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.

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...
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
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 \( M_m \subset \mathbb{R}^3 \), where \( m \) is the index of the model, a valid annotation from the \( c \)-th person is a keypoint set \( \{(l_{m,k})^c | y_{m,k} \in M_m, k = 1, \ldots, K_c\} \), 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 \( \mathcal{Y} = \{y_{m,k} | y_{m,k} \in M_m, m = 1, \ldots, M, k = 1, \ldots, K_m\} \), where \( K_m \) is the number of proposed keypoints for each model \( M_m \), 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:

\[
p(x|x) = \frac{\phi(l_{m,k}, x)}{Z(\phi)},
\]

where \( \phi \) is Gaussian kernel function. \( Z \) is a normalization constant. This contradicts many previous methods on annotating keypoints where a \( \delta \)-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 \( g_\theta : M \rightarrow \mathbb{R}^d \) whose parameters are \( \theta \), and the aggregated ground-truth keypoint set \( \mathcal{Y} \). \( g_\theta \) transforms each point into an high-dimensional embedding vector in \( \mathbb{R}^d \). Specifically, we solve the following optimization problem:

\[
(\theta^*, \mathcal{Y}^*) = \arg \min_{\theta, \mathcal{Y}} \left[ f(\mathcal{Y}, g_\theta) + H(g_\theta) \right]
\]

subject to \( g_\theta(y_{m_1,k}) \equiv g_\theta(y_{m_2,k}), \forall m_1, m_2, k \).

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 \quad We define \( f(\mathcal{Y}, g_\theta) \) as:

\[
f(\mathcal{Y}, g_\theta) = \sum_{m=1}^M C \sum_{c=1}^C K_m \int_{\mathcal{M}_m} d_\theta(x, y_{m,k}) \frac{\phi(l_{m,k}, x)}{Z(\phi)} dx,
\]
where $d_\theta$ is the L2 distance between two vectors in embedding space:

$$d_\theta(a, b) = \|g_\theta(a) - g_\theta(b)\|_2^2,$$

and

$$n^* = \arg \min_n d_\theta(l^{(c)}_{m,n}, y_m).$$

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 $\theta$ 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 $g_\theta$, so that we solve the following dual problem instead (by slightly loosening the constraints):

$$\theta^* = \arg \min_\theta \|H(g_\theta)\|_2 + \lambda \sum_{m1, m2} \sum_k \|g_\theta(y_{m1,k}) - g_\theta(y_{m2,k})\|_2^2,$$

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.
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.

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

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

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

(3)

and alternate between the two equations until convergence.

By solving this problem, we find both an optimal embedding function \( g_{\theta} \), 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:

\[
f(y_m, g_{\theta}) = \sum_{m' \neq m} \sum_{c=1}^{C} \int_{\mathcal{M}_{m'}} d\theta(x, y_{m'}) \frac{\phi(t_{m,n}^{(c)}, x)}{Z(\phi)} dx,
\]

(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:

\[
f(y_m^*, g_{\theta}) \leq f(y_m, g_{\theta}), \forall y_m \in \mathcal{M}_m, d\theta(y_m, y_m^*) < \delta,
\]

(5)
The learning rate is 1e-3 and the optimizer is Adam \[\text{Eq. 3}\]. After NMS, we would get several ground-truth points \(y_{m,1}, y_{m,2}, \ldots, y_{m,k}\) for each manifold \(M_m\). 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 \[\text{Eq. 4}\].

**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 \(Y\) 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 \[\text{Eq. 4}\].

### 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%.

### 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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Airplane</th>
<th>Bathrub</th>
<th>Bed</th>
<th>Bottle</th>
<th>Cap</th>
<th>Chair</th>
<th>Guitar</th>
<th>Helmet</th>
<th>Knife</th>
<th>Laptop</th>
<th>Motor</th>
<th>Mug</th>
<th>Skate</th>
<th>Table</th>
<th>Vessel</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>PointNet</td>
<td>9.1/8.5</td>
<td>0.5/3.6</td>
<td>6.4/6.4</td>
<td>0.0/1.3</td>
<td>0.0/1.2</td>
<td>0.0/2.3</td>
<td>4.5/5.6</td>
<td>0.01/1.0</td>
<td>0.0/0.4</td>
<td>0.3/4.72</td>
<td>2.1/2.7</td>
<td>29.8/38.8</td>
<td>16.1/14.5</td>
<td>2.0/0.55</td>
<td>2.0/0.55</td>
<td>20.5/33.6</td>
</tr>
<tr>
<td>PointNet++</td>
<td>20.5/33.6</td>
<td>10.2/3.7</td>
<td>16.1/18.8</td>
<td>12.0/26.0</td>
<td>30.7/32.2</td>
<td>40.3/49.9</td>
<td>27.3/39.7</td>
<td>31.5/56.6</td>
<td>24.0/27.5</td>
<td>15.5/21.3</td>
<td>15.8/31.5</td>
<td>18.0/30.2</td>
<td>13.9/11.9</td>
<td>10.3/10.5</td>
<td>8.5/8.6</td>
<td>11.0/12.0</td>
</tr>
<tr>
<td>RSNet</td>
<td>20.5/31.1</td>
<td>12.8/18.7</td>
<td>19.2/29.5</td>
<td>13.1/12.8</td>
<td>15.7/21.8</td>
<td>41.1/50.2</td>
<td>18.5/23.4</td>
<td>13.9/50.9</td>
<td>6.4/16.1</td>
<td>8.4/16.1</td>
<td>4.3/35.0</td>
<td>20.2/26.1</td>
<td>16.8/23.2</td>
<td>4.0/5.4</td>
<td>4.8/3.1</td>
<td>26.0/31.3</td>
</tr>
<tr>
<td>SpiderCNN</td>
<td>22.3/25.8</td>
<td>7.2/6.7</td>
<td>17.7/19.1</td>
<td>4.1/2.7</td>
<td>2.7/4.0</td>
<td>5.5/6.5</td>
<td>5.9/18.9</td>
<td>7.7/10.5</td>
<td>0.0/0.4</td>
<td>30.0/28.4</td>
<td>22.4/34.7</td>
<td>14.5/15.0</td>
<td>4.9/5.3</td>
<td>0.0/1.7</td>
<td>23.9/30.2</td>
<td>8.5/8.9</td>
</tr>
<tr>
<td>PointConv</td>
<td>23.5/32.8</td>
<td>15.2/24.6</td>
<td>32.4/45.8</td>
<td>7.3/10.1</td>
<td>13.5/15.3</td>
<td>20.3/24.6</td>
<td>21.7/30.8</td>
<td>21.2/21.7</td>
<td>2.1/2.0</td>
<td>5.0/17.3</td>
<td>27.8/46.5</td>
<td>18.9/29.3</td>
<td>21.2/22.1</td>
<td>13.2/18.9</td>
<td>26.8/42.4</td>
<td>13.9/22.5</td>
</tr>
<tr>
<td>RSCNN</td>
<td>21.0/34.4</td>
<td>11.9/19.7</td>
<td>19.3/28.4</td>
<td>11.6/20.6</td>
<td>18.9/31.8</td>
<td>15.8/16.8</td>
<td>17.6/21.7</td>
<td>17.9/18.2</td>
<td>0.0/0.3</td>
<td>2.4/23.6</td>
<td>25.5/37.9</td>
<td>13.0/23.8</td>
<td>17.2/22.4</td>
<td>5.9/9.7</td>
<td>23.7/41.4</td>
<td>10.3/14.7</td>
</tr>
<tr>
<td>DGCNN</td>
<td>32.3/43.8</td>
<td>17.7/26.2</td>
<td>21.6/33.4</td>
<td>15.0/20.7</td>
<td>21.5/27.6</td>
<td>15.1/21.4</td>
<td>23.8/30.3</td>
<td>20.7/22.9</td>
<td>3.5/4.8</td>
<td>29.4/40.5</td>
<td>30.1/46.4</td>
<td>23.8/29.2</td>
<td>18.1/24.9</td>
<td>12.8/17.3</td>
<td>31.7/52.1</td>
<td>15.6/19.7</td>
</tr>
<tr>
<td>GraphCNN</td>
<td>22.9/25.9</td>
<td>12.5/14.9</td>
<td>0.7/0.8</td>
<td>1.8/5.2</td>
<td>0.2/0.3</td>
<td>0.3/1.0</td>
<td>1.2/4.0</td>
<td>1.6/1.8</td>
<td>0.2/0.2</td>
<td>0.0/0.0</td>
<td>0.5/0.5</td>
<td>0.3/0.5</td>
<td>0.6/0.6</td>
<td>0.6/0.6</td>
<td>0.4/0.4</td>
<td>24.2/34.4</td>
</tr>
<tr>
<td>Harris3D</td>
<td>0.4/0.4</td>
<td>1.0/1.0</td>
<td>0.5/0.5</td>
<td>0.9/0.6</td>
<td>1.9/0.8</td>
<td>1.0/0.8</td>
<td>0.2/0.2</td>
<td>0.6/0.6</td>
<td>0.2/0.2</td>
<td>0.2/0.2</td>
<td>0.3/0.3</td>
<td>0.3/0.3</td>
<td>0.5/0.5</td>
<td>0.5/0.5</td>
<td>0.5/0.5</td>
<td>0.5/0.5</td>
</tr>
<tr>
<td>SIFT3D</td>
<td>4.5/5.5</td>
<td>0.8/0.8</td>
<td>0.9/0.9</td>
<td>0.7/0.7</td>
<td>1.0/0.9</td>
<td>1.2/1.2</td>
<td>0.9/0.9</td>
<td>0.2/0.2</td>
<td>0.0/0.0</td>
<td>0.3/0.3</td>
<td>0.3/0.3</td>
<td>0.3/0.3</td>
<td>0.3/0.3</td>
<td>0.3/0.3</td>
<td>0.3/0.3</td>
<td>1.0/1.0</td>
</tr>
<tr>
<td>ISS3D</td>
<td>0.4/0.4</td>
<td>1.0/1.0</td>
<td>0.5/0.5</td>
<td>0.9/0.6</td>
<td>1.9/0.8</td>
<td>1.0/0.8</td>
<td>0.2/0.2</td>
<td>0.6/0.6</td>
<td>0.2/0.2</td>
<td>0.2/0.2</td>
<td>0.3/0.3</td>
<td>0.3/0.3</td>
<td>0.5/0.5</td>
<td>0.5/0.5</td>
<td>0.5/0.5</td>
<td>0.5/0.5</td>
</tr>
</tbody>
</table>

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

Figure 5. Visualizations of detected keypoints for six algorithms.
**Evaluation Metrics** Two metrics are adopted to evaluate the performance of keypoint saliency estimation. Firstly, we evaluate their mean Intersection over Unions (mIoU), which can be calculated as

\[
    IoU = \frac{TP}{TP + FP + FN}.
\]

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.

**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.

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

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

Figure 9. 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.

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.

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


