KeypointNet: A Large-scale 3D Keypoint Dataset Aggregated from Numerous Human Annotations

Yang You, Yujing Lou,* Chengkun Li,* Zhoujun Cheng, Liangwei Li,
Lizhuang Ma, Cewu Lu, Weiming Wang†
Shanghai Jiao Tong University, China

Abstract

*Detecting 3D objects keypoints is of great interest to the areas of both graphics and computer vision. There have been several 2D and 3D keypoint datasets aiming to address this problem in a data-driven way. These datasets, however, either lack scalability or bring ambiguity to the definition of keypoints. Therefore, we present **KeypointNet**: the first large-scale and diverse 3D keypoint dataset that contains 83,231 keypoints and 8,329 3D models from 16 object categories, by leveraging numerous human annotations. To handle the inconsistency between annotations from different people, we propose a novel method to aggregate these keypoints automatically, through minimization of a fidelity loss. Finally, ten state-of-the-art methods are benchmarked on our proposed dataset.*

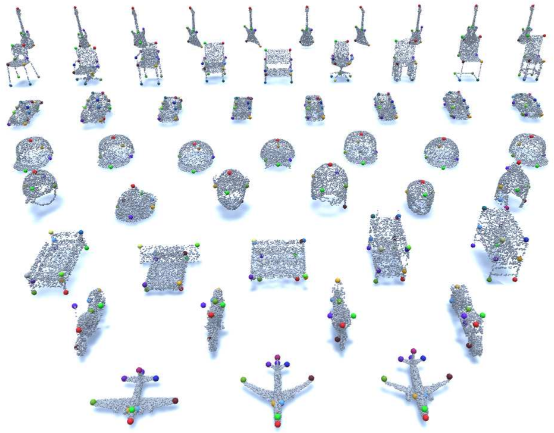

Figure 1. **We propose a large-scale KeypointNet dataset.** It contains 8K+ models and 83K+ keypoint annotations.

1. Introduction

Detection of 3D keypoints is essential in many applications such as object matching, object tracking, shape retrieval and registration [21, 6, 36]. Utilization of keypoints to match 3D objects has its advantage of providing features that are semantically significant and such keypoints are usually made invariant to rotations, scales and other transformations.

In the trend of deep learning, 2D semantic point detection has been boosted with the help of a large quantity of high-quality datasets [3, 22]. However, there are few 3D datasets focusing on the keypoint representation of an object. Dutagaci et al. [10] collect 43 models and label them according to annotations from various persons. Annotations from different persons are finally aggregated by geodesic clustering. ShapeNetCore keypoint dataset [41], and a similar dataset [14], in another way, resort to an expert's annotation on keypoints, making them vulnerable and biased.

In order to alleviate the bias of experts' definitions on keypoints, we ask a large group of people to annotate various keypoints according to their own understanding. Challenges rise in that different people may annotate different keypoints and we need to identify the consensus and patterns in these annotations. Finding such patterns is not trivial when a large set of keypoints spread across the entire model. A simple clustering would require a predefined distance threshold and fail to identify closely spaced keypoints. As shown in Figure 1, there are four closely spaced keypoints on each airplane empennage and it is extremely hard for simple clustering methods to distinguish them. Besides, clustering algorithms do not give semantic labels of keypoints since it is ambiguous to link clustered groups with each other. In addition, people's annotations are not always exact and errors of annotated keypoint locations are inevitable. In order to solve these problems, we propose a novel method to aggregate a large number of keypoint annotations from distinct people, by optimizing a fidelity loss. After this auto aggregation process, we verify these generated keypoints based on some simple priors such as symmetry.

In this paper, we build the first large-scale and diverse

*These authors contributed equally.
†Weiming Wang is the corresponding author.

dataset named **KeypointNet** which contains 8,329 models with 83,231 keypoints. These keypoints are of high fidelity and rich in structural or semantic meanings. Some examples are given in Figure 1. We hope this dataset could boost semantic understandings of common objects.

In addition, we propose two large-scale keypoint prediction tasks: keypoint saliency estimation and keypoint correspondence estimation. We benchmark ten state-of-the-art algorithms with mIoU, mAP and PCK metrics. Results show that the detection and identification of keypoints remain a challenging task.

In summary, we make the following contributions:

- To the best of our knowledge, we provide the first large-scale dataset on 3D keypoints, both in number of categories and keypoints.
- We come up with a novel approach on aggregating people's annotations on keypoints, even if their annotations are independent from each other.
- We experiment with ten state-of-the-art benchmarks on our dataset, including point cloud, graph, voxel and local geometry based keypoint detection methods.

2. Related Work

2.1. Detection of Keypoints

Detection of 3D keypoints has been a very important task for 3D object understanding which can be used in many applications, such as object pose estimation, reconstruction, matching, segmentation, etc. Researchers have proposed various methods to produce interest points on objects to help further objects processing. Traditional methods like 3D Harris [30], HKS [31], Salient Points [7], Mesh Saliency [17], Scale Dependent Corners [23], CGF [13], SHOT [34], etc, exploit local reference frames (LRF) to extract geometric features as local descriptors. However, these methods only consider the local geometric information without semantic knowledge, which forms a gap between detection algorithms and human understanding.

Recent deep learning methods like SyncSpecCNN [41], deep functional dictionaries [32] are proposed to detect keypoints. Deep learning methods utilize the ground-truth labels which are annotated by human with expert verification. Exploiting the semantic labels, keypoint detectors can learn from the dataset and perform well.

2.2. Keypoint Datasets

Keypoint datasets have its origin in 2D images, where plenty of datasets on human skeletons and object interest points are proposed. For human skeletons, MPII human pose dataset [3], MSCOCO keypoint challenge [1] and PoseTrack [2] annotate millions of keypoints on humans. For more general objects, SPair-71k [22] contains 70,958 image pairs with diverse variations in viewpoint and scale, with a number of corresponding keypoints on each image pair. PUB [35] provides 15 part locations on 11,788 images from 200 bird categories and PASCAL [5] provides keypoint annotations for 20 object categories. HAKE [18] provides numerous annotations on human interactiveness keypoints. ADHA [24] annotates key adverbs in videos, which is a sequence of 2D images.

Keypoint datasets on 3D objects, include Dutagaci et al. [10], SyncSpecCNN [41] and Kim et al. [14]. Dutagaci et al. [10] aggregates multiple annotations from different people with an ad-hoc method while the dataset is extremely small. Though SyncSpecCNN [41], Pavlakos et al. [25] and Kim et al. [14] give a relatively large keypoint dataset, they rely on a manually designed template of keypoints, which is inevitably biased and flawed. GraspNet [11] gives dense annotations on 3D object grasping. The differences between theirs and ours are illustrated in Table 1.

3. KeypointNet: A Large-scale 3D Keypoint Dataset

3.1. Data Collection

KeypointNet is built on ShapeNetCore [8]. ShapeNetCore covers 55 common object categories with about 51,300 unique 3D models.

We filter out those models that deviate from the majority and keep at most 1000 instances for each category in order to provide a balanced dataset. In addition, a consistent canonical orientation is established (e.g., upright and front) for every category because of the incomplete alignment in ShapeNetCore.

We let annotators determine which points are important, and same keypoint indices should indicate same meanings for each annotator. Though annotators are free to give their own keypoints, three general principles should be obeyed: (1) each keypoint should describe an object's semantic information shared across instances of the same object category, (2) keypoints of an object category should spread over the whole object and (3) different keypoints have distinct semantic meanings. After that, we utilize a heuristic method to aggregate these points, which will be discussed in Section 4.

Keypoints are annotated on meshes and these annotated meshes are then downsampled to 2,048 points. Our final dataset is a collection of point clouds, with keypoint indices.

3.2. Annotation Tools

We develop an easy-to-use web annotation tool based on NodeJS. Every user is allowed to click up to 20 interest points according to his/her own understanding. The UI interface is shown in Figure 2. Annotated models are shown

Dataset	Domain	Correspondence	Template-free	Instances	Categories	Keypoints	Format
FAUST [4]	human	√	×	100	1	689K	mesh
SyncSpecCNN [41]	chair	√	×	6243	1	∼60K	point cloud
Dutagaci et al. [10]	general	×	√	43	16	<1K	mesh
Kim et al. [15]	general	√	×	404	4	∼3K	mesh
PASCAL 3D+ [39]	general	×	×	36292	12	150K+	RGB w. 3D model
Ours	general	√	√	8329	16	83K+	point cloud & mesh

Table 1. **Comparison of 3d keypoint datasets**. **Correspondence** indicates whether keypoints are indexed correspondingly. **Template-free** indicates whether it avoids hardcoded keypoint templates.

in the left panel while the next unprocessed model is shown in the right panel.

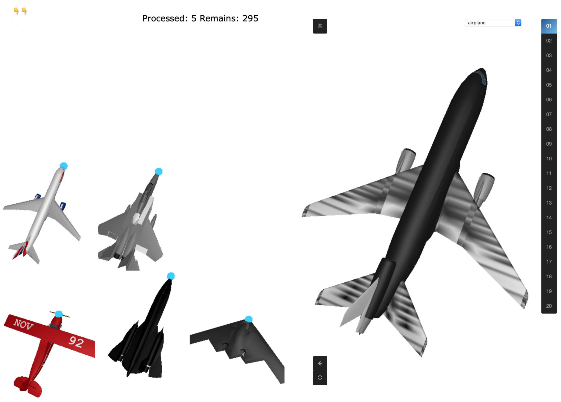

Figure 2. Web interface of the annotation tool.

3.3. Dataset Statistics

At the time of this work, our dataset has collected 16 common categories from ShapeNetCore, with 8329 models. Each model contains 3 to 20 keypoints. Our dataset is divided into train, validation and test splits, with 7:1:2 ratio. Table 2 gives detailed statistics of our dataset. Some visualizations of our dataset is given in Figure 3.

4. Keypoint Aggregation

Given all human labeled raw keypoints, we leverage a novel method to aggregate them together into a set of *ground-truth* keypoints.

There are generally two reasons: 1) distinct people may annotate different sets of keypoints and human labeled keypoints are sometimes erroneous, so we need an elegant way to aggregate these keypoints; 2) a simple clustering algorithm would fail to distinguish those closely spaced keypoints and cannot give consistent semantic labels.

4.1. Problem Statement

Given a 2-dimensional sub-manifold $\mathcal{M}_m \subset \mathbb{R}^3$, where m is the index of the model, a valid annotation from the c-th person is a keypoint set $\{l_{m,k}^{(c)} | l_{m,k}^{(c)} \in \mathcal{M}_m\}_{k=1}^{K_c}$, where k is the keypoint index and K_c is the number of keypoints annotated by person c. Note that different people may have different sets of keypoint indices and these indices are independent.

Our goal is to aggregate a set of potential ground-truth keypoints $\mathcal{Y} = \{y_{m,k} | y_{m,k} \in \mathcal{M}_m, m = 1, \dots M, k = 1, \dots K_m\}$, where K_m is the number of proposed keypoints for each model $\mathcal{M}_m$, so that $y_{m_1,k}$ and $y_{m_2,k}$ share the same semantic.

4.2. Keypoint Saliency

Each annotation is allowed to be erroneous within a small region, so that a keypoint distribution is defined as follows:

$$p(x|x \text{ is the } k\text{-th keypoint}, x \in \mathcal{M}_m) = \frac{\phi(l_{m,k}, x)}{Z(\phi)},$$

where ϕ is Gaussian kernel function. Z is a normalization constant. This contradicts many previous methods on annotating keypoints where a δ-function is implicitly assumed. We argue that it is common that humans make mistakes when annotating keypoints and due to central limit theorem, the keypoint distribution would form a Gaussian.

4.3. Ground-truth Keypoint Generation

We propose to jointly output a dense mapping function $g_\theta : \mathcal{M} \to \mathbb{R}^d$ whose parameters are θ, and the aggregated ground-truth keypoint set $\mathcal{Y}$. g_θ transforms each point into an high-dimensional embedding vector in $\mathbb{R}^d$. Specifically, we solve the following optimization problem:

$$\begin{aligned}(\theta^*, \mathcal{Y}^*) = \arg\min_{\theta, \mathcal{Y}} [f(\mathcal{Y}, g_\theta) + H(g_\theta)] \\ s.t.\ g_\theta(y_{m_1,k}) \equiv g_\theta(y_{m_2,k}), \forall m_1, m_2, k.\end{aligned} \quad (1)$$

where $f(\mathcal{Y}, g_\theta)$ is the data fidelity loss and $H(g_\theta)$ is a regularization term to avoid trivial solution like $g_\theta \equiv 0$. The constraint states that the embedding of ground-truth keypoints with the same index should be the same.

Fidelity Loss We define $f(\mathcal{Y}, g_\theta)$ as:

$$f(\mathcal{Y}, g_\theta) = \sum_{m=1}^{M} \sum_{c=1}^{C} \sum_{k=1}^{K_m} \int_{\mathcal{M}_m} \mathbf{d}_\theta(x, y_{m,k}) \frac{\phi(l_{m,n^*}^{(c)}, x)}{Z(\phi)} dx,$$

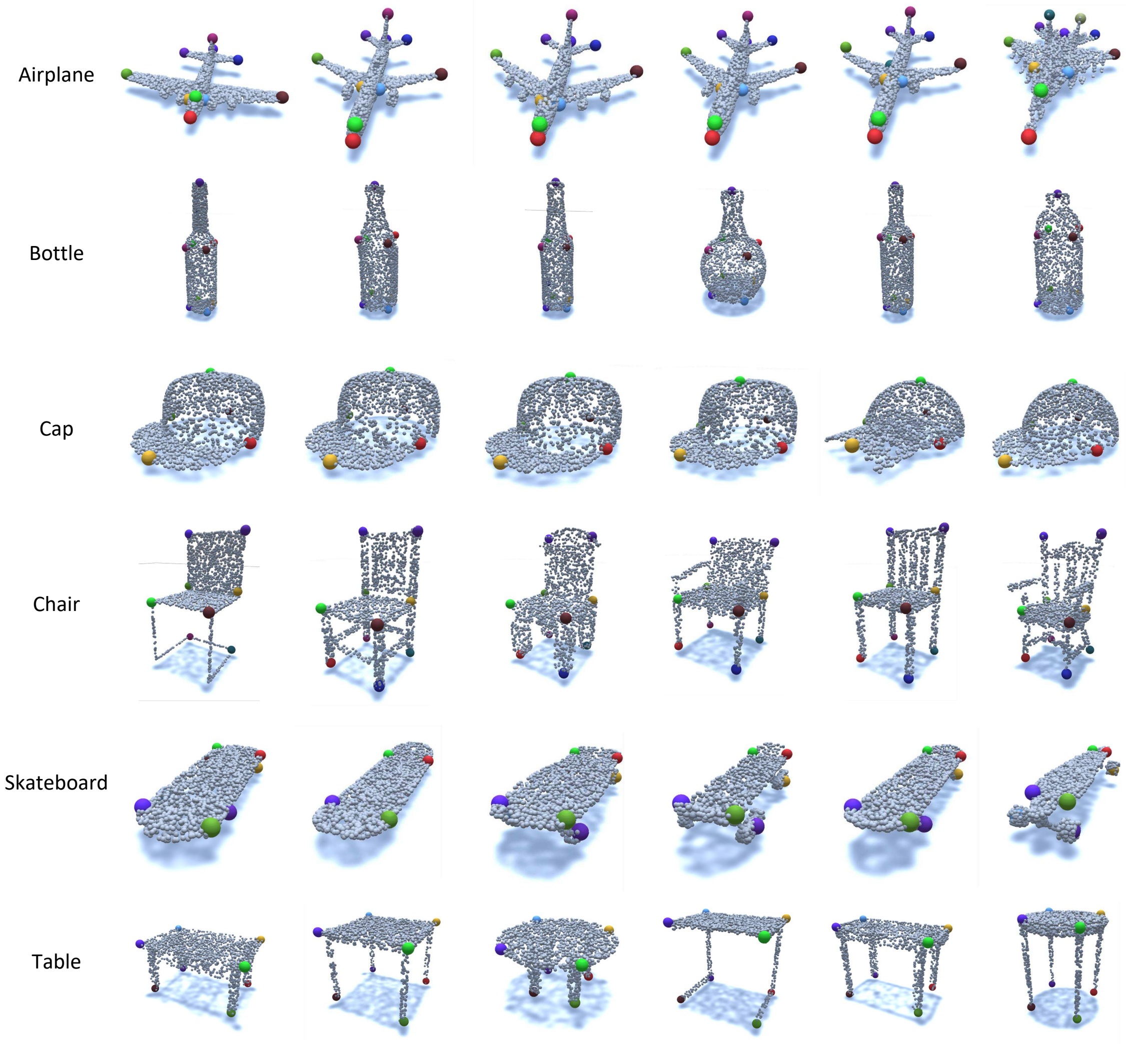

Figure 3. **Dataset Visualization.** Here we plot ground-truth keypoints for several categories. We can see that by utilizing our automatic aggregation method, keypoints of high fidelity are extracted.

where $\mathbf{d}_\theta$ is the L2 distance between two vectors in embedding space:

$$\mathbf{d}_\theta(a, b) = \|g_\theta(a) - g_\theta(b)\|_2^2,$$

and

$$n^* = \arg\min_n \mathbf{d}_\theta(l_{m,n}^{(c)}, y_m).$$

Unlike previous methods such as Dutagaci et al.[10] where a simple geodesic average of human labeled points is given as ground-truth points, we seek a point whose expected embedding distance to all human labeled points is smallest. The reason is that geodesic distance is sensitive to the misannotated keypoints and could not distinguish closely spaced keypoints, while embedding distance is more robust to noisy points as the embedding space encodes the semantic information of an object.

Equation 1 involves both θ and $\mathcal{Y}$ and it is impractical to solve this problem in closed form. In practice, we use alternating minimization with a deep neural network to approximate the embedding function g_θ, so that we solve the following dual problem instead (by slightly loosening the constraints):

$$\begin{aligned}\theta^* = &\arg\min_\theta [H(g_\theta) \\ &+ \lambda \sum_{m_1, m_2}^{M} \sum_{k}^{K_m} \|g_\theta(y_{m_1,k}) - g_\theta(y_{m_2,k})\|_2^2],\end{aligned} \tag{2}$$

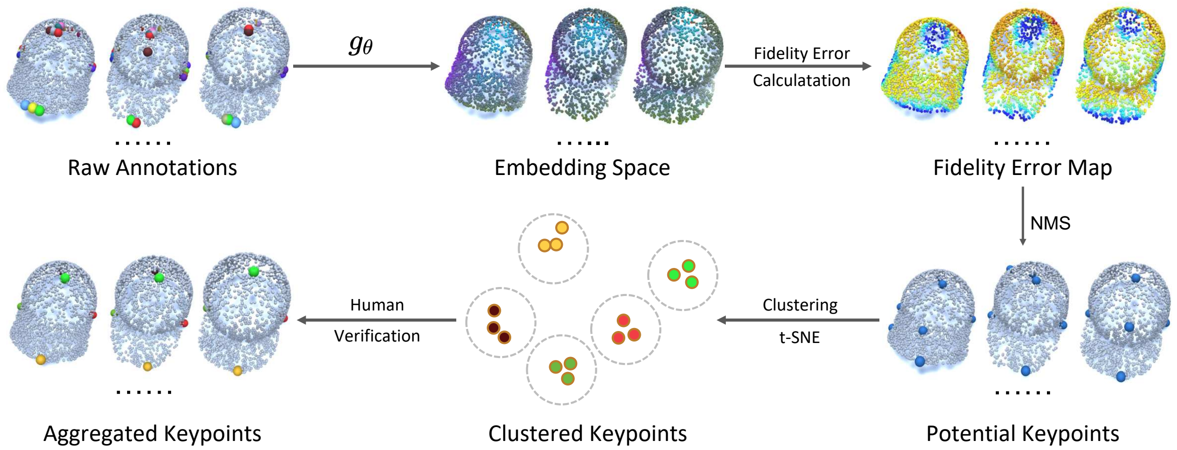

Figure 4. **Keypoint aggregation pipeline.** We first infer dense embeddings from human labeled raw annotations. Then fidelity error maps are calculated by summing embedding distances to human labeled keypoints. Non Minimum Suppression is conducted to form a potential set of keypoints. These keypoints are then projected onto 2D subspace with t-SNE and verified by humans.

Category	Train	Val	Test	All	#Annotators
Airplane	718	103	206	1027	21
Bathtub	351	50	101	502	11
Bed	110	16	32	158	6
Bottle	277	39	80	396	8
Cap	29	4	9	42	6
Car	703	101	201	1005	14
Chair	714	102	205	1021	15
Guitar	430	62	123	615	11
Helmet	70	10	21	101	5
Knife	217	31	62	310	5
Laptop	312	44	90	446	10
Motorcycle	210	30	61	301	7
Mug	134	19	39	192	8
Skateboard	105	15	31	151	6
Table	793	113	227	1133	13
Vessel	650	93	186	929	13
Total	5823	832	1674	8329	159

Category	Train	Val	Test	All
Airplane	6987	1014	2061	10062
Bathtub	4620	682	1368	6670
Bed	1386	206	420	2012
Bottle	2488	348	716	3552
Cap	145	20	44	209
Car	11189	1601	3209	15999
Chair	8508	1174	2442	12124
Guitar	2686	404	769	3859
Helmet	478	68	143	689
Knife	651	93	186	930
Laptop	1859	263	537	2659
Motorcycle	1679	240	488	2407
Mug	1472	208	429	2109
Skateboard	792	112	232	1136
Table	6392	904	1825	9121
Vessel	6781	979	1933	9693
Total	58113	8316	16802	83231

Table 2. **Keypoint Dataset statistics.** Left: number of models in each category. Right: number of keypoints in each category.

$$\mathcal{Y}^* = \arg\min_{\mathcal{Y}} \sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{k=1}^{K_m}\int_{\mathcal{M}_m} \mathbf{d}_\theta(x, y_{m,k}) \frac{\phi(l^{(c)}_{m,n^*}, x)}{Z(\phi)} dx,$$
$$s.t.\ g_\theta(y_{m_1,k}) \equiv g_\theta(y_{m_2,k}), \forall m_1, m_2, k, \tag{3}$$

and alternate between the two equations until convergence.

By solving this problem, we find both an optimal embedding function g_θ, together with intra-class consistent ground-truth keypoints $\mathcal{Y}$, while keeping its embedding distance from human-labeled keypoints as close as possible. The ground-truth keypoints can be viewed as the projection of human labeled data onto embedding space.

Non Minimum Suppression Equation 3 may be hard to solve since K_m is also unknown beforehand. For each model $\mathcal{M}_m$, the fidelity error associated with each potential keypoint $y_m \in \mathcal{M}_m$ is:

$$f(y_m, g_\theta) = \sum_{m' \neq m}^{M}\sum_{c=1}^{C}\int_{\mathcal{M}_{m'}} \mathbf{d}_\theta(x, y_{m'}) \frac{\phi(l^{(c)}_{m',n^*}, x)}{Z(\phi)} dx, \tag{4}$$

where $y_{m'} = \arg\min_{y_{m'} \in \mathcal{M}_{m'}} \|g_\theta(y_{m'}) - g_\theta(y_m)\|_2^2$.

Then y_m^* is found by conducting Non Minimum Suppression (NMS), such that:

$$f(y_m^*, g_\theta) \leq f(y_m, g_\theta),$$
$$\forall y_m \in \mathcal{M}_m, \mathbf{d}_\theta(y_m, y_m^*) < \delta, \tag{5}$$

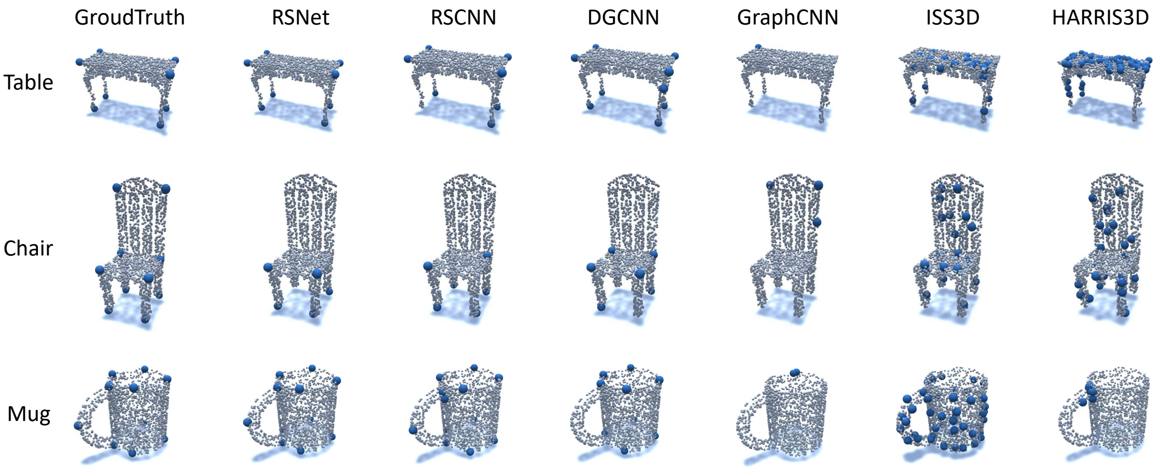

Figure 5. Visualizations of detected keypoints for six algorithms.

	Airplane	**Bathtub**	**Bed**	**Bottle**	**Cap**	**Car**	**Chair**	**Guitar**	**Helmet**	**Knife**	**Laptop**	**Motor**	**Mug**	**Skate**	**Table**	**Vessel**	**Average**
PointNet	9.1/8.5	0.5/3.6	6.4/6.4	0.0/1.3	0.0/3.2	0.0/2.3	4.5/9.6	0.0/1.0	0.0/0.4	0.0/16.3	11.6/14.5	1.9/2.6	0.0/3.4	0.0/1.7	11.0/12.0	0.0/2.2	2.8/5.6
PointNet++	20.5/33.6	10.2/5.7	16.1/18.4	**22.0/26.0**	**30.7/32.2**	**40.3/49.9**	**27.3/39.7**	**31.5/36.6**	**42.3/47.2**	20.5/27.8	29.8/38.9	15.7/14.3	**22.0/35.4**	**48.2/31.3**	18.0/25.9	12.4/16.7	**25.5/30.0**
RSNet	20.5/31.1	12.8/17.8	19.2/29.5	13.1/12.8	15.7/21.8	15.1/21.8	13.9/15.4	16.4/16.1	8.4/6.1	18.3/31.5	22.8/35.0	20.2/26.1	16.8/23.2	4.0/5.4	15.4/45.3	9.7/12.8	15.1/22.0
SpiderCNN	22.2/25.8	7.2/6.7	17.7/19.8	4.1/2.7	2.7/4.0	5.5/6.5	15.9/18.9	7.1/10.5	0.0/0.4	**30.0**/28.4	22.4/34.3	14.5/15.0	4.9/5.3	0.0/1.7	23.9/30.2	8.5/8.9	11.7/13.7
PointConv	25.3/28.1	15.2/24.6	**32.4/45.8**	7.3/10.1	13.5/15.7	20.3/24.6	21.7/30.8	21.2/21.7	2.1/2.0	5.0/17.3	27.8/**46.5**	18.9/**29.3**	21.7/27.3	13.2/18.9	26.8/42.4	13.9/22.6	17.9/25.5
RSCNN	21.0/34.4	11.9/17.3	19.3/28.4	11.6/16.8	18.9/31.6	15.8/16.3	17.6/21.7	17.9/18.2	0.0/3.1	24.2/30.6	25.3/37.9	13.4/23.8	17.2/25.4	5.9/9.7	23.7/41.4	10.1/14.7	15.9/23.2
DGCNN	**32.3/43.8**	**17.7/26.2**	21.6/33.4	15.0/20.7	21.5/27.6	15.1/21.4	23.8/30.3	20.7/22.9	3.5/4.8	29.4/**40.5**	**30.1**/46.4	**23.5**/29.2	18.1/24.9	12.8/17.3	**31.7/52.1**	**15.6/19.7**	20.8/28.8
GraphCNN	22.9/25.9	12.5/14.9	0.7/0.8	1.8/5.2	0.2/0.3	10.3/10.5	12.7/14.8	1.0/5.7	0.3/0.4	0.1/0.2	0.3/0.3	18.7/23.0	10.8/11.4	0.4/0.5	24.2/34.4	8.9/9.7	7.9/9.9
Harris3D	0.4/-	0.3/-	1.0/-	1.0/-	0.0/-	0.7/-	1.4/-	1.6/-	0.2/-	0.0/-	0.0/-	0.3/-	0.3/-	0.5/-	0.7/-	3.3/-	0.7/-
SIFT3D	4.5/-	0.9/-	0.9/-	0.7/-	1.0/-	1.2/-	0.9/-	0.2/-	0.9/-	0.0/-	0.5/-	0.7/-	0.7/-	0.3/-	1.0/-	2.2/-	1.0/-
ISS3D	0.4/-	1.0/-	0.5/-	0.9/-	1.9/-	2.0/-	0.0/-	0.6/-	0.8/-	0.0/-	0.2/-	0.3/-	0.5/-	0.5/-	0.0/-	3.3/-	0.8/-

Table 3. mIoU and mAP results (in percentage) for compared methods with distance threshold 0.01.

where δ is some neighborhood threshold.

After NMS, we would get several ground-truth points $y_{m,1}, y_{m,2}, \ldots, y_{m,k}$ for each manifold $\mathcal{M}_m$. However, the arbitrarily assigned index k within each model does not provide a consistent semantic correspondence across different models. Therefore we cluster these points according to their embeddings by first projecting them onto 2D subspace with t-SNE [20].

Ground-truth Verification Though the above method automatically aggregate a set of potential set of keypoints with high precision, it omits some keypoints in some cases. As the last step, experts manually verify these keypoints based on some simple priors such as the rotational symmetry and centrosymmetry of an object.

4.4. Implementation Details

At the start of the alternating minimization, we initialize $\mathcal{Y}$ to be sampled from raw annotations and then run one iteration, which is enough for the convergence. We choose PointConv with hidden dimension 128 as the embedding function g. During the optimization of Equation 3, we classify each point into K classes with a SoftMax layer and extract the feature of the last but one layer as the embedding. The learning rate is 1e-3 and the optimizer is Adam [16].

4.5. Pipeline

The whole pipeline is shown in Figure 4. We first infer dense embeddings from human labeled raw annotations. Then fidelity error maps are calculated by summing embedding distances to human labeled keypoints. Non Minimum Suppression is conducted to form a potential set of keypoints. These keypoints are then projected onto 2D subspace with t-SNE and verified by humans.

5. Tasks and Benchmarks

In this section, we propose two keypoint prediction tasks: keypoint saliency estimation and keypoint correspondence estimation. Keypoint saliency estimation requires evaluated methods to give a set of potential indistinguishable keypoints while keypoint correspondence estimation asks to localize a fixed number of distinguishable keypoints.

5.1. Keypoint Saliency Estimation

Dataset Preparation For keypoint saliency estimation, we only consider whether a point is the keypoint or not, without giving its semantic label. Our dataset is split into train, validation and test sets with the ratio 70%, 10%, 20%.

Evaluation Metrics Two metrics are adopted to evaluate the performance of keypoint saliency estimation. Firstly, we evaluate their mean Intersection over Unions [33] (mIoU), which can be calculated as

$$IoU = \frac{TP}{TP + FP + FN}. \quad (6)$$

mIoU is calculated under different error tolerances from 0 to 0.1. Secondly, for those methods that output keypoint probabilities, we evaluate their mean Average Precisions (mAP) over all categories.

Benchmark Algorithms We benchmark seven state-of-the-art algorithms on point cloud semantic analysis: PointNet [26], PointNet++ [27], RSNet [12], SpiderCNN [40], PointConv [38], RSCNN [19], DGCNN [37] and GraphCNN [9]. Three traditional local geometric keypoint detectors are also considered: Harris3D [30], SIFT3D [28] and ISS3D [29].

Evaluation Results For deep learning methods, we use the default network architectures and hyperparameters to predict the keypoint probability of each point and mIoU and mAP are adopted to evaluate their performance. For local geometry based methods, mIoU is used. Each method is tested with various geodesic error thresholds. In Table 3, we report mIoU and mAP results under a restrictive threshold 0.01. Figure 6 shows the mIoU curves under different distance thresholds from 0 to 0.1 and Figure 7 shows the mAP results. We can see that under a restrictive distance threshold 0.01, geometric and deep learning methods both fail to predict qualified keypoints.

Figure 5 shows some visualizations of the results from RSNet, RSCNN, DGCNN, GraphCNN, ISS3D and Harris3D. Deep learning methods can predict some of ground-truth keypoints while the predicted keypoints are sometimes missing. For local geometry based methods like ISS3D and Harris3D, they give much more interest points spread over the entire model while these points are agnostic of semantic information. Learning discriminative features for better localizing accurate and distinct keypoints across various objects is still a challenging task.

5.2. Keypoint Correspondence Estimation

Keypoint correspondence estimation is a more challenging task, where one needs to predict not only the keypoints, but also their semantic labels. The semantic labels should be consistent across different objects in the same category.

Dataset Preparation For keypoint correspondence estimation, each keypoint is labeled with a semantic index. For those keypoints that do not exist on some objects, index -1 is given. Similar to SyncSpecCNN [41], the maximum number of keypoints is fixed and the data split is the same as keypoint saliency estimation.

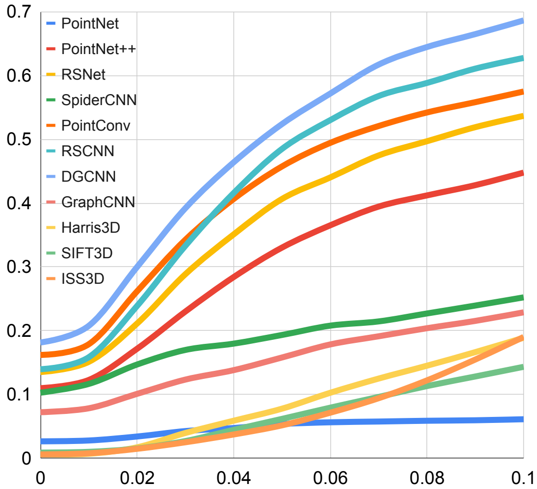

Figure 6. mIoU results under various distance thresholds (0-0.1) for compared algorithms.

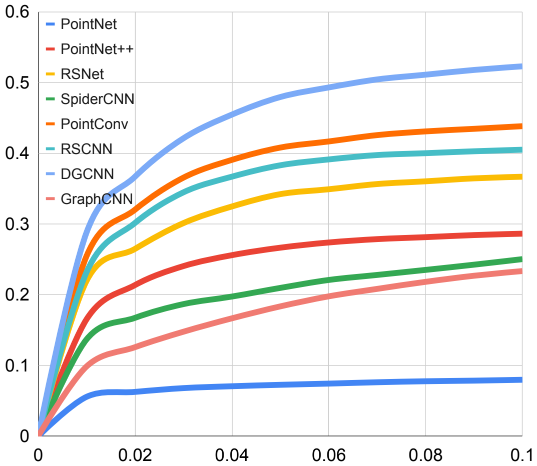

Figure 7. mAP results under various distance thresholds (0-0.1) for compared algorithms.

Evaluation Metric The prediction of network is evaluated by the percentage of correct keypoints (PCK), which is used to evaluate the accuracy of keypoint prediction in many previous works [41, 32].

Benchmark Algorithms We benchmark seven state-of-the-art algorithms on point cloud semantic analysis: PointNet [26], PointNet++[27], RSNet [12], SpiderCNN [40], PointConv [38], RSCNN [19], DGCNN [37] and GraphCNN [9].

Evaluation Results Similarly, we use the default network architectures. Table 4 shows the PCK results with error dis-

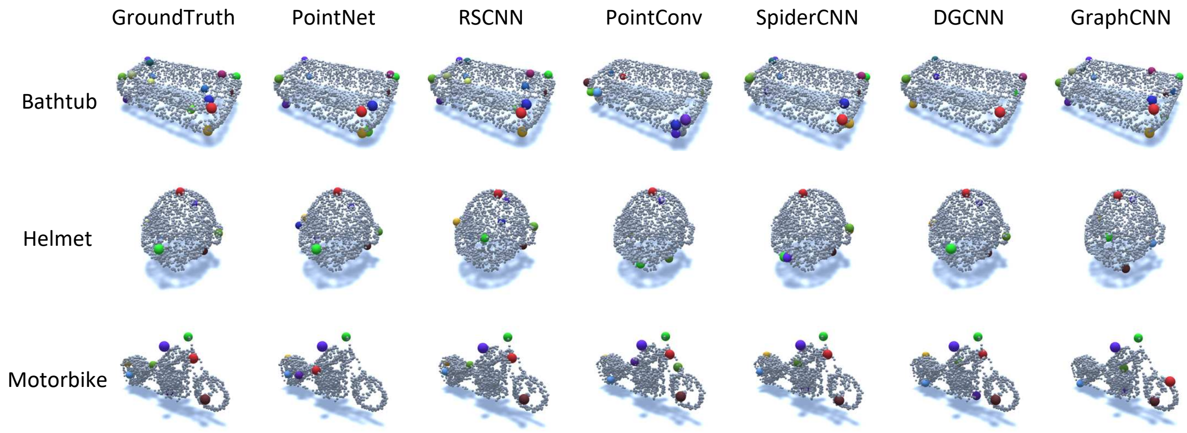

Figure 8. Visualizations of detected keypoints and their semantic labels. Same colors indicate same semantic labels.

	Airplane	**Bath**	**Bed**	**Bottle**	**Cap**	**Car**	**Chair**	**Guitar**	**Helmet**	**Knife**	**Laptop**	**Motor**	**Mug**	**Skate**	**Table**	**Vessel**	**Average**
PointNet	42.7	19.5	28.9	**25.0**	**85.0**	22.9	13.2	18.8	2.8	54.2	47.9	25.0	30.2	15.9	39.6	16.7	30.5
PointNet++	42.3	24.1	32.7	21.5	45.0	30.4	25.3	23.3	6.5	26.7	42.2	32.5	22.4	12.1	55.8	17.6	28.8
RSCNN	36.9	27.8	34.6	15.6	38.3	30.8	21.5	**32.6**	4.7	33.3	52.2	40.0	25.5	17.2	49.2	19.4	30.0
RS-Net	38.3	**28.8**	**46.3**	24.0	20.0	**39.3**	**24.1**	29.2	**9.6**	**57.8**	**60.0**	**45.8**	**31.7**	18.2	36.7	19.4	**33.1**
SpiderCNN	**44.3**	19.4	32.2	12.6	80.0	18.3	23.7	26.7	6.5	24.4	40.0	34.2	21.2	19.8	**54.2**	22.4	30.0
PointConv	40.3	0.0	0.0	15.6	55.0	13.3	22.6	20.9	3.7	33.3	50.0	35.0	25.5	**25.0**	42.5	21.2	25.2
DGCNN	38.9	20.3	21.9	14.2	10.0	16.2	13.9	19.8	8.3	38.9	44.4	21.9	16.0	9.8	36.8	9.0	21.3
GraphCNN	41.1	23.3	25.0	16.0	13.3	18.5	20.0	17.5	3.0	37.8	44.4	31.7	15.0	11.5	40.0	**24.4**	23.9

Table 4. PCK results under distance threshold 0.01 for various deep learning networks.

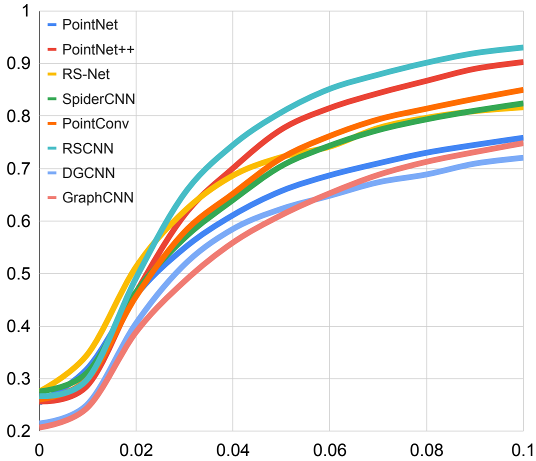

Figure 9. PCK results under various distance thresholds (0-0.1) for compared algorithms.

tance threshold 0.01. Figure 9 illustrates the percentage of correct points curves with distance thresholds varied from 0 to 0.1. RS-Net performs relatively better than other methods with the distance threshold under 0.02. RSCNN gives better results by a large margin with the distance threshold above 0.02. However, all seven methods face big difficulties in predicting exact consistent semantic keypoints.

Figure 8 shows some visualizations of the results for different methods. Same colors denote same semantic labels. We can see that most methods can accurately predict some of keypoints. However, there are still some missing keypoints and inaccurate localizations.

Keypoint saliency estimation and keypoint correspondence estimation are both important for object understanding. Keypoint saliency estimation gives a spare representation of object by extracting meaningful points. Keypoint correspondence estimation establishes relations between points on different objects. From the results above, we can see that these two tasks still remain challenging. The reason is that object keypoints from human perspective are not simply geometrically salient points but abstracts semantic meanings of the object.

6. Conclusion

In this paper, we propose a large-scale and high-quality KeypointNet dataset. In order to generate ground-truth keypoints from raw human annotations where identification of their modes are non-trivial, we transform the problem into an optimization problem and solve it in an alternating fashion. By optimizing a fidelity loss, ground-truth keypoints, together with their correspondences are generated. In addition, we evaluate and compare several state-of-the-art methods on our proposed dataset and we hope this dataset could boost the semantic understanding of 3D objects.

References

[1] Mscoco keypoint challenge 2016, 2016. 2

[2] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5167–5176, 2018. 2

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1, 2

[4] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE. 3

[5] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1365–1372. IEEE, 2009. 2

[6] M Bueno, J Martínez-Sánchez, H González-Jorge, and H Lorenzo. Detection of geometric keypoints and its application to point cloud coarse registration. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41, 2016. 1

[7] 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

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 7

[10] Helin Dutagaci, Chun Pan Cheung, and Afzal Godil. Evaluation of 3d interest point detection techniques via human-generated ground truth. *The Visual Computer*, 28(9):901–917, 2012. 1, 2, 3, 4

[11] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet: A large-scale clustered and densely annotated datase for object grasping. *arXiv preprint arXiv:1912.13470*, 2019. 2

[12] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2635, 2018. 7

[13] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 153–161, 2017. 2

[14] Vladimir G Kim, Wilmot Li, Niloy J Mitra, Siddhartha Chaudhuri, Stephen DiVerdi, and Thomas Funkhouser. Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG)*, 32(4):70, 2013. 1, 2

[15] Vladimir G Kim, Wilmot Li, Niloy J Mitra, Stephen DiVerdi, and Thomas Funkhouser. Exploring collections of 3d models using fuzzy correspondences. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 3

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[17] Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. *ACM transactions on graphics (TOG)*, 24(3):659–666, 2005. 2

[18] Yong-Lu Li, Liang Xu, Xijie Huang, Xinpeng Liu, Ze Ma, Mingyang Chen, Shiyi Wang, Hao-Shu Fang, and Cewu Lu. Hake: Human activity knowledge engine. *arXiv preprint arXiv:1904.06539*, 2019. 2

[19] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 7

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 6

[21] Ajmal S Mian, Mohammed Bennamoun, and Robyn Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1584–1601, 2006. 1

[22] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019. 1, 2

[23] John Novatnack and Ko Nishino. Scale-dependent 3d geometric features. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2

[24] Bo Pang, Kaiwen Zha, Yifan Zhang, and Cewu Lu. Further understanding videos through adverbs: A new video task. In *AAAI*, 2020. 2

[25] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2011–2018. IEEE, 2017. 2

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 7

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 7

[28] Blaine Rister, Mark A Horowitz, and Daniel L Rubin. Volumetric image registration from invariant keypoints.

IEEE Transactions on Image Processing, 26(10):4900–4910, 2017. 7

[29] Samuele Salti, Federico Tombari, Riccardo Spezialetti, and Luigi Di Stefano. Learning a descriptor-specific 3d keypoint detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2318–2326, 2015. 7

[30] Ivan Sipiran and Benjamin Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963, 2011. 2, 7

[31] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009. 2

[32] Minhyuk Sung, Hao Su, Ronald Yu, and Leonidas J Guibas. Deep functional dictionaries: Learning consistent semantic structures on 3d models from functions. In *Advances in Neural Information Processing Systems*, pages 485–495, 2018. 2, 7

[33] Leizer Teran and Philippos Mordohai. 3d interest point detection via discriminative learning. In *European Conference on Computer Vision*, pages 159–173. Springer, 2014. 7

[34] 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. 2

[35] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2

[36] Hanyu Wang, Jianwei Guo, Dong-Ming Yan, Weize Quan, and Xiaopeng Zhang. Learning 3d keypoint descriptors for non-rigid shape matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1

[37] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):146, 2019. 7

[38] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 7

[39] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014. 3

[40] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018. 7

[41] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2282–2290, 2017. 1, 2, 3, 7