Pose2Room: Understanding 3D Scenes from Human Activities

Yinyu Nie, Angela Dai, Xiaoguang Han, and Matthias Nießner

Abstract

With wearable IMU sensors, one can estimate human poses from wearable devices without requiring visual input[1]. In this work, we pose the question: Can we reason about object structure in real-world environments solely from human trajectory information? Crucially, we observe that human motion and interactions tend to give strong informa tion about the objects in a scene –for instance a person sitting indicates the likely presence of a chair or sofa. To this end, we propose P2R-Net to learn a probabilistic 3D model of the objects in a scene characterized by their class categories and oriented 3D bounding boxes, based on an input observed human trajectory in the environment. P2R-Net models the probability distribution of object class as well as a deep Gaussian mixture model for object boxes, enabling sampling of multiple, diverse, likely modes of object configurations from an observed human trajec tory. In our experiments we show that P2R-Net can effectively learn multi-modal distributions of likely objects for human motions, and pro duce a variety of plausible object structures of the environment, even without any visual information. The results demonstrate that P2R-Net consistently outperforms the baselines on the PROX dataset and the VirtualHome platform.

Keywords：3D Scene Understanding, Shape-from-X, Probabilistic Model.

1 Introduction

In computer vision, we can effectively capture the geometric structure information of objects in the scene with strong visual information by using RGB, RGB-D images or a single image as input. We consider whether it is possible to infer the geometric structure information of objects in the scene if there is a lack of visual information. At present, wearable devices can estimate the posture of human body, use it to obtain the trajectory information of human body, and then infer the geometric structure information of objects, which is the idea of this paper.

The paper reproduction of this course intends to learn the relationship between scene interaction and geometric structure information of objects in the scene through the information of human motion trajectory, so as to estimate the category of interactive objects in the scene, as well as the direction and position of bounding box.

In this paper, non-visual information is used to understand the scene, and the results show that this method can effectively simulate the possible object configuration in the scene, and generate a reasonable object layout according to the input gesture trajectory. In qualitative and quantitative comparative experiments, the method in this paper is superior to the baseline method.

2 Related works

2.1 Predicting Human Interactions in Scenes.

Capturing and modeling interactions between human and scenes has seen impressive progress in recent years, following significant advances in 3D reconstruction and 3D deep learning. From a visual observation of a scene, interactions and human-object relations are estimated. Several methods have been proposed for understanding the relations between scene and human poses via object functionality prediction[2-5] and affordance analysis[6-11]. These methods explore human-scene interaction understanding by estimating object functionalities or human interactions as poses in a given 3D scene environment. In contrast, we take a converse perspective, and aim to estimate the 3D scene arrangement from human pose trajectory observations.

2.2 Scene Understanding with Human Priors.

As many environments, particularly indoor scenes, have been designed for people's daily usage, human behavioral priors can be leveraged to additionally reason about 2D or 3D scene observations.The recent approach of Mura et al.[12] poses the task of floor plan estimation from 2D human walk trajectories, and proposes to predict occupancy-based floor plans that indicate structure and object footprints, but without object instance distinction and employs a fully-deterministic prediction. To the best of our knowledge, we introduce the first method to learn 3D object arrangement distributions from human pose trajectories, without any visual input.

2.3 Pose Tracking with IMUs.

Our method takes the input of human pose trajectories, which is built on the success of motion tracking techniques. Seminal work on pose estimation from wearable sensors have demonstrated effective pose estimation from wearable sensors, such as optical markers[13-14] or IMUs[1,15-18]. Our work is motivated by the capability of reliably estimating human pose from these sensor setups without visual data, from which we aim to learn human-object interaction priors to estimate scene object configurations。

3 Method

3.1 Overview

From only a human pose trajectory as input, we aim to estimate a distribution of likely object configurations, from which we can sample plausible hypotheses of objects in the scene as sets of class category labels and oriented 3D bounding boxes. We observe that most human interactions in an environment are targeted towards specific objects, and that general motion behavior is often influenced by the object arrangement in the scene. We thus aim to discover potential objects that each pose may be interacting with.

We first extract meaningful features from the human pose sequence with a position encoder to disentangle each frame into a relative position encoding and a position-agnostic pose, as well as a pose encoder to learn the local spatiotemporal feature for each pose in consecutive frames. We then leverage these features to vote for a potential interacting object for each pose. From these votes, we learn a probabilistic mixture decoder to propose box proposals for each object, characterizing likely modes for objectness, class label, and box parameters. An illustration of our approach is shown in Figure 1.

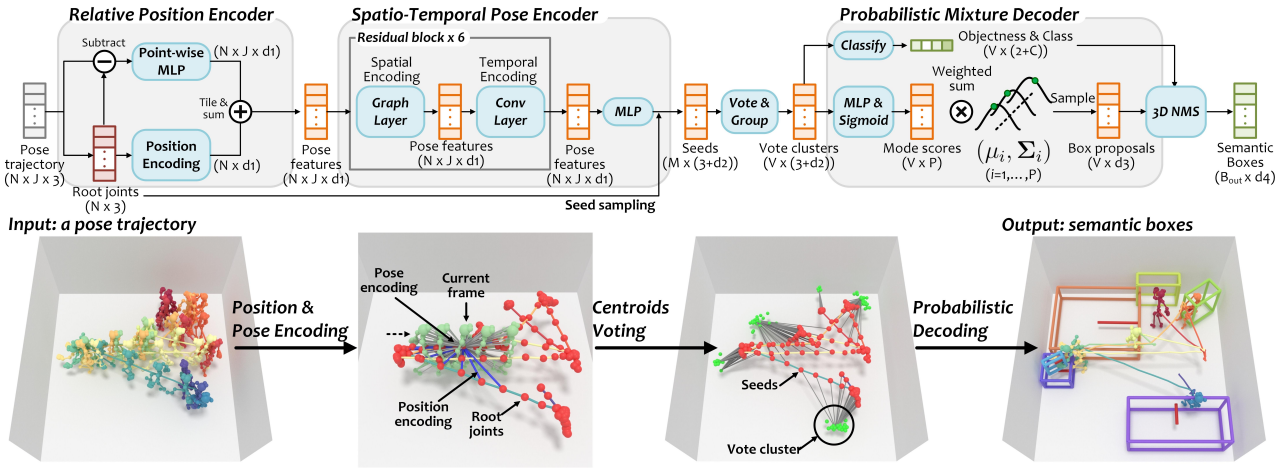


Figure 1: Overview of P2R-Net. Given a pose trajectory with N frames and J joints, a position encoder decouples each skeleton frame into a relative position encoding (from its root joint as the hip centroid) and a position-agnostic pose. After combining them, a pose encoder learns local pose features from both body joints per skeleton (spatial encoding) and their changes in consecutive frames (temporal encoding). Root joints as seeds are then used to vote for the center of a nearby object that each pose is potentially interacting with. A probabilistic mixture network learns likely object box distributions, from which object class labels and oriented 3D boxes can be sampled.

3.2 Feature extraction

Step 1：Relative Position Encoding

To learn informative pose features, we first disentangle for each frame the absolute pose joint coordinates into a relative position encoding $\boldsymbol{Q} \in \mathbb{R}^{N\times d_1}$ and a position-agnostic pose feature $\boldsymbol{P} \in \mathbb{R}^{N\times J\times d_1}$, which are formulated as:

$$\begin{aligned} \boldsymbol{Q} &= \text{Pool}\left[f_1(\mathcal{N}(\boldsymbol{r}) - \boldsymbol{r})\right], \\ \boldsymbol{P} &= f_2(\boldsymbol{T} - \boldsymbol{r}), \quad \mathcal{N}(\boldsymbol{r}) \in \mathbb{R}^{N\times k\times 3}, \end{aligned} \tag{1}$$

where $f_1(*), f_2(*)$ are point-wise MLP layers. $\mathcal{N}(\boldsymbol{r})$ is the set of k temporal neighbors to each root joint in r, and $Pool(*)$ denotes neighbor-wise average, and $Pool(*)$ denotes neighbor-wise average pooling. By broadcast summation, we output $\boldsymbol{P}^r = \boldsymbol{P} + \boldsymbol{Q}$ for further spatio temporal pose encoding.

Step 2：Spatio-Temporal Pose Encoding

We first use a graph convolution layer to learn intra-skeleton joint features. Edges in the graph convolution are constructed following the skeleton bones, which encodes skeleton-wise spatial information. For each joint, we then use a 1-D convolution layer to capture temporal features from its inter-frame neighbors.A graph layer and an 1-D convolution layer are linked into a block with a residual connection to process the input $\boldsymbol{P}^r$ (see Fig. 2). By stacking six blocks, we obtain a deeper spatio-temporal pose encoder with a wider receptive field in temporal domain, enabling reasoning over more temporal neighbors for object box estimation. Finally, we adopt an MLP to process all joints per skeleton to obtain pose features $\boldsymbol{P}^{st} \in \mathbb{R}^{N\times d_2}$.

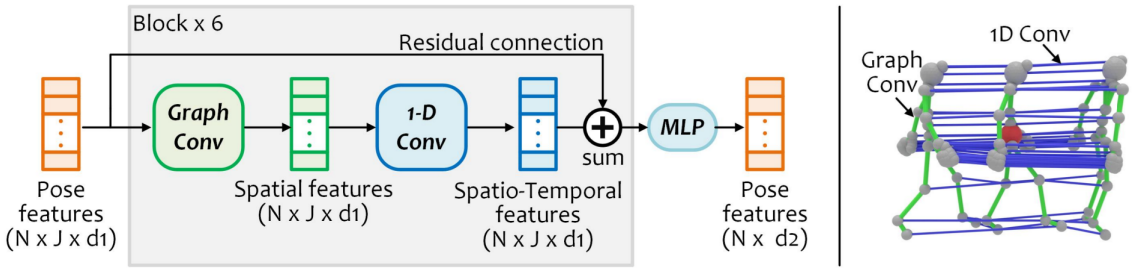


Figure 2: Pose encoding with spatio-temporal convolutions.

Step 3：Locality-Sensitive Voting

For each pose feature $p^{st} \in \boldsymbol{P}^{st}$, we use its root joint $r \in \boldsymbol{r}$ as a seed location, and vote for an object center by learning the displacement from the seed:

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{r}_s + f_3\left(\boldsymbol{P}_s^{st}\right), \quad \boldsymbol{r}_s, \boldsymbol{v} \in \mathbb{R}^{M\times 3}, \\ \boldsymbol{P}^v &= \boldsymbol{P}_s^{st} + f_4\left(\boldsymbol{P}_s^{st}\right), \quad \boldsymbol{P}_s^{st}, \boldsymbol{P}^v \in \mathbb{R}^{M\times d_2}, \end{aligned} \tag{2}$$

where $\boldsymbol{r}_s$ are the evenly sampled M seeds from $\boldsymbol{r}$; P_s^{st} are the corresponding pose features of $\boldsymbol{r}_s$; f_3, f_4 are MLP layers; $\boldsymbol{r}$, $\boldsymbol{P}^v$ denote the vote coordinates and features learned from $\boldsymbol{P}_s^{st}$. We evenly sample seeds $\boldsymbol{r}_s$ from to make them cover the whole trajectory and adaptive to sequences with different length.

Since there are several objects in a scene, for each seed in $\boldsymbol{r}_s$, we vote for the center to the nearest one. For the seeds which vote for the same object, we group their votes to a cluster. This outputs cluster centers $\boldsymbol{v}^c \in \mathbb{R}^{V\times 3}$ with aggregated cluster feature $\boldsymbol{P}^c \in \mathbb{R}^{V\times d_2}$ where V denotes the number of vote clusters. We then use the $\boldsymbol{P}^c$ to decode to distributions that characterize semantic 3D boxes, which is described in Step 4.

Step 4：Probabilistic Mixture Decoder

We decode vote clusters $(\boldsymbol{v}^c, \boldsymbol{P}^c)$ to propose oriented 3D bounding boxes for each object, along with their class label and objectness score. Each box is represented by a 3D center c, 3D size s and 1D orientation θ, where we represent the size by $log(s)$ and orientation by $(sin(\theta), cos(\theta))$ for regression. We propose to learn a probabilistic mixture decoder to predict the box centroid, size and orientation with multiple modes, from a vote cluster $v^c \in \boldsymbol{v}^c$, $P^c \in \boldsymbol{P}^c$:

$$\begin{aligned} y_\tau &= \textstyle\sum_{k=1}^{P} f_\tau^k\left(P^c\right) \cdot y_\tau^k, \quad \tau \in \{c, s, \theta\}, \\ y_c^* &= v^c + y_c, \quad y_\tau^k \sim \mathcal{N}\left(\mu_\tau^k, \textstyle\sum_\tau^k\right), \quad y_\tau, y_\tau^k \in \mathbb{R}^{d_\tau}, \end{aligned} \tag{3}$$

where $\tau \in \{c, s, \theta\}$ denote the regression targets for center, size, and orientation; $\mathcal{N}\left(\mu_\tau^k, \sum_\tau^k\right)$ is the learned multivariate Gaussian distribution of the k-th mode for τ, where y_τ^k is sampled from; P is the number of Gaussian distributions (i.e., modes); $f_\tau^k(*) \in [0, 1]$ is the learned score for the k-th mode; y_τ is the weighted sum of the samples from all modes, which is the prediction of the center/size/orientation; and d_τ is their output dimension (d_c=3, d_s=3, d_θ=2). Note that the box center y_c^* is obtained by regressing the offset y_c from cluster center v^c . We predict the proposal objectness and the probability distribution for class category directly from P^c , using an MLP.

3.3 Loss

The loss consists of classification losses for objectness $\mathcal{L}_{obj}$ and class label $\mathcal{L}_{cls}$, and regression losses for votes $\mathcal{L}_v$, box center $\mathcal{L}_c$, size Ls and orientation $\mathcal{L}_\theta$.

Classification Losses. $\mathcal{L}_{obj}$ and $\mathcal{L}_{cls}$ are supervised by cross entropy losses, wherein the objectness score is used to classify if a vote cluster center is close to ($\leqslant$ 0.3 m, positive) or far from ($\geqslant$ 0.6 m, negative) the ground truth. Proposals from the clusters with positive objectness are further supervised with box regression losses.

Regression Losses. We supervise all the predicted votes, box centers, sizes and orientations with a Huber loss. For poses that are located within d_p to objects (d_p = 1 m), we use the closest object center to supervise their vote. Votes from those poses that are far from all objects are not considered. For center predic tions, we use their nearest ground-truth center to calculate $\mathcal{L}_c$. Since box sizes and orientations are predicted from vote clusters, we use the counterpart from the ground-truth box that is nearest to the vote cluster for supervision. Then the final loss function is $\mathcal{L} = \sum_\tau \lambda_\tau \mathcal{L}_\tau$, where $\mathcal{L}_\tau \in \{\mathcal{L}_{obj}, \mathcal{L}_{cls}, \mathcal{L}_v, \mathcal{L}_c, \mathcal{L}_s, \mathcal{L}_\theta\}$ and $\{\lambda_\tau\}$ are constant weights that balance the losses.

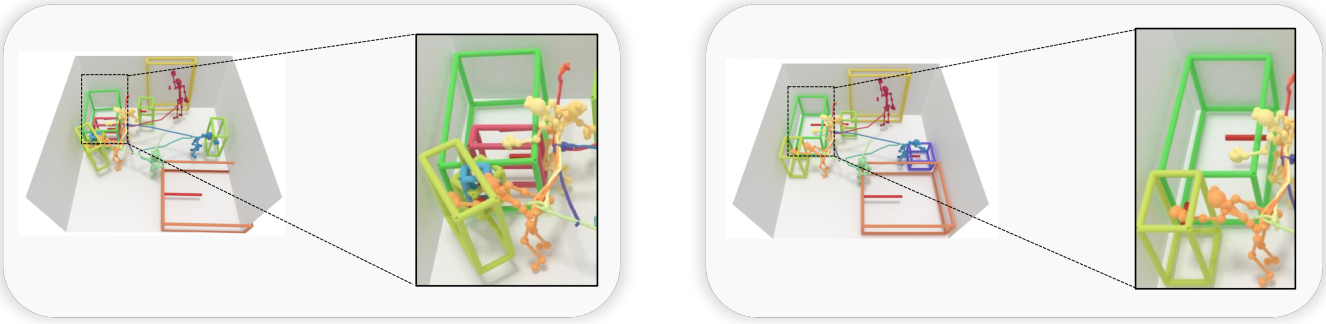
4 Implementation details

4.1 Comparing with released source codes

The source code of the paper is available. Based on the source code, I have improved the predicted results of P2R-Net in some aspects. It is found that it tends to predict extra objects after observing the failed experimental results of the paper. Three cases of extra objects are introduced as follows:

Case 1: Contained in a large object

These large objects are embedded with small objects, such as wardrobes and desks with drawers. If people interact with these small objects, they will be explicitly predicted separately. But the ground truth is that they are contained in large objects and don't need to be predicted as separate objects.



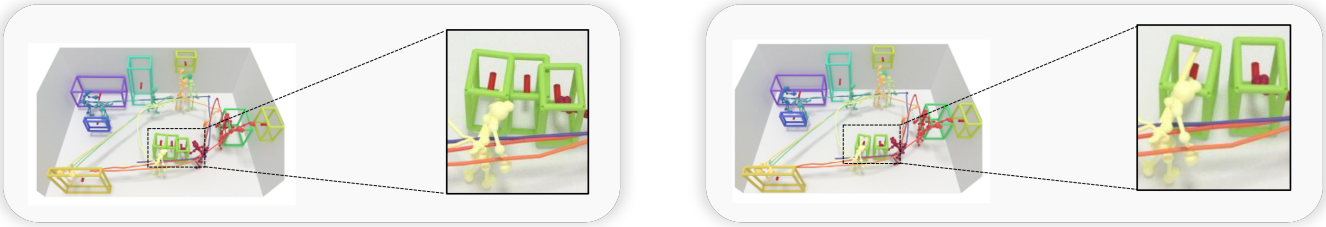
Ours Ground Truth

Figure 3: Contained in a large object.

Case 2: Intersect with other objects

Since it is impossible to determine in advance how many objects are in the room, the existence of each object is determined by its own probability, instead of taking the highest value for the existence probability of several objects. In order to improve the diversity of the prediction results, this paper uses the gaussian mixture

model to predict the probability of the existence of objects where there is interaction, so it may leave a variety of prediction results.



Ours Ground Truth

Figure 4: Intersect with other objects.

Case 3: Not intersecting with other objects

Because human's random trajectory falsely suggests that there may be objects. This object is recognized as an obstacle to movement. Therefore, some extra objects are predicted.

Ours Ground Truth

Figure 5: Not intersecting with other objects.

We will improve the two failed cases of case 1 and case 2 in the prediction results, so as to reduce the probability of predicting extra objects. Finally, we will visualize the improved results.

4.2 Main contributions

Restatement of the problem. This paper predicts the bounding box (OBB) of objects in the room according to the motion trajectory. As a result, there are two failure cases in the figure. One is to predict that an extra OBB will be included in the large OBB. The other predicted that an extra OBB would intersect with the other OBB. This work will improve these two cases and reduce the occurrence of failures.

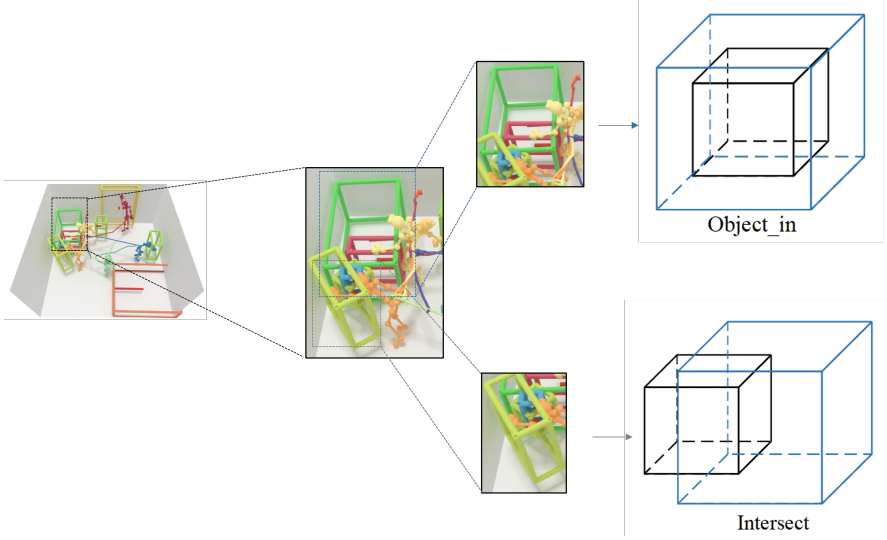


Figure 6: Restatement of the problem.

We will first find the adjacent objects for each object in the prediction results. Because it can reduce the number of objects that need to judge the intersection type. Then, getting the intersection types of these adjacent objects. There are three possible types including contained, intersected and disjoint. Finally, according to the type of intersection, select the deleted object.

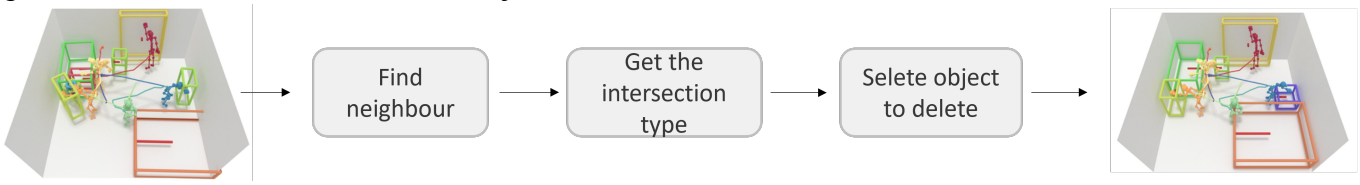


Figure 7: The pipeline of my improvement.

Step 1: Find neighbour.

For the OBB of the predicted object, we can get the center, size and orientation. Our goal is to determine whether the OBBs in each room intersect and the type of intersection. In order to reduce the subsequent calculation, we only judge the types of intersection for the adjacent OBBs.

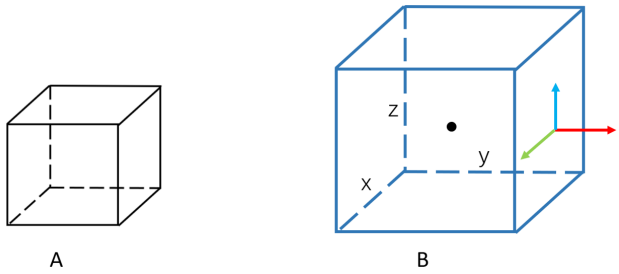


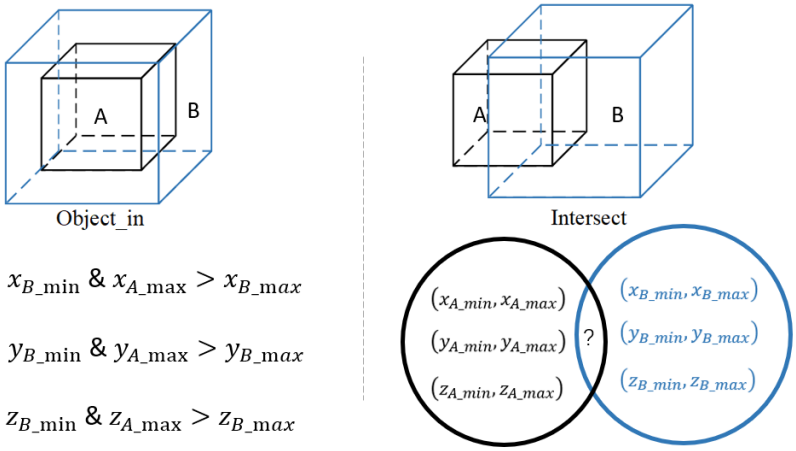
Figure 8: Definition of adjacent OBBs.

Assuming that two objects in the room are A and B, respectively, we calculate the distance between their center points and screen out the adjacent objects whose distance is less than a certain threshold:

$$|center_A - center_B| < threshold. \tag{4}$$

Step 2: Get the intersection type.

Given the center and size of the OBB, we can calculate the minimum and maximum values of eight vertices in X, Y and Z directions. Then, we can judge the type of intersection by comparing the maximum range of coordinates. As shown in Figure 1, if the coordinate range of object A is between the coordinate ranges of object B, then A is inside B. If their coordinate ranges intersect, then A and B intersect.



$$x_{A\_min} < x_{B\_\min} \;\&\; x_{A\_\max} > x_{B\_max}$$

$$y_{A\_min} < y_{B\_\min} \;\&\; y_{A\_\max} > y_{B\_max}$$

$$z_{A\_min} < z_{B\_\min} \;\&\; z_{A\_\max} > z_{B\_max}$$

Figure 9: Get the intersection type.

Because the angles of intersection of objects are various, the orientation of objects needs to be considered. If the coordinates of the eight vertices are changed in orientation are changed, it is necessary to compare the changed results to get the maximum value of the coordinates again. We introduce the diagonal vector of OBB. Its starting point is the minimum value of OBB, and its ending point is the maximum value of OBB. After improvement, the coordinate range of OBB can be obtained by changing the direction of a vector.

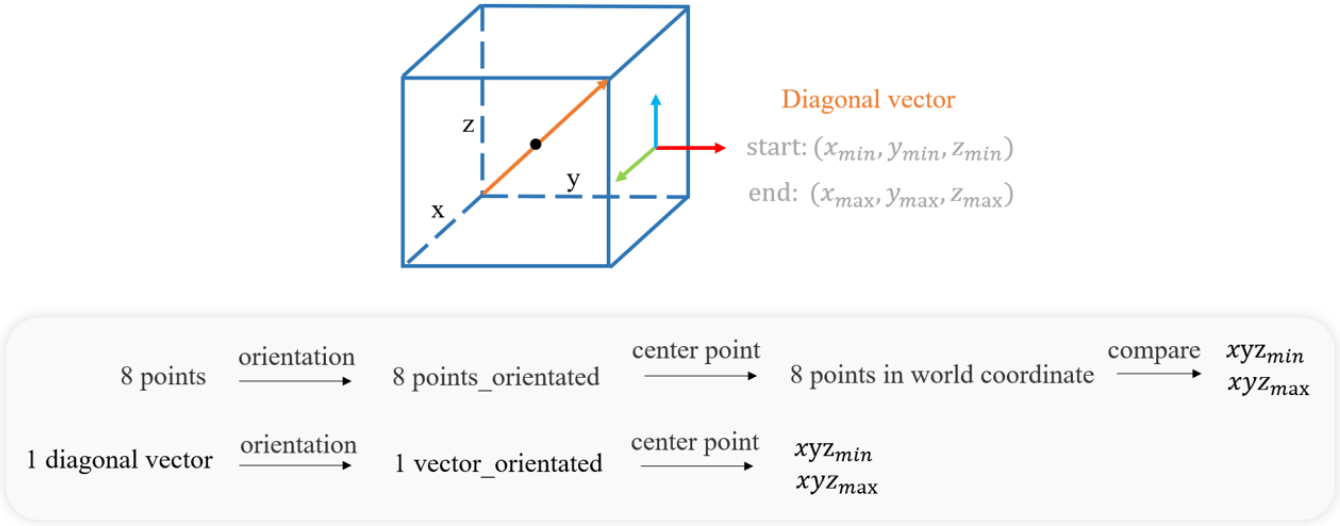


Figure 10: The significance of diagonal vector.

Step 3: Select object to delete.

For small objects contained in large objects, choose to delete them; for intersecting objects, choose to leave the objects with high probability of existence and delete the objects with low probability of existence.

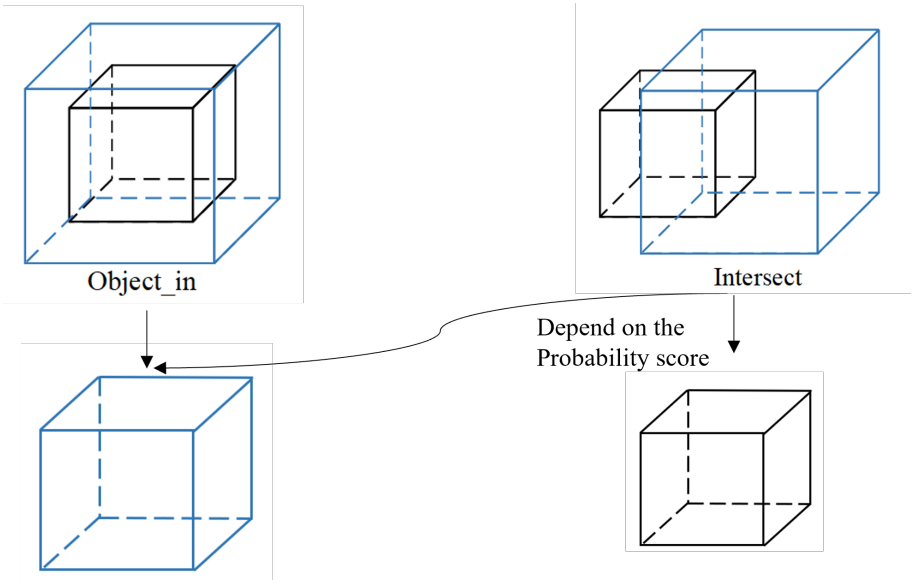


Figure 11: Select object to delete.

4.3 Experimental environment setup

We train P2R-Net end-to-end from scratch with the batch size at 32 on 4 NVIDIA TITAN XP GPUs for 180 epochs, where Adam is used as the optimizer. The initial learning rate is at 1e-3 in the first 80 epochs, which is decayed by 0.1× every 40 epochs after that. The losses are weighted by $\tau_{obj} = 5$, $\tau_{cls} = 1$, $\{\tau_v, \tau_c, \tau_s, \tau_\theta\} = 10$ to balance the loss values. During training, we use pose distance threshold $t_d = 1m$. At inference time, we output box predictions after 3D NMS with an IoU threshold of 0.1. We use an objectness threshold of $t_o = 0.5$.

4.4 Interface design

Visualization Toolkit (VTK) is used to visualize the 3D scene. It is a GUI window and you can interact with the scene.

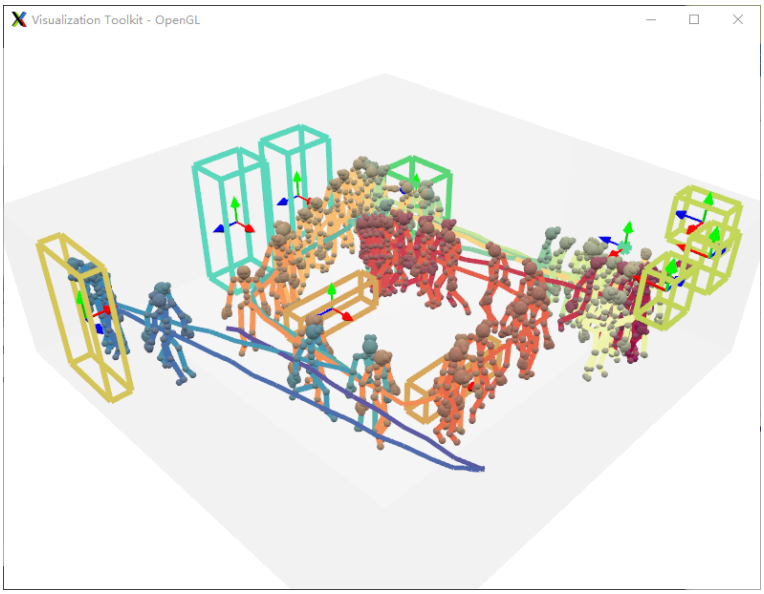


Figure 12: An example of VTK.

5 Results and analysis

We evaluate our method for the case that additional objects are included or intersect with other objects in the prediction results. In order to compare with the results of the paper, we test the pre-training model of the paper, and post-process the test results with our improved method.

Figure 13 visualizes the improvement of the intersection with other objects. Two similar OBBs (blue boxes) are reserved in the prediction results before improvement. After improvement, one OBB is deleted and the OBB with high probability of existence is reserved. The result is consistent with the Ground Truth.

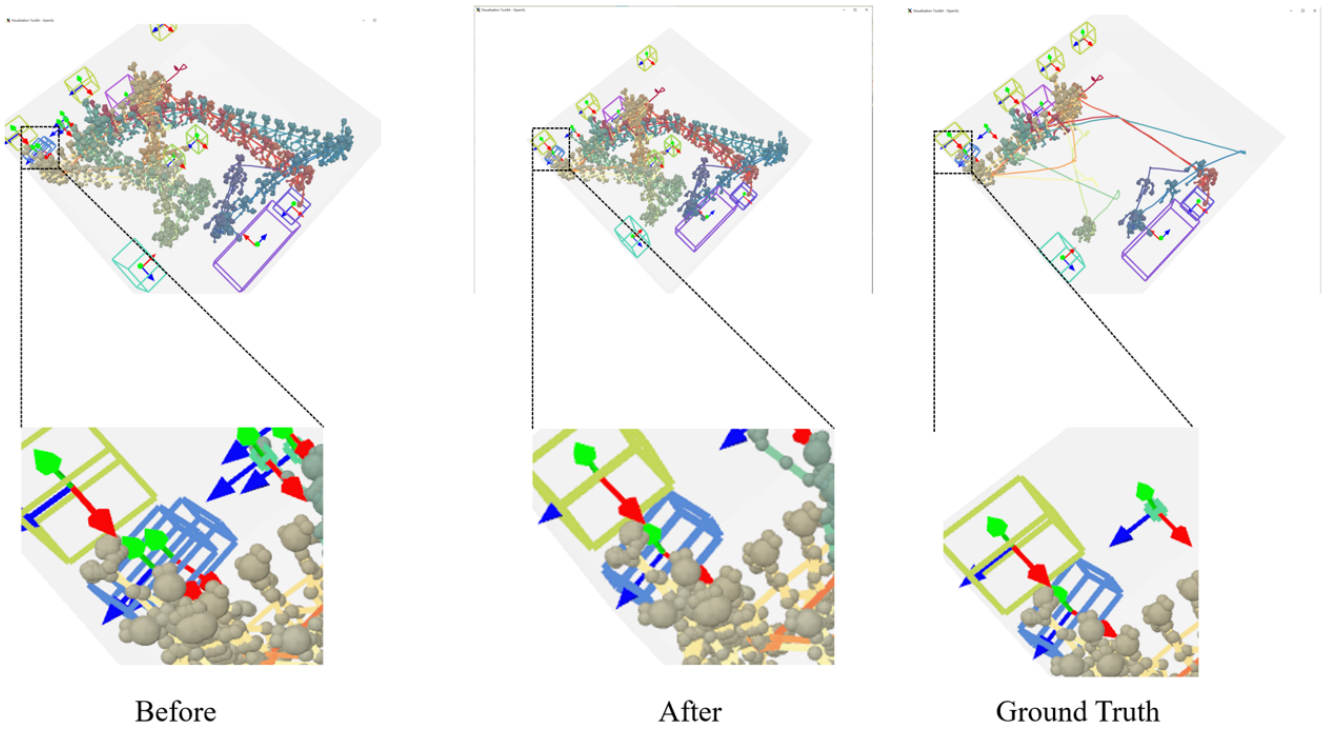


Figure 13: The improvement of the intersection with other objects.

Figure 14 visualizes the improvement of the inclusion of objects. The large object (green box) in the prediction results before improvement include small object (red box). After improvement, the small OBB is deleted and the large OBB is reserved, and the result is consistent with the Ground Truth.

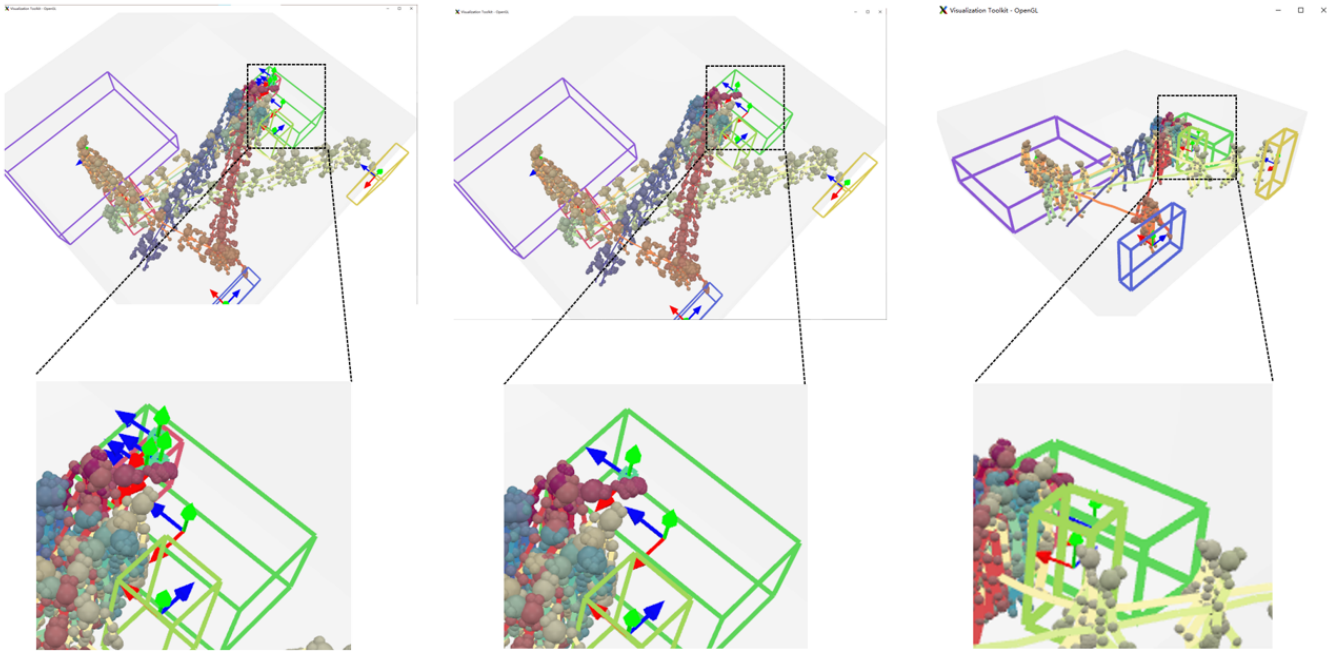


Figure 14: The improvement of the inclusion of objects.

Figure 15 visualizes an example of failure cases. If the distance between the center points of two OBBs is too large, it is not within the scope of this improvement. Because, in order to reduce the amount of calculation, the judgment of finding adjacent objects is based on the fact that their center points are less than a certain threshold (see Step 1 in Section 4.2).

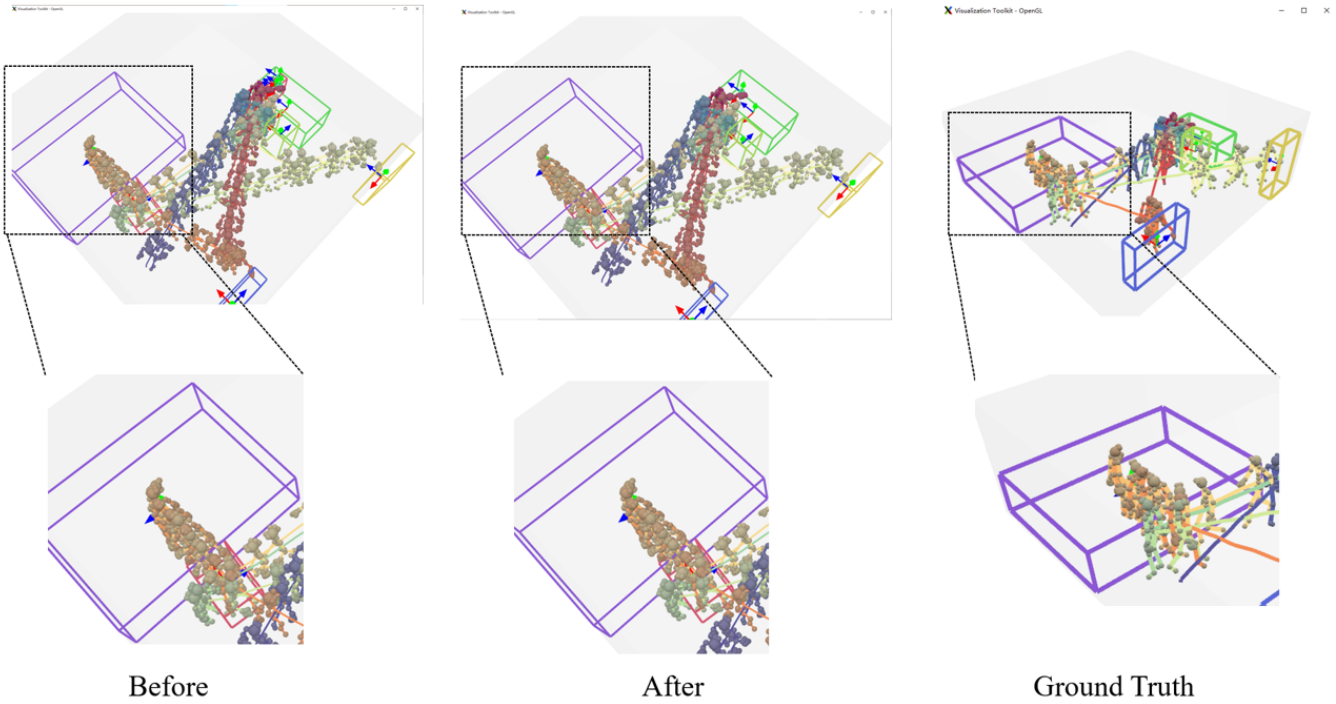


Figure 15: An example of failure cases.

6 Conclusion and future work

In this paper, given a motion trajectory, the center, size and orientation of the OBBs of objects in a room are predicted. The failure of prediction results is usually due to over-prediction of objects. We have improved two kinds of failures, including the intersection of multi-predicted objects or being contained in objects. The improved method first finds the neighbors of all objects, then gets the intersection type, and finally chooses to delete the extra predicted objects according to different situations. However, not all intersecting objects can be handled smoothly, and the failure situation is described in Section 5. In the future, we can consider improving the method of screening intersecting objects, so as to reduce the amount of calculation and ensure that all intersecting objects can be detected.

References

[1] Von MARCARD T, ROSENHAHN B, BLACK M, et al. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs[C]//Computer Graphics Forum: vol. 36: 2. 2017: 349-360.

[2] GRABNER H, GALL J, VAN GOOL L. What makes a chair a chair?[C]//CVPR 2011. 2011: 1529-1536.

[3] ZHU Y, ZHAO Y, CHUN ZHU S. Understanding tools: Task-oriented object modeling, learning and recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 2855-2864.

[4] PIEROPAN A, EK C H, KJELLSTRÖM H. Functional object descriptors for human activity modeling [C]//2013 IEEE International Conference on Robotics and Automation. 2013: 1282-1289.

[5] HU R, van KAICK O, WU B, et al. Learning how objects function via co-analysis of interactions[J]. ACM Transactions on Graphics (TOG), 2016, 35(4): 1-13.

[6] GUPTA A, SATKIN S, EFROS A A, et al. From 3d scene geometry to human workspace[C]//CVPR 2011. 2011: 1961-1968.

[7] SAVVA M, CHANG A X, HANRAHAN P, et al. Pigraphs: learning interaction snapshots from observations[J]. ACM Transactions on Graphics (TOG), 2016, 35(4): 1-12.

[8] SAWATZKY J, SRIKANTHA A, GALL J. Weakly supervised affordance detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 2795-2804.

[9] WANG Z, CHEN L, RATHORE S, et al. Geometric pose affordance: 3d human pose with scene constraints[J]. arXiv preprint arXiv:1905.07718, 2019.

[10] DENG S, XU X, WU C, et al. 3d affordancenet: A benchmark for visual object affordance understanding [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 1778-1787.

[11] RUIZ E, MAYOL-CUEVAS W. Where can i do this? Geometric affordances from a single example with the interaction tensor[C]//2018 IEEE International Conference on Robotics and Automation (ICRA). 2018: 2192-2199.

[12] MURA C, PAJAROLA R, SCHINDLER K, et al. Walk2Map: Extracting Floor Plans from Indoor Walk Trajectories[C]//Computer Graphics Forum: vol. 40: 2. 2021: 375-388.

[13] CHAI J, HODGINS J K. Performance animation from low-dimensional control signals[G]//ACM SIGGRAPH 2005 Papers. 2005: 686-696.

[14] HASSAN M, CEYLAN D, VILLEGAS R, et al. Stochastic scene-aware motion prediction[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 11374-11384.

[15] LIU H, WEI X, CHAI J, et al. Realtime human motion control with a small number of inertial sensors [C]//Symposium on interactive 3D graphics and games. 2011: 133-140.

[16] HUANG Y, KAUFMANN M, AKSAN E, et al. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time[J]. ACM Transactions on Graphics (TOG), 2018, 37(6): 1-15.

[17] KAUFMANN M, ZHAO Y, TANG C, et al. Em-pose: 3d human pose estimation from sparse electromagnetic trackers[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 11510-11520.

[18] GLAUSER O, WU S, PANOZZO D, et al. Interactive hand pose estimation using a stretch-sensing soft glove[J]. ACM Transactions on Graphics (ToG), 2019, 38(4): 1-15.