1. A shorter distance d_{ij} between the proposal keypoints Q_i and Q'_j gives a higher probability that Q_i and Q'_j are highly repeatable and accurately localized keypoints in the point clouds $\mathbf{X}$ and $\tilde{\mathbf{X}}$. Assuming i.i.d for all $d_{ij} \in D_{ij}$, the joint distribution between $\mathbf{Q}$ and $\mathbf{Q}'$ is given by:

$$p(D_{ij} \mid \Sigma_{ij}) = \prod_{i=1}^{M} p(d_{ij} \mid \sigma_{ij}). \tag{3}$$

It is important to note that the probability distribution is not symmetrical when the order of the point cloud is swapped, *i.e.*, $\mathbf{Q}'$ and $\mathbf{Q}$, due to a different set of nearest neighbors, *i.e.*, $d_{ij} \neq d_{ji}$ and $\sigma_{ij} \neq \sigma_{ji}$. Hence, the joint distribution between $\mathbf{Q}'$ and $\mathbf{Q}$ is given by:

$$p(D_{ji} \mid \Sigma_{ji}) = \prod_{j=1}^{M} p(d_{ji} \mid \sigma_{ji}), \qquad \text{where}$$
$$\sigma_{ji} = \frac{\sigma'_j + \sigma_i}{2} > 0, \quad d_{ji} = \min_{Q_i \in \mathbf{Q}} \|Q_i - Q'_j\|_2 \geq 0. \tag{4}$$

Finally, the probabilistic chamfer loss $\mathcal{L}_c$ between $\mathbf{Q}'$ and $\mathbf{Q}$ is given by the negative log-likelihood of the joint distributions defined in Eq. 3 and 4:

$$\begin{aligned}\mathcal{L}_c &= \sum_{i=1}^{M} -\ln p(d_{ij} \mid \sigma_{ij}) + \sum_{j=1}^{M} -\ln p(d_{ji} \mid \sigma_{ji}) \\ &= \sum_{i=1}^{M} \left(\ln \sigma_{ij} + \frac{d_{ij}}{\sigma_{ij}}\right) + \sum_{j=1}^{M} \left(\ln \sigma_{ji} + \frac{d_{ji}}{\sigma_{ji}}\right). \end{aligned} \tag{5}$$

We analyze the physical meaning of σ_{ij} or σ_{ji} by computing the extrema of Eq. 2 from its first derivative over σ_{ij}:

$$\frac{\partial p(d_{ij} \mid \sigma_{ij})}{\partial \sigma_{ij}} = \frac{d_{ij}\exp(-d_{ij}/\sigma_{ij})}{\sigma_{ij}^3} - \frac{\exp(-d_{ij}/\sigma_{ij})}{\sigma_{ij}^2}, \tag{6}$$

and solve for the stationary points:

$$\frac{\partial p(d_{ij} \mid \sigma_{ij})}{\partial \sigma_{ij}} = 0 \Rightarrow \sigma_{ij} = d_{ij}. \tag{7}$$

Furthermore, the second derivative $p''(d_{ij} \mid \sigma_{ij})|_{\sigma_{ij}=d_{ij}} < 0$ means that given a fixed $d_{ij} \neq 0$, the highest probability $p(d_{ij} \mid \sigma_{ij})$ is achieved at $\sigma_{ij} = d_{ij}$. Consider any triplet of proposal keypoints $\{Q_i, Q'_j, Q'_k\}$, where d_{ij} and d_{ki} are the distances between the nearest neighbors $\{Q_i, Q'_j\}$ and $\{Q'_k, Q_i\}$ (Q_i can be the nearest neighbor in both orders of $\mathbf{Q}$ and $\mathbf{Q}'$ since chamfer distance is not bijective). σ'_k has to take a large value when $d_{ij} \to 0$ and d_{kj} is large because we have shown that $\sigma_{ij} = d_{ij}$ and $\sigma_{ki} = d_{ki}$ at optimum. Furthermore, $d_{ij} \to 0$ and d_{kj} is large implies that $\{Q_i, Q'_j\}$ are repeatable and accurately localized keypoints while Q'_k is not. Hence, a large saliency uncertainty σ'_k for a bad proposal keypoint Q'_k at optimum shows that our probabilistic chamfer loss is guiding the FPN to learn correctly.

Point-to-Point Loss To avoid quantization error in the positions of the keypoints, we design the FPN such that it is not necessary that the proposal keypoints $\mathbf{Q}$ to be any of the points in $\mathbf{X}$. However, this can cause the FPN to give erroneous proposal keypoints $\mathbf{Q}$ that are far away from the point cloud $\mathbf{X}$. We circumvent this problem by adding a loss function $\mathcal{L}_p$ that penalizes $Q_m \in \mathbf{Q}$ for being too far from $\mathbf{X}$. We also apply similar penalty on $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{X}}$. This loss can be formulated as either the point-to-point loss [1]:

$$\mathcal{L}_{\text{point}} = \sum_{i=1}^{M} \min_{X_j \in \mathbf{X}} \|Q_i - X_j\|_2^2 + \sum_{i=1}^{M} \min_{\tilde{X}_j \in \tilde{\mathbf{X}}} \|\tilde{Q}_i - \tilde{X}_j\|_2^2, \tag{8}$$

where $X_j \in \mathbf{X}$ is the nearest neighbor of Q_i or the point-to-plane loss [23, 4]:

$$\mathcal{L}_{\text{plane}} = \sum_{i=1}^{M} \mathcal{N}_j^T (Q_i - X_j) + \sum_{i=1}^{M} \tilde{\mathcal{N}}_j^T (\tilde{Q}_i - \tilde{X}_j), \tag{9}$$

where $\mathcal{N}_j$ and $\tilde{\mathcal{N}}_j$ are the nearest surface normal in $\mathbf{X}$ to Q_i and $\tilde{\mathbf{X}}$ to $\tilde{Q}_i$, respectively. We set $\mathcal{L}_p = \mathcal{L}_{\text{point}}$ by default since we found experimentally that both loss functions give similar performances.

4. Feature Proposal Network

The network architecture of our FPN is shown in Fig. 2(b). We first sample M nodes denoted as $\mathbf{S} = [S_1, \cdots, S_M] \in \mathbb{R}^{3\times M}$ with Farthest Point Sampling (FPS) from a given input point cloud $\mathbf{X} \in \mathbb{R}^{3\times N}$. A neighborhood of points is built for each node $S_m \in \mathbf{S}$ using point-to-node grouping [15, 14], which is denoted as $\{\{X_1^1|S_1, ..., X_1^{K_1}|S_1\}, \cdots, \{X_M^1|S_M, ..., X_M^{K_M}|S_M\}\}$. $K_1, \cdots, K_M$ represents the number of points associated with the each of the nodes in $\mathbf{S}$. The advantage of point-to-node association over node-to-point kNN search or radius-based ball-search is two-fold: (1) Every point in $\mathbf{X}$ is associated with one node, while some points may be left out in node-to-point kNN search and ball-search. (2) Point-to-node grouping automatically adapts to various scale and point density, while kNN search and ball-search are vulnerable to density variation and varying scales, respectively. To make FPN *translation equivariant*, we normalize each neighborhood point $\{X_m^1|S_m, \cdots, X_m^{K_m}|S_m\}$ into $\{\hat{X}_m^1|S_m, \cdots, \hat{X}_m^{K_m}|S_m\}$ by subtracting from its respective node S_m, *i.e.*, $\hat{X}_m^{k_m} = X_m^{k_m} - S_m$. Each cluster of normalized local neighborhood points is then fed into a PointNet-like network [21] shown in Fig. 2(b) to get a local feature vector G_m associated with S_m. A kNN grouping layer is applied on the set of local feature vectors $\{G_1|S_1, \cdots, G_M|S_M\}$ to achieve hierarchical information aggregation. Specifically, the k nearest neighbors of each pair of $(G_m|S_m)$ are retrieved as $\{(G_m^1|S_m^1)|S_m, \cdots, (G_m^K|S_m^K)|S_m\}$. These kNN local feature vectors are then normalized by subtracting with its respective S_m to get a position-independent neighborhood denoted as $\{(G_m^1|\hat{S}_m^K)|S_m, \cdots, (G_m^K|\hat{S}_m^K)|S_m\}$, where $\hat{S}_m^K = S_m^K - S_m$, before feeding into another network to get a set of feature vectors $\{H_1, \cdots, H_M\}$. A simple Multi-Layer Perceptron (MLP) is then used to estimate M proposal keypoints $\{\hat{Q}_1|S_1, \cdots, \hat{Q}_M|S_M\}$, where $\hat{Q}_m \in \mathbb{R}^3$, and saliency uncertainties $\{\sigma_1, \cdots, \sigma_M\}$, where $\sigma_m \in \mathbb{R}^+$ from $\{H_1, \cdots, H_M\}$. Finally, we un-normalize each $\hat{Q}_m$ with S_m, *i.e.*, $Q_m = \hat{Q}_m + S_m$ to get the final proposal keypoints $\{Q_1, \cdots, Q_M\}$. It is important to note that the size of the receptive field is controlled by the number of proposals M and K in kNN layers and it determines the level-of-detail for each feature. Large receptive field leads to features that are salient on a large-scale and vice versa.

5. Degeneracy Analysis

Let us denote the FPN as $f(\mathbf{Y}) : \mathbf{Y} \to \mathbb{R}^{3\times M}$, where $\mathbf{Y} = [Y_1, \cdots, Y_N] \in \mathbb{R}^{3\times N}$ is the input of the network. We further denote a transformation matrix $T \in \text{SE}(3)$, where $R \in \text{SO}(3)$ and $t \in \mathbb{R}^3$ are the rotation matrix and translation vector in T. We get $\mathbf{Y}' = R\mathbf{Y} \oplus t$, where $\oplus$ is the operator to denote the addition of t to every 3×1 entries of the other term. We say that the network is degenerate when it outputs ***trivial solutions*** where $f(\mathbf{Y}') \equiv Rf(\mathbf{Y}) \oplus t$ is satisfied for all R and t.

Lemma 1. $f(\mathbf{Y}') \equiv Rf(\mathbf{Y}) \oplus t$ *when* $f(.)$ *outputs the* ***centroid*** *of the input point cloud,* i.e., $f(\mathbf{Y}) = \frac{1}{N}\sum_n Y_n$ *and* $f(\mathbf{Y}') = \frac{1}{N}\sum_n Y'_n$.

Proof. Putting $Y'_n = RY_n + t$ into $f(\mathbf{Y}') = \frac{1}{N}\sum_n Y'_n$, we get $f(\mathbf{Y}') = \frac{1}{N}\sum_n (RY_n + t) = R(\frac{1}{N}\sum_n Y_n) + t = Rf(\mathbf{Y}) \oplus t$. Hence, $f(Y') \equiv Rf(Y) \oplus t$ which completes our proof that the network degenerates when it outputs the centroid of the input point cloud. $\square$

Lemma 2. $f(\mathbf{Y}') \equiv Rf(\mathbf{Y}) \oplus t$ *when* $f(.)$ *is translational equivariant,* i.e., $f(\cdot) \oplus t = f(\cdot \oplus t)$, *and outputs points that are in the linear subspace of any* ***principal axis*** *from the input point cloud denoted as* $\mathbf{U} = [U_1, U_2, U_3] \in \mathbb{R}^{3\times 3}$, i.e., $f(\mathbf{Y}) = [c_1 U_i^T, \cdots, c_M U_i^T]^T$ *and*

$$\begin{aligned} f(\mathbf{Y}') &= f(R\mathbf{Y} \oplus t) \\ &= f(R\mathbf{Y}) \oplus t \qquad \textit{(translation equivariance)} \\ &= [c_1 U_i'^T, \cdots, c_M U_i'^T]^T \oplus t, \end{aligned} \tag{10}$$

where U_i *can be any principal axis in* $\mathbf{U}$ *and* $c_1, \cdots, c_M$ *are scalar coefficients in* $\mathbb{R}$.

Proof. Let $V = \frac{1}{N}\sum_n (Y_n - \bar{Y})(Y_n - \bar{Y})^T$ and $V' = \frac{1}{N}\sum_n (Y'_n - \bar{Y}')(Y'_n - \bar{Y}')^T$ denote the covariance matrices of $\mathbf{Y}$ and $\mathbf{Y}'$, respectively. $\bar{Y} = \frac{1}{N}\sum_n Y_n$ and $\bar{Y}' = \frac{1}{N}\sum_n Y'_n$ are the centroids of $\mathbf{Y}$ and $\mathbf{Y}'$, respectively. Putting $Y'_n = RY_n + t$ into $\bar{Y}'$ and V', we get:

$$V' = R\frac{1}{N}\sum_n (Y_n - \bar{Y})(Y_n - \bar{Y})^T R^T = RVR^T. \tag{11}$$

Taking the Singular Value Decomposition (SVD) of V and V', we get $V = \mathbf{U}\mathbf{D}\mathbf{U}^T$ and $V' = \mathbf{U}'\mathbf{D}'\mathbf{U}'^T$, where $\mathbf{D}$ and $\mathbf{D}'$ are the 3×3 diagonal matrices of singular values, and $\mathbf{U}$ and $\mathbf{U}'$ are the 3×3 Eigenvectors that are also the principal axes of $\mathbf{Y}$ and $\mathbf{Y}'$, respectively. Putting the SVD of V and V' into Eq. 11, we get:

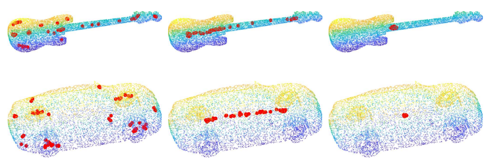
$$\begin{aligned} V' &= RVR^T = R\mathbf{U}\mathbf{D}\mathbf{U}^T R^T = (R\mathbf{U})\mathbf{D}(R\mathbf{U})^T \\ &\equiv \mathbf{U}'\mathbf{D}'\mathbf{U}'^T \quad \Rightarrow \mathbf{U}' = R\mathbf{U}. \end{aligned} \tag{12}$$

Putting the relationship from Eq. 12 into $f(\mathbf{Y}') = [c_1 U_i'^T, \cdots, c_M U_i'^T]^T \oplus t$, we get:

$$f(\mathbf{Y}') = R[c_1 U_i^T, \cdots, c_M U_i^T]^T \oplus t \equiv Rf(\mathbf{Y}) \oplus t, \tag{13}$$

which completes our proof that the network degenerates when it outputs a set of points on any principal axis. $\square$

Discussions We note that the network requires sufficient global semantic information of the input point cloud, *e.g.*, the input is the whole point cloud or clusters of local neighbor points that contain large receptive fields, to learn the trivial solutions of centroid or set of points on the principal axes. Hence, the degeneracies can be easily prevented by limiting the receptive fields of the FPN. We achieve this by setting the the number of clusters M and K nearest neighbors of the clusters in the FPN (refer to Sec. 4 for the definitions of M and K) to reasonable values. Small values for M or high values for K increases the receptive field and causes the FPN to degenerate. Fig. 3 show some examples of the degeneracies with different K values at $M = 64$. It is interesting to note that the principal axis degeneracy occurs when K is set to a mid-range value, and centroid degeneracy occurs when K is set to a high value. This implies that larger receptive fields, *i.e.*, a higher global semantic information is needed for the network to learn the centroid. We also notice experimentally that the degeneracies (both centroid and principal axis) occur in point clouds with more regular shapes, *e.g.* objects from ModelNet40 where the centroid and principal axes are more well-defined.



(a) (b) (c)

Figure 3. Increasing K values in FPN causes degeneracies ($M = 64$). (a) No degeneracy with $K = 9$ (low value). (b) Principal axis degeneracy with $K = 24$ (mid-range value). (c) Centroid degeneracy with $K = 64$ (high value).

6. Experiments

Following [28], we evaluate the *repeatability* (Sec. 6.1), *distinctiveness* (Sec. 6.2) and *computational efficiency* (Sec. 6.4) of our USIP detector on 4 datasets in Tab. 1.

Implementation Details Three USIP detectors are respectively trained for outdoor Lidars, RGB-D scans and object models. Specifically, we use the Oxford [18] for outdoor Lidar, "RGB-D reconstruction dataset" [34] for RGB-D, and ModelNet40 [31] for object models. The PCL [26] implementations of the classical detectors, *i.e.*, ISS [35], Harris-3D [11] and SIFT-3D [17] are used for the comparisons. We take the pretrained models of 3DFeat-Net [32] for KITTI [9] and Oxford, and train separate models for Redwood and ModelNet40 using its open-sourced codes.

Qualitative Visualization Fig. 7 shows some results from our USIP detector on ModelNet40. Our USIP learns keypoints on corners, edges, center of small surfaces, etc. Keypoints in the first row of Fig. 7 are selected with Non-Maximum Suppression (NMS) and thresholding on the saliency uncertainty σ. In the second row, keypoints are selected with only NMS. Keypoints with small σ are shown in bright red and get darker with larger σ.

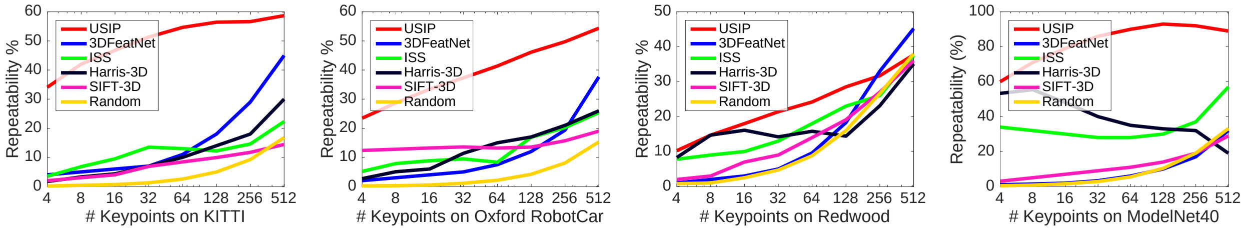


Figure 4. Relative repeatability when different number of keypoints are detected. Left to right: KITTI, Oxford, Redwood, ModelNet40.

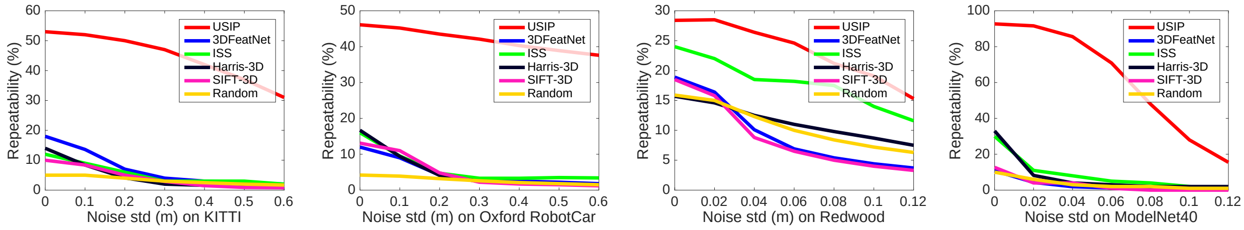


Figure 5. Relative repeatability when Gaussian noise $\mathcal{N}(0, \sigma_{noise})$ is added to the input point clouds. Keypoint number is fixed to 128.

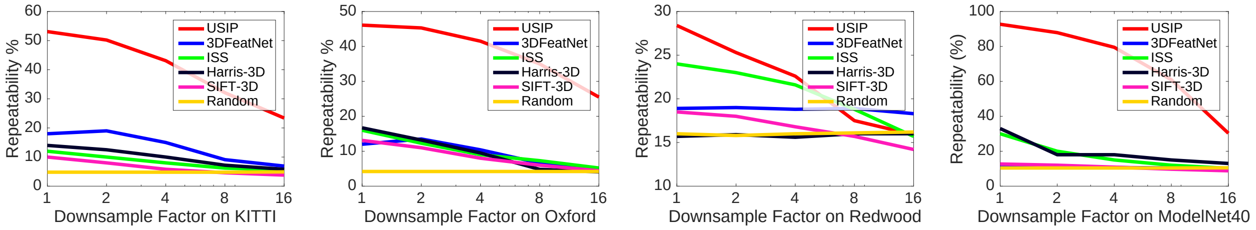


Figure 6. Relative repeatability when the input point cloud is randomly downsampled by some factors. Keypoint number is fixed to 128.

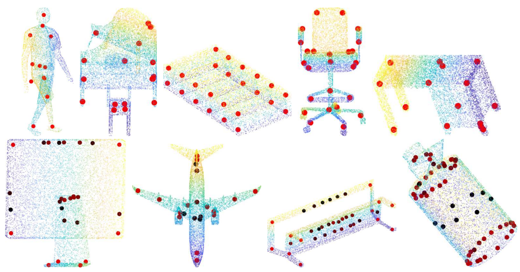


Figure 7. Examples of keypoints from our USIP on ModelNet40.

	KITTI	Oxford	Redwood	ModelNet40
Type	Velodyne lidar	SICK lidar	RGB-D	CAD Model
Scale (diameter)	200m	60m	10m	2
# point	16,384	16,384	10,240	5,000
ϵ in Eq. 14	0.5m	0.5m	0.1m	0.03
Rotation	2D	2D	3D	3D
Noise	Sensor	Sensor	Gaussian	Gaussian
Occlusion	Yes	Yes	Yes	No
Density Variation	Yes	No	No	No
Missing Parts	Yes	Yes	Yes	No

Table 1. Datasets used in evaluating keypoint repeatability.

6.1. Repeatability

Repeatibility refers to the ability of a detector to detect keypoints in the same locations under various disturbances such as view-point variations, noise, missing parts, etc. It is often taken as the most important measure of keypoint detectors because it is a standalone measure that depends only on the detector (without a descriptor). Given two point clouds $\{\mathbf{X}, \tilde{\mathbf{X}}\}$ of a scene captured from different viewpoints such that $\{\mathbf{X}, \tilde{\mathbf{X}}\}$ are related by a rotation matrix $R \in \text{SO}(3)$ and a translational vector $t \in \mathbb{R}^3$. A keypoint detector detects a set of keypoints $\mathbf{Q} = [Q_1, \cdots, Q_M]$ and $\tilde{\mathbf{Q}} = [\tilde{Q}_1, \cdots, \tilde{Q}_M]$ from $\{\mathbf{X}, \tilde{\mathbf{X}}\}$, respectively. A keypoint $Q_i \in \mathbf{Q}$ is repeatable if the distance between $RQ_i + t$ and its nearest neighbor $\tilde{Q}_j \in \tilde{\mathbf{Q}}$ is less than a threshold ϵ, *i.e.*,

$$\|RQ_i + t - \tilde{Q}_j\|_2 < \epsilon. \tag{14}$$

Test Datasets We evaluate repeatability on KITTI, Oxford, Redwood and ModelNet40. Note that our USIP is not trained on KITTI nor Redwood. The KITTI and Oxford test datasets are prepared by 3DFeat-Net [32]. Each pair of point clouds $\{\mathbf{X}, \tilde{\mathbf{X}}\}$ are captured from nearby locations of within 10m and manually augmented with random 2D rotations. $\{\mathbf{X}, \tilde{\mathbf{X}}\}$ in Redwood are from simulated RGB-D cameras with 3D rotations / translations and Gaussian noise. The overlap between $\{\mathbf{X}, \tilde{\mathbf{X}}\}$ is as low as 30%. In ModelNet40, $\tilde{\mathbf{X}}$ is obtained by augmenting $\mathbf{X}$ with random 3D rotations. Details of the datasets are shown in Tab. 1.

Relative Repeatability We use relative repeatability that normalizes over the total number of detected keypoints $|\mathbf{Q}|$ for fair comparisons, *i.e.*, repeatability $= |\mathbf{Q}_{\text{rep}}|/|\mathbf{Q}|$, where

$\mathbf{Q}_{\text{rep}}$ is the number of keypoints that passed the repeatability test in Eq. 14. We set the parameters of each keypoint detector in each dataset to generate 4, 8, 16, 32, 64, 128, 256 and 512 keypoints or close to these numbers when it is not possible to set the detectors (SIFT-3D, Harris-3D and ISS) to generate exact number of keypoints. Note that in general the repeatability should be proportional to the number of keypoints. In the extreme case that $\mathbf{Q} = \mathbf{X}$, *i.e.*, each point is regarded as a keypoint, the repeatability is the same as the percentage of overlap between $\{\mathbf{X}, \tilde{\mathbf{X}}\}$. As shown in Fig. 4, our USIP outperforms other detectors by a significant margin on the 4 datasets over 8 different # of keypoints.

Robustness to Noise The original points in KITTI and Oxford are already corrupted with sensor noise. We further augment the point clouds in the 4 datasets with Gaussian noise $\mathcal{N}(0, \sigma_{noise})$, where σ_{noise} is up to 0.6m for KITTI and Oxford, 0.12m for Redwood and 0.12 (no unit) for ModelNet40. The number of keypoints is fixed to 128. Our USIP is a lot more robust than other detectors as shown in Fig. 5. In KITTI and Oxford, the performances of other detectors fall to the level of random sampling when $\sigma_{noise} \geq$ 0.2m, while our USIP does not show significant drop in performance even with $\sigma_{noise} \geq 0.6m$. In Redwood, other methods except USIP and ISS deteriorate to random sampling with $\sigma_{noise} \geq$ 0.02m. In ModelNet40, our method maintain high repeatability of 91% with $\sigma_{noise} = 0.02$, while all other methods drop below 8%.

Robustness to Downsampling We evaluate the repeatability of the detectors on input point clouds downsampled by some factors using random selection. The results are shown in Fig. 6, where the down-sample factor denoted as α means the number of points is reduced to $\frac{1}{\alpha}$ of the original number shown in Tab. 1. We can see that the repeatability of our USIP remains satisfactory even with a 16× downsampling on KITTI, Oxford and ModelNet40. The only exception is the Redwood dataset, where almost all detectors perform poorly on high downsample factors.

6.2. Distinctiveness: Point Cloud Registration

Distinctiveness is a measure of the performance of keypoint detectors and descriptors for finding correspondences in point cloud registration. Hence, distinctiveness is not as good as repeatability as an evaluation criterion on keypoint detectors because it is confounded with the performance of the descriptor. We mitigate this limitation by evaluating point cloud registration over several existing keypoint descriptors. We also use the results to show that our USIP detector works with different existing keypoint descriptors.

Experiment Setup We follow the point cloud registration pipeline from 3DFeat-Net [32] on their KITTI test dataset. Four descriptors are used to perform keypoint description, *i.e.*, three off-the-shelf descriptors: 3DFeatNet, FPFH [25], SHOT[27], and our own descriptor inspired by 3DFeat-Net with minor modifications, which is denoted as "Our Desc." (details are in our supplementary material). Registration of a pair of point clouds involves 4 steps: (a) Extract keypoints and their corresponding descriptor vectors from each point cloud. (b) Establish keypoint-to-keypoint correspondences by nearest neighbor search of the descriptor vectors. (c) Perform RANSAC on the two matched keypoint sets to find the rotation and translation that have the most inliers. (d) Compare the resulted rotation and translation with the ground truth. A pair of point cloud is regarded as successfully registered if Relative Translational Error (RTE) $<$ 2m, and Relative Rotation Error (RRE) $< 5°$.

Registration Results We perform registration evaluations over the combination of 6 keypoint detectors and 4 descriptors. The registration failure rate and keypoint inlier ratio are shown in Tab. 2. Compared to other detectors, our USIP achieves the lowest registration failure rate and the highest inlier ratio with a considerable margin on all the 4 descriptors. The significance of the results in Tab. 2 is two fold. First, our USIP works well with various handcrafted and deep learning-based descriptors. Second, our USIP produces more distinctive keypoints since it consistently outperforms other keypoint detectors over different descriptors. The experimental configurations in Tab. 2 is not the optimal setting for our USIP detector and descriptor nor the 3DFeatNet because we have to fix the number of keypoints for fair comparison. In Tab. 3, we illustrate the best registration results for our USIP and 3DFeatNet on KITTI without limitation on the number of keypoints. In addition, we show the visualization of keypoint matching results of two examples from KITTI and Oxford in Fig. 8.

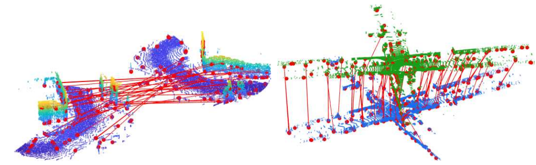


Figure 8. Keypoints and matches from our USIP detector and "Our Desc.". Best view with color and zoom-in.

6.3. Ablation Study

Point-to-node grouping vs. kNN / ball grouping Point-to-node grouping ensures the use of every point in the point cloud without cumbersome tuning of any hyperparameter since it associates each point with its nearest node, *i.e.*, one of the M points sampled from Farthest Point Sampling (FPS). Hence, no information is lost. In contrast, kNN and ball-search groupings do not guarantee this due to the sensitivity of the hyperparameter settings (#NN k and radius r for kNN and ball-search, respectively). Tab. 4 shows experimentally a drop in performance on the KITTI dataset with kNN and ball-search groupings. We further note that kNN

	Registration Failure Rate (%)				Inlier Ratio (%)			
	Our Desc.	3DFeatNet[32]	FPFH[24]	SHOT[27]	Our Desc.	3DFeatNet	FPFH	SHOT
Random	18.83	42.14	49.95	68.39	7.47	4.48	5.45	4.46
SIFT-3D[26, 17]	15.44	42.63	79.72	84.49	7.36	5.47	4.24	4.11
ISS[26, 35]	5.97	25.96	37.09	69.83	8.52	4.71	4.44	3.45
Harris-3D[26, 11]	3.81	13.56	49.49	51.29	10.57	6.58	4.78	5.00
3DFeatNet[32]	2.61	2.26	12.15	11.76	15.66	10.76	9.55	8.46
USIP	**1.41**	**1.55**	**8.37**	**5.40**	**32.20**	**22.48**	**18.77**	**18.21**

Table 2. Point cloud registration results on KITTI. The number of keypoints is fixed to 256.

Detector	Descriptor	Fail(%)	Inlier(%)	RTE(m)	RRE (°)
3DFeat-Net	3DFeat-Net	0.57	12.9	0.26 ± 0.26	0.56 ± 0.46
USIP	Our Desc.	**0.24**	**28.0**	$\mathbf{0.21 \pm 0.24}$	$\mathbf{0.42 \pm 0.32}$

Table 3. Point cloud registration on KITTI with optimal settings.

is used in the subsequent layers since the grouping is centered on each of the M sampled points from FPS, *i.e.*, it is now impossible for any points to be discarded.

M=512, # keypoint=128	point-to-node	kNN, k=64	Ball, r=2m
Repeatability (%)	53.6	46.9	43.8

Table 4. Keypoint repeatability with various grouping methods.

Probabilistic Chamfer loss vs. normal Chamfer loss Fig. 9 shows the results from the network with our probablistic Chamfer loss vs normal Chamfer loss on the KITTI and ModelNet40 datasets, respectively. Our probabilistic Chamfer loss clearly outperforms the normal Chamfer loss on both datasets. Note that Non-Maximum Suppression (NMS) is not used in normal Chamfer loss since it does not give the keypoint uncertainty σ required for thresholding.

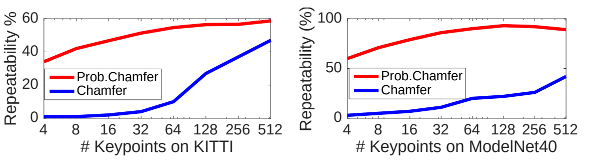


Figure 9. Relative repeatability. Left: KITTI. Right: ModelNet40.

Effect of point-to-point loss As shown by the example in Fig. 10, the point-to-point loss is needed to constrain the keypoints close to the input point cloud since the FPN does not require any keypoint to be in the input point cloud.

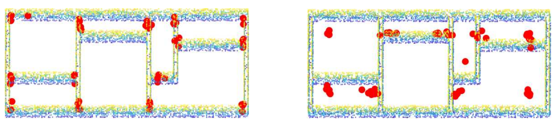


Figure 10. Visualization of USIP keypoints with point-to-point loss enabled, *i.e.*, $\lambda = 1$ (left) and disabled, *i.e.*, $\lambda = 0$ (right). Keypoints are closer to the point cloud with point-to-point loss.

6.4. Computational Efficiency

Hand-crafted detectors are deployed with single thread C++ codes on an Intel i7 6950X CPU. Our USIP and 3DFeatNet are deployed on a Nvidia 1080Ti, with PyTorch and TensorFlow, respectively. Computational efficiency is evaluated with 2,391 KITTI point clouds. **Scalability over number of keypoints.** Tab. 5 shows the time needed to compute the saliency of $M = \{128, 256, 512, 1024\}$ keypoints from a KITTI frame of 16,384 input points. We see that there is no substantial increase in the computational time. **Scalability over number of input points.** The computational times of all other 3D detectors increase with increasing input points since saliency is computed for every point in the input point cloud. In contrast, USIP requires lower computational time by directly computing saliency for M keypoints. Tab. 6 shows the time taken to compute 256 keypoints from input point clouds of increasing size with different methods. The computational times of other methods increase substantially, while USIP remains low.

# of Keypoints	128	256	512	1024
Average Time (Seconds)	0.004	0.007	0.011	0.028

Table 5. Average time for USIP to extract keypoints.

Input Point #	4096	8192	16,384	32,768	65,536
Random	0.0001	0.0003	0.0005	0.0013	0.0025
SIFT-3D	0.07	0.11	0.16	0.175	0.18
ISS	0.04	0.11	0.39	1.45	6.15
Harris-3D	0.03	0.06	0.15	0.38	1.12
3DFeatNet	0.05	0.14	0.44	1.45	5.34
USIP	0.005	0.007	0.011	0.023	0.052

Table 6. Average time (seconds) taken to compute 256 keypoints from input point clouds of increasing size with different methods.

7. Conclusion

In this paper, we present the USIP detector, an unsupervised deep learning-based keypoint detector for 3D point clouds. A probabilistic chamfer loss is proposed to guide the network to learn highly repeatable keypoints. We provide mathematical analysis and solutions for network degeneracy, which are supported by experimental results. Extensive evaluations are performed with Lidar scans, RGB-D images and CAD models. Our USIP detector out-performs existing detectors by a significant margin in terms of repeatability, distinctiveness and computational efficiency.

Acknowledgment. This work is supported in part by a Singapore MOE Tier 1 grant R-252-000-A65-114.

References

[1] 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 586–607. International Society for Optics and Photonics, 1992. 1, 4

[2] Umberto Castellani, Marco Cristani, Simone Fantoni, and Vittorio Murino. Sparse points matching by combining 3d mesh saliency with statistical descriptors. In *Computer Graphics Forum*, volume 27, pages 643–652. Wiley Online Library, 2008. 2

[3] Hui Chen and Bir Bhanu. 3d free-form object recognition in range images using local surface patches. *Pattern Recognition Letters*, 28(10):1252–1262, 2007. 2

[4] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 4

[5] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. *arXiv preprint arXiv:1808.10322*, 2018. 1, 2

[6] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. *Computer Vision and Pattern Recognition (CVPR). IEEE*, 1, 2018. 1, 2

[7] Chitra Dorai and Anil K. Jain. Cosmos-a representation scheme for 3d free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, 1997. 2

[8] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2472–2481. IEEE, 2017. 2

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 5

[10] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. *arXiv preprint arXiv:1811.06879*, 2018. 1, 2

[11] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 1, 5, 8

[12] R.C. Veltkamp J.W. Tangelder. A survey of content based 3d shape retrieval methods. 2008. 1

[13] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 153–61, 2017. 1, 2

[14] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1):1–6, 1998. 4

[15] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9397–9406, 2018. 4

[16] Zhouhui Lian and Afzal A. Godil. A comparison of methods for non-rigid 3d shape retrieval. 2012. 1

[17] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 2, 5, 8

[18] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 1, 5

[19] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3):348–361, 2010. 2

[20] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002. 1

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 4

[22] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011. 1

[23] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001. 4

[24] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. Citeseer, 2009. 8

[25] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2155–2162. IEEE, 2010. 7

[26] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4, May 2011. 1, 5, 8

[27] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010. 7, 8

[28] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220, 2013. 1, 2, 5

[29] Ranjith Unnikrishnan and Martial Hebert. Multi-scale interest regions from unorganized point clouds. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008. 2

[30] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. 2018. 1

[31] 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. 1, 5

[32] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 607–623, 2018. 1, 2, 5, 6, 7, 8

[33] Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, and Radu Horaud. Surface feature detection and description with applications to mesh matching. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–380. IEEE, 2009. 2

[34] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 199–208. IEEE, 2017. 1, 2, 5

[35] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696. IEEE, 2009. 1, 2, 5, 8