Indoor Objects Rearrangement via Semantic Segmentation

이름: 박수완
학번: 20150805
연구지도교수: 이승용

---

1 Problem Statement

Rearranging indoor scene objects can be achieved by utilizing two fields of research, semantic segmentation and scene augmentation. The former enables object-level extracting from a scene and the later applies resulted 3D models to complete a new scene that is totally different from original one. This task can also be performed with scene filmed using smartphone LiDAR sensor, which implies that development of application, easy to use in everyday life, is feasible.

2 Theoretical Background

2.1 3D Scene Semantic Segmentation

Semantic segmentation methodology of 3D scene is divided into one of directly handling 3D data and the other of indirectly merging results of 2D image segmentation generated by projecting the scene[1]. Two methods both have pros and cons but the former is widely being adopted because its procedure of process is simpler. Research of Liu et al.[2] performs State-Of-The-Art level for semantic segmentation on 3D indoor scene.

2.2 Open3D

Open3D is python package which provides large number of functions for processing 3D data, which includes point cloud clustering, normal vector estimation, and conversion to 3D meshes or other several types of data.

2.3 S3DIS Dataset

S3DIS dataset is composed two hundred seventy indoor 3D scene data in total. These scenes are classified in ten classes, from office and conference room to storage and pantry, and it distinguishes objects into thirteen types including floor, chair, bookcase, and so on.

2.4 Indoor Scene Arrangement

Several researches about object arrangement in indoor scene are progressed. For examples, one[3] focuses on building fluent knowledge models about positional relationships between indoor objects in order to inference each object's optimal position and orientation and places them one by one. Another[4] uses tree graph which represents reliance relationships and determines all objects' arrangement at once.

2.5 Smartphone LiDAR

LiDAR sensor attached to smartphone or tablet PC is relatively small yet has no shortage of performance, supported by several compatible indoor scanning applications[5, 6]. Using them allows to acquire 3D model of specific indoor scene, which is able to be processed with semantic segmentation.

3 Design and Implementation

3.1 PVCNN Model Modification

PVCNN is selected for semantic segmentation network but its structure initially accepted only labeled S3DIS dataset, which leads to modify several code lines appropriately to take custom data filmed using LiDAR sensor also as an input. Also, PVCNN does not print any prediction results, thus extorting values of specific network cells at appropriate training cycle. S3DIS dataset consists of point cloud data, therefore point-level prediction value array should be acquired.

3.2 Object-Level Extraction

Filtering points using prediction value enables to acquire objects set of specific object type and clustering points with proper neighborhood radius allows to extract object level 3D model data. Implementation details are described in [**Algorithm 1**]

Algorithm 1 Object-level extraction

```
Load scene and prediction data
for type in range num_type do
    points = Extract all points whose prediction value is type
    labels = Cluster points                                  ▷ len(labels) == len(points)
    for label in labels do
        object = Extract all points whose label value is label
        if len(object) ≤ min_points then
            continue
        end if
        Determine object is outer or inner
        Translate object to (0, 0)
        if object is outer then
            Rotate object to orient +x direction
        end if
        Save object
    end for
end for
```

3.3 Empty Room Acquisition

Removing objects which will be rearranged simply makes the room empty but it is not sufficient because unpredictable points from PVCNN and dismissed points from clustering procedure are remain in midair in the scene. These points also should be deleted by extra filtering. In addition, bounding box representation of empty room can help better understand of rearranged objects because semantic segmentation does now show perfect performance. Implementation details are described in [**Algorithm 2**]

Algorithm 2 Empty room acquisition

Load scene and prediction data
for *type* in range *num_type* **do**
  if *type* == 3 **then** ▷ Wall
    Extract x, y components of bounding box
  else if *type* == 1||*type* == 2 **then** ▷ Ceiling, Floor
    Extract max, min of z component of bounding box, relatively
  end if
  if *type* is in types_to_be_removed **then** ▷ Table, Chair, Sofa, Shelf, and Board
    Delete all points whose prediction value is *type* from scene
  end if
end for
Draw bounding box
labels = Cluster remained points ▷ $len(labels) == len(points)$
for *label* in *labels* **do**
  if *label* == −1 **then**
    delete corresponding point from scene
  end if
end for
Save empty scene

3.4 Placement of Extracted Objects

Objects can be considered as either outer or inner object and placement proceeds from outer to inner one. Outer object should be aligned with walls and its orientation is depended on which side of wall that the objects is aligned. Meanwhile, inner object can have random position and orientation, however, it tries taking multiple places and chooses the highest-scored one resulted from knowledge model function. Implementation details and template of knowledge model function are described in [**Algorithm 3, 4**], relatively.

Algorithm 3 Placement of extracted objects

Load empty scene and bounding box
objects = Load objects from outer to inner ones
room = Prepare 2D-array by projecting scene from +z direction
for *object* in *objects* **do**
  if *object* is outer **then**
    boundaries = Calculate available boundary via shape of *object* from *room*
    Randomly select *boundary* from *boundaries*
    Calculate exact position and orientation from relation between *boundary* and *object*
    Randomly select offset not to make *object* invade over *boundary*
  else
    Calculate available area that *object* can be located
    for *iter* in num_try **do**
      Randomly select temporary position of *object*
      Calculate score based on knowledge model function
        ▷ Only function between table and chair is defined in this research
      Choose highest-scored try as position of *object* and randomly select orientation
    end for
  end if
  Rotate and Translate *object*
  Update *room*
end for

Algorithm 4 Template of knowledge model function

```
Input variables of x, y, w, h, room, pluses, minuses
                                  ▷ pluses contains types usually be located closer to current object type
                                  ▷ minuses contains types usually be located farther to current object type
Set score = 0
for i in range of (x − w, x + w) do
    for j in range of (y − h, y + h) do
        if room[i][j] is in pluses then
            if (i, j) overlaps with current object then
                score −= some weight
            else
                score += some weight
            end if
        else if room[i][j] is in minuses then
            score −= some weight
        end if
    end for
end for
```

Knowledge model calculates score based on objects' positions but does not consider about orientation due to limitation of detecting original direction.

4 Results and Evaluation

4.1 PVCNN Model Modification

Modified PVCNN can now accept any unlabeled, space delimited, and xy-plane aligned xyzrgb data and prints point-level prediction value array after one cycle of scene evaluation. [**Figure 1**] is an example of visualization of semantic segmentation result.

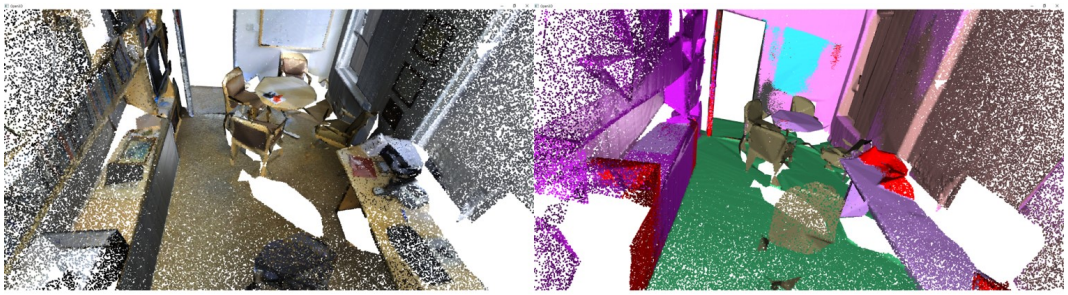

Figure 1: Example of segmentation visualization

4.2 Object-Level Extraction

[**Figure 2**] shows an example of object-level extraction. In process of clustering, some clusters which have a few number of points are dismissed even though they have corresponding valid prediction value. Also, Some other clusters can comprise more than one object because the objects are located too closely.

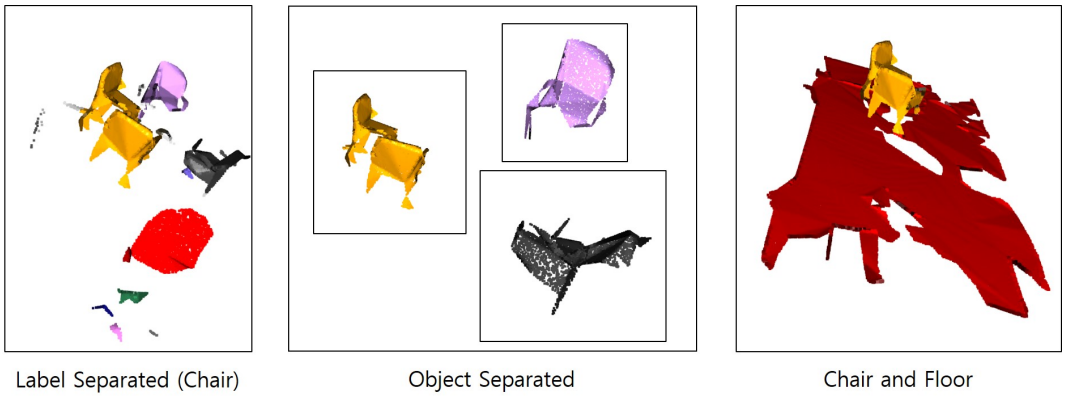

Figure 2: Example of object-level extraction

4.3 Empty Room Acquisition

An example of empty room acquisition is shown in [**Figure 3**] and it can be found that empty scene has several blanks because there is no points at corresponding position. Large one of them can confuse boundaries of the scene but bounding box does resolve this issue.

Figure 3: Example of empty room acquisition

4.4 Placement of Extracted Objects

[**Figure 4-7**] shows examples of scene rearrangement. Sub-figure (a) displays original scene without ceiling, (b) displays voxelized version of rearranged objects with bounding box, (c) displays voxelized version of rearranged objects with empty scene, (d) displays rearranged scene for each figure, and (e) and (f) display (d) in different angle.

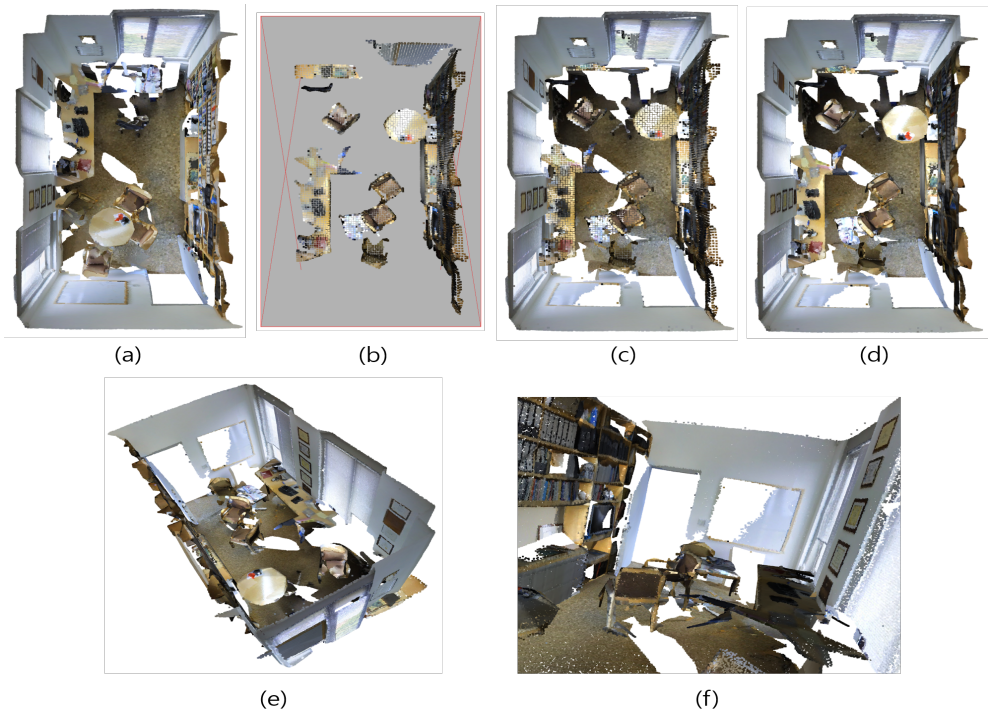

Figure 4: Rearrangement example of S3DIS dataset, Area 5, Office 1 - Case 1. (a) for original scene, (b) for voxelized rearranged objects with bounding box, (c) for voxelized rearranged objects with empty scene, (d) for rearranged scene, and (e) and (f) for different angles of (d).

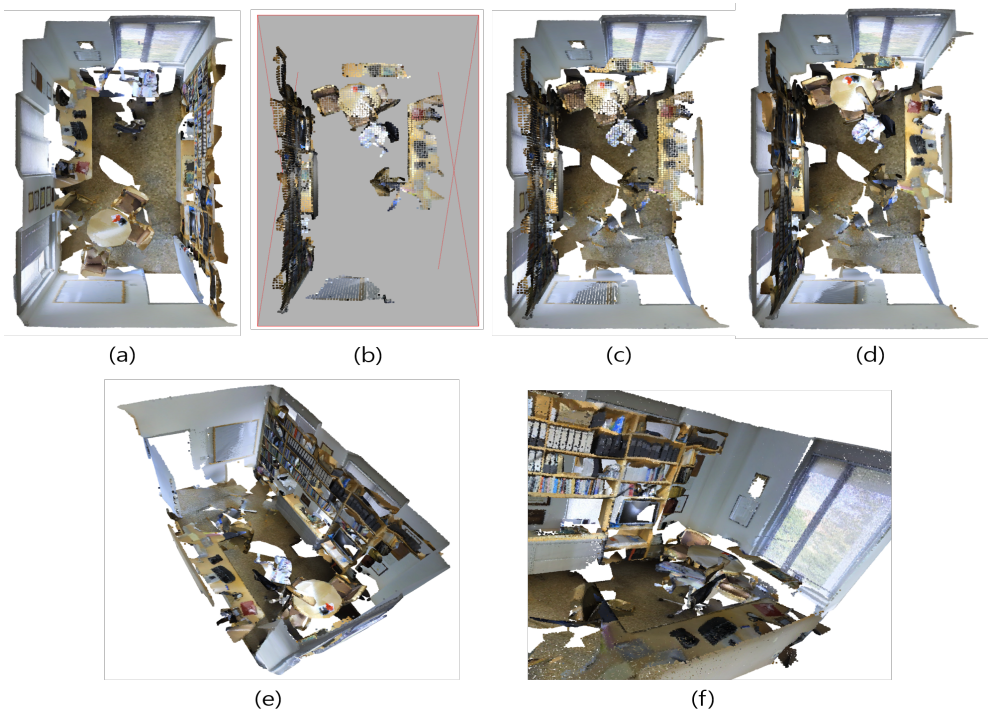

Figure 5: Rearrangement example of S3DIS dataset, Area 5, Office 1 - Case 2. (a) for original scene, (b) for voxelized rearranged objects with bounding box, (c) for voxelized rearranged objects with empty scene, (d) for rearranged scene, and (e) and (f) for different angles of (d).

Due to randomness of choosing position and orientation, every execution generates different result. In that sense, [**Figure 4**] and [**Figure 5**] show different arrangement each other on the same scene. It is possible to check that objects which were originally aligned with wall are also attached to boundary in rearranged scene. However, new positions of inside objects, specifically chairs, are quite awkward because only object position is affected by knowledge model while orientation is not.

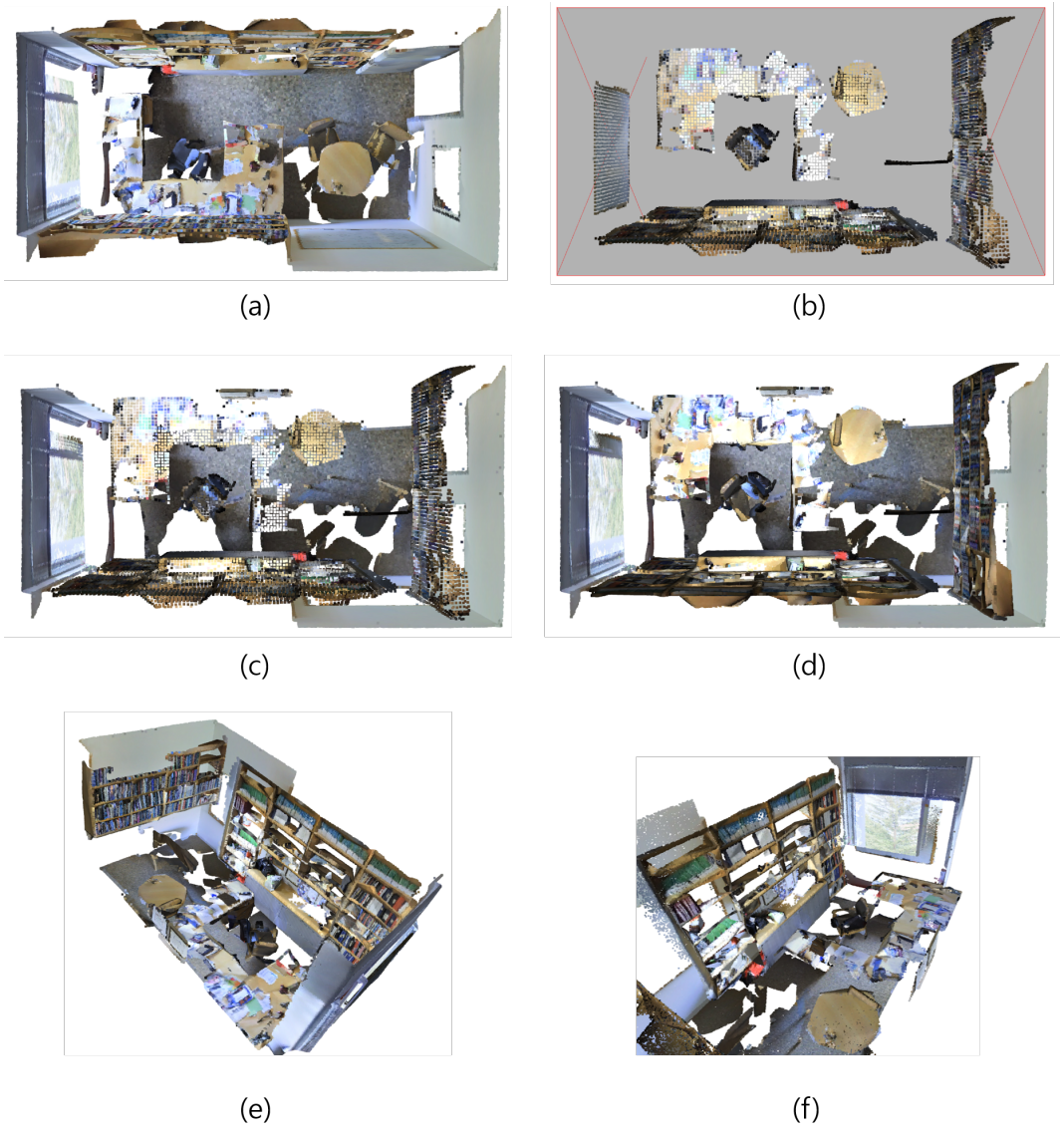

Figure 6: Rearrangement example of S3DIS dataset, Area 5, Office 3. (a) for original scene, (b) for voxelized rearranged objects with bounding box, (c) for voxelized rearranged objects with empty scene, (d) for rearranged scene, and (e) and (f) for different angles of (d).

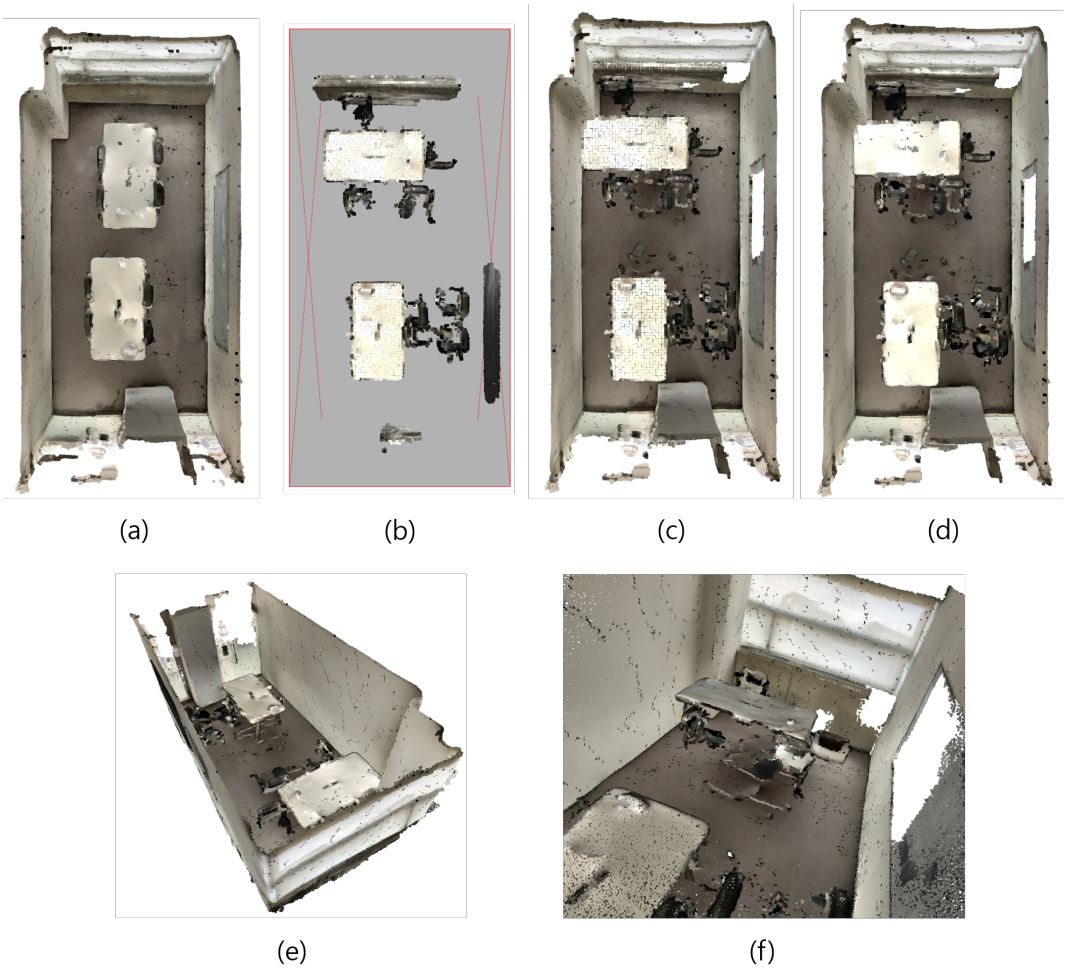

Figure 7: Rearrangement example of Self-made data, Tae-Joon Park Library, 4F GSR-J. (a) for original scene, (b) for voxelized rearranged objects with bounding box, (c) for voxelized rearranged objects with empty scene, (d) for rearranged scene, and (e) and (f) for different angles of (d).

[**Figure 7**] shows rearrangement of self-made data. It is one of group study rooms in POSTECH's Tae-Joon Park library and filmed using LiDAR sensor of iPad Pro 5th Generation. It demonstrates that modified PVCNN, object-level extraction, and rearrangement algorithms also work well on custom, unlabeled dataset.

5 Discussion and Future Work

This research tried to utilize 3D semantic segmentation and indoor scene augmentation. As a result, indoor objects were successfully rearranged even in the case of self-made data. However, there are several points that can be further developed.

5.1 Inner object orientation detection

In scene augmentation researches[3][4], every objects' orientation is known because 3D models of each object are given. However, it is not possible to detect object orientation with only semantic segmentation. If there is an effective way to identify direction of objects, rearrangement quality will be increased. For example, chair will get higher score when directing desk.

5.2 More sophisticated knowledge model

Knowledge model of this research is too simple. It was not possible to acquire elaborated knowledge model built in other researches because it was not open in public. Building more sophisticated knowledge model or getting other researches' one will upgrade overall level of rearrangement.

5.3 Other types of 3D model representation

This research only used normal or voxelized point cloud. Polygonal and triangle meshes are also tried but their visual quality was not sufficient because additional blanks were also generated with executing conversion algorithm. Applying reconstruction algorithm can resolve this issue.

5.4 Supporting variant room shape

Currently boundary calculation for outer object can only fit with rectangular shape of room. Supporting other shapes will increase generality and fluency of rearrangement.

References

[1] John McCormac, Ankur Handa, Andrew Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. pages 4628–4635, 2017.

[2] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, 2019.

[3] Mohammad Keshavarzi, Aakash Parikh, Xiyu Zhai, Melody Mao, Luisa Caldas, and Allen Yang. Scenegen: Generative contextual scene augmentation using scene graph priors. *arXiv preprint arXiv:2009.12395*, 2020.

[4] Mohammad Keshavarzi and Mohammad Rahmani-Asl. Genfloor: Interactive generative space layout system via encoded tree graphs. *arXiv preprint arXiv:2102.10320*, 2021.

[5] Manu.Vision. What i've learned after 100 days of 3d scans with the iphone 12 pro lidar, Jun 2021.

[6] Scott Stein. Lidar is one of the iphone and ipad's coolest tricks., Aug 2021.