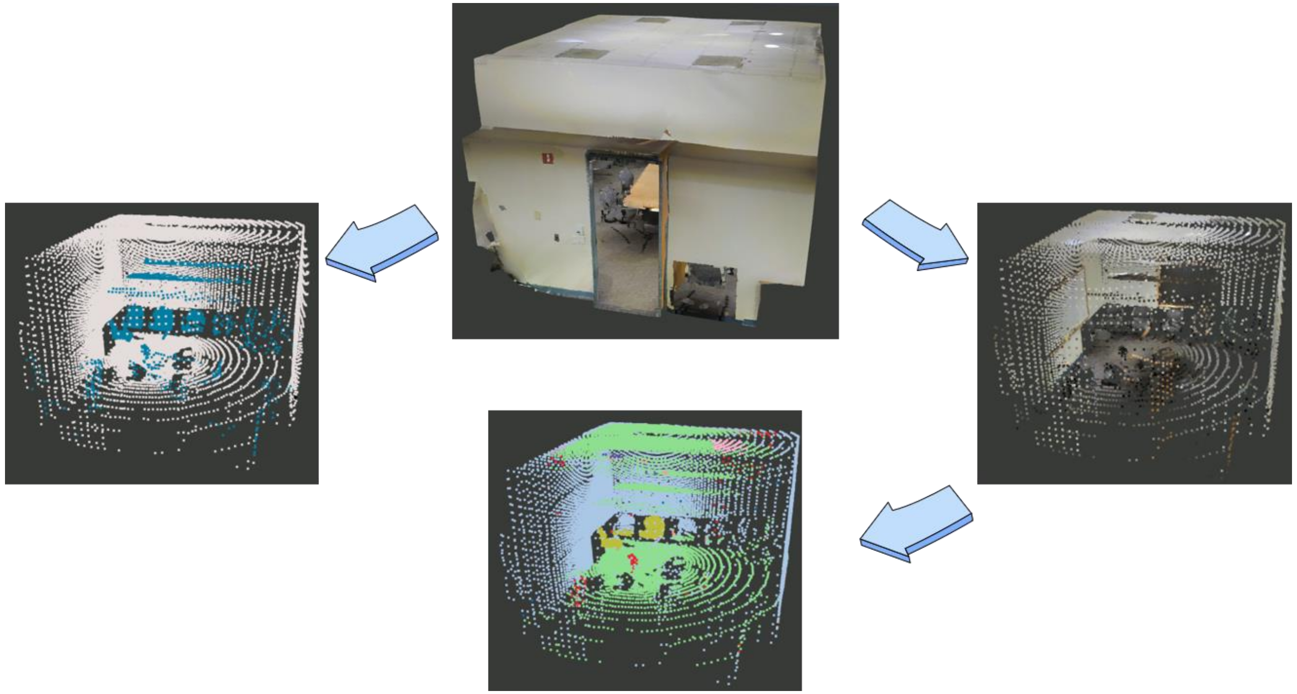
University of Zurich[UZH]

Analyzing the Impact of Occlusion on the Quality of Semantic Segmentation Methods for Point Cloud Data

Bachelor Thesis Final Presentation



Yufeng Xiao

19-763-663

Supervisor: Lizeth Joseline Fuentes Perez

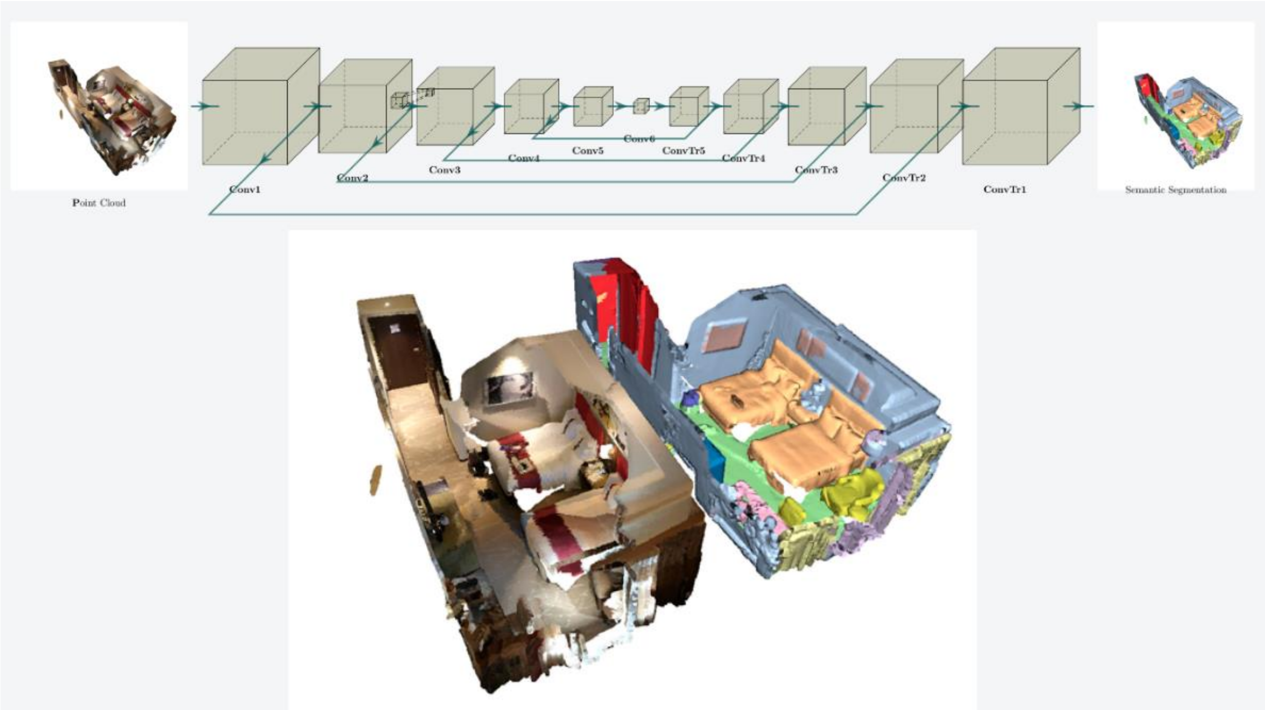
Table of Contents

Project Description - Background

- ***Occlusion*** in point cloud of indoor scene.
  - Missing points in objects or structures.
  - Generated in the process of scanning of the real scene.
  - Cannot be seen by the scanner (viewpoint) because they are obstructed by other objects or structures.

Project Description - Background

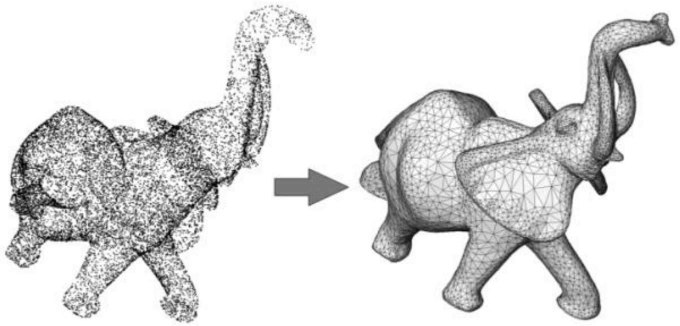
- Minkowski Engine
  - Deep Learning.
  - An auto-differentiation library for ***sparse tensors***.
  - ***Semantic segmentation***.



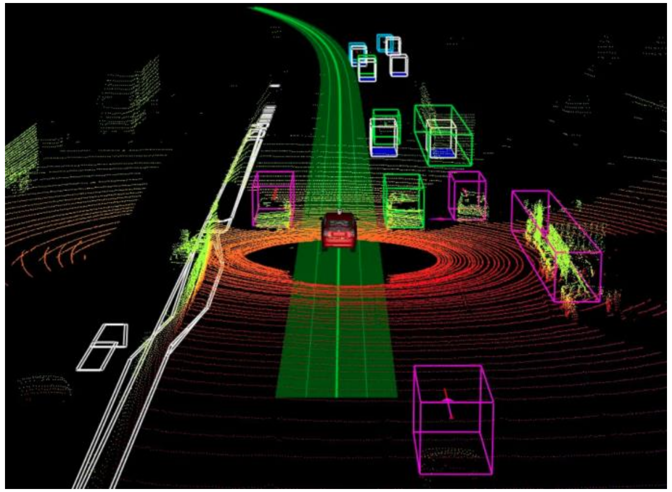
Project Description - Motivation

- Occlusion in point cloud affect different kinds of tasks.
  - Surface reconstruction
  - Object detection
  - 3D model tracking

- **Semantic segmentation** affected by occlusion.
  - It might cause worse **classification**.



Surface reconstruction



Object detection & 3D model tracking

Project Description – Related work

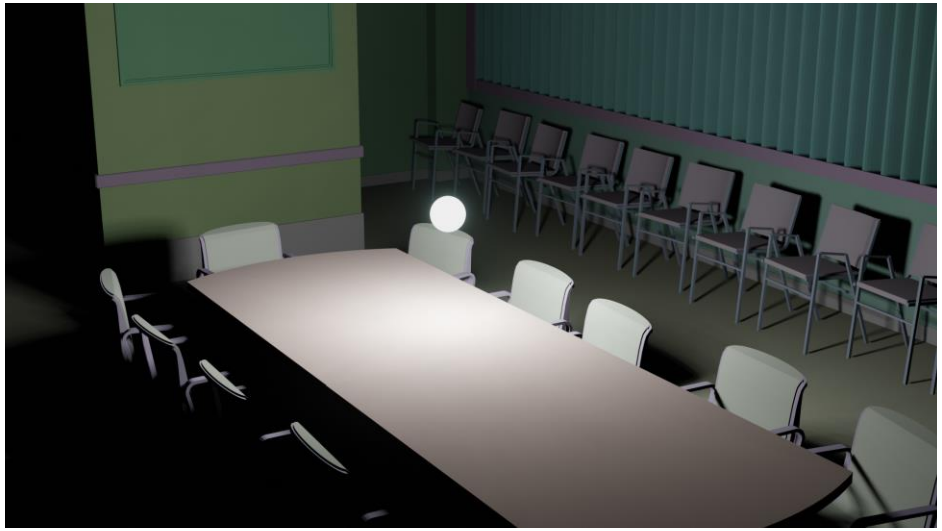
- There is no explicit work on evaluating the ***impact of occlusion*** on performance of semantic segmentation.

- **Occlusion Guided Scene Flow Estimation on 3D Point Clouds**
  - Presented *OGSF-Net*, a novel architecture designed to address the challenges of occlusions in 3D scene flow estimation.
  - Applied ***F1 score*** to evaluate result.

- **OcCo: Unsupervised Point Cloud Pre-training via Occlusion Completion**
  - Proposed *Occlusion Completion,* an unsupervised pre-training method.
  - Use different viewpoints to ***generate occlusion*** in point cloud.

Problem Statement

- In this work we want to find correlation between ***occlusion level*** of ***indoor point cloud (room)*** and ***performance of semantic segmentation***.
  - The core part is to propose a metric to evaluate ***occlusion level*** of the indoor scene and validate its reliability.
- ***Occlusion level*** of ***mesh***.
  - Mesh is the best format to represent a ***real*** scene.
  - ***Occluded area*** represents its occlusion. Propose the metric ***occluded area ratio***.
  - With ***more*** viewpoints there should be ***less*** occluded area.



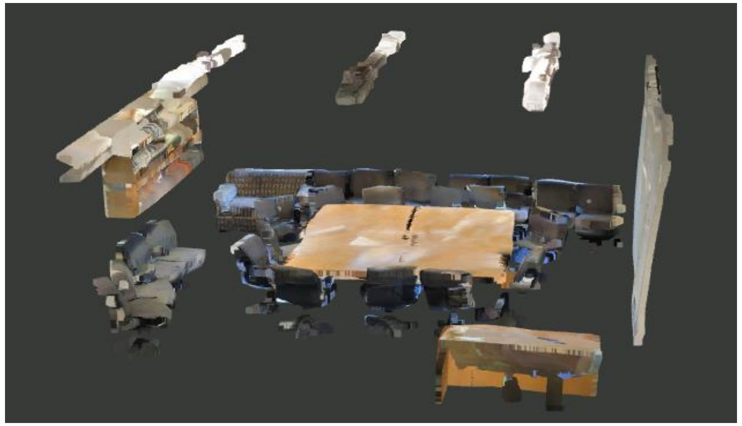
Mesh with 1 viewpoint

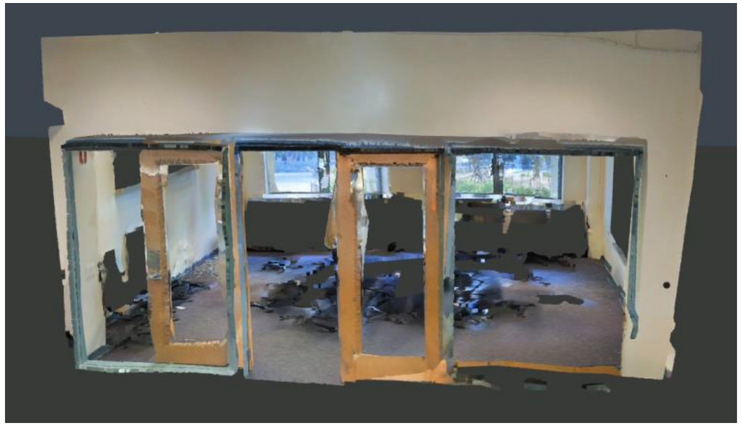
Mesh with 2 viewpoints

Problem Statement

- ***Occlusion level*** of point cloud.
  - There is ***no area*** in a point cloud, we can't compute occluded area.
  - Our point cloud data describe ***closed indoor scene***. Occlusion happens in ***interior items*** and also the ***exterior structures***, but it is hard to quantify occlusion in interior items because of variations of structures.
  - Thus we consider ***exterior/outermost structure*** where most occlusion is shown there. We also call it ***boundary***.
  - Boundary are usually points which represent wall, ceiling and floor etc.

Interior



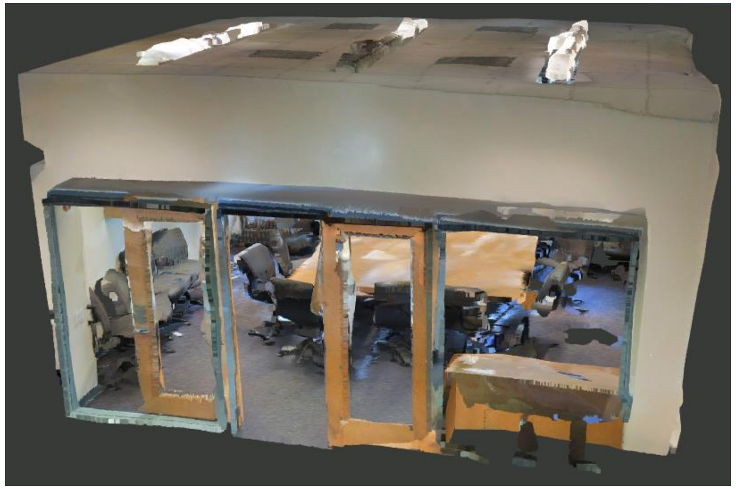


Exterior

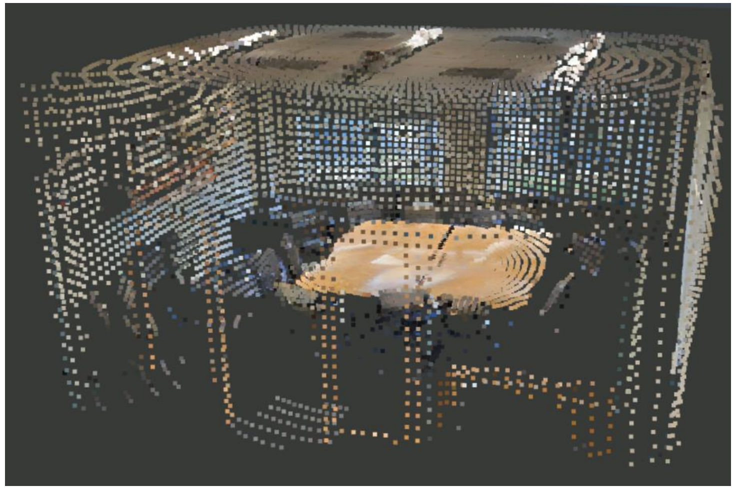
  - Compute ***ray-boundary*** intersection to classify rays, intersect with boundary or not.
  - Propose the metric ***boundary ray ratio*** to estimate occlusion level of point cloud.

Problem Statement

– Compare ***occluded area ratio*** and ***boundary ray ratio*** to prove that the boundary based ratio is ***reliable*** enough to directly estimate occlusion level of point cloud.
  – Since we don't have ***ground truth mesh*** of the indoor scene, we have to estimate mesh from point cloud so that we can compute a comparable ***occluded area ratio***.
  – It is difficult to validate the reliability by comparing these metrics between different scenes since they have ***different structures***.
  – We would do the comparison within the same scene, therefore, it is necessary to generate a set of ***sub-sampled*** point clouds with different level of occlusion.



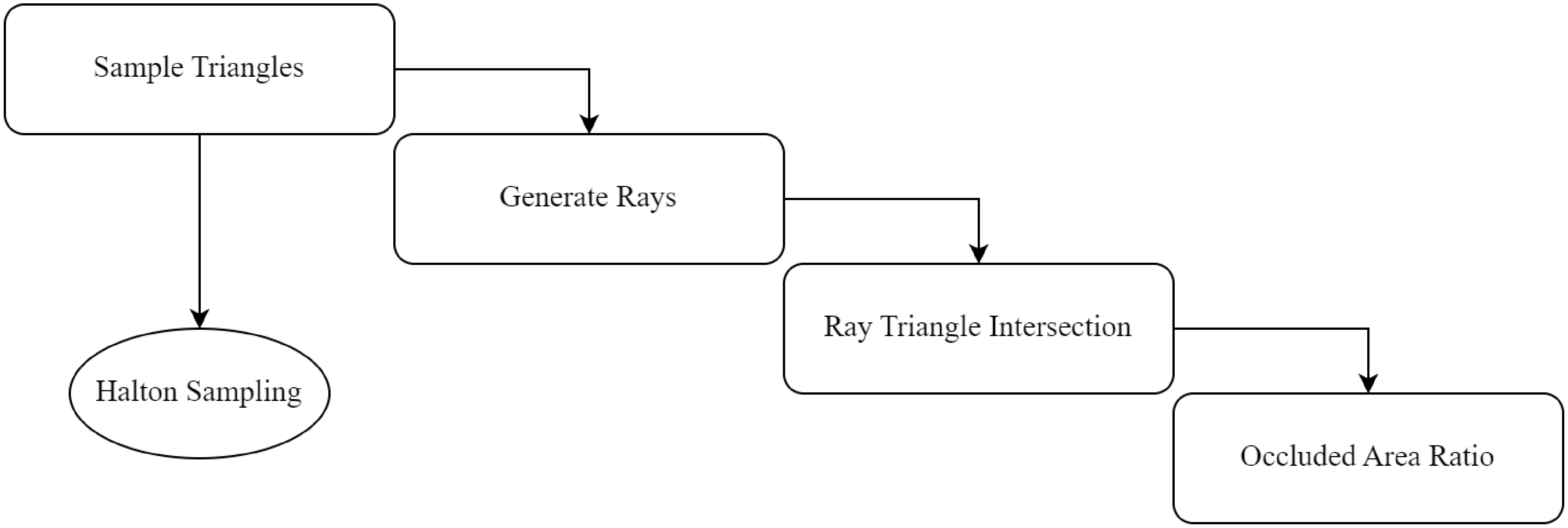
Original



Sub-sampled

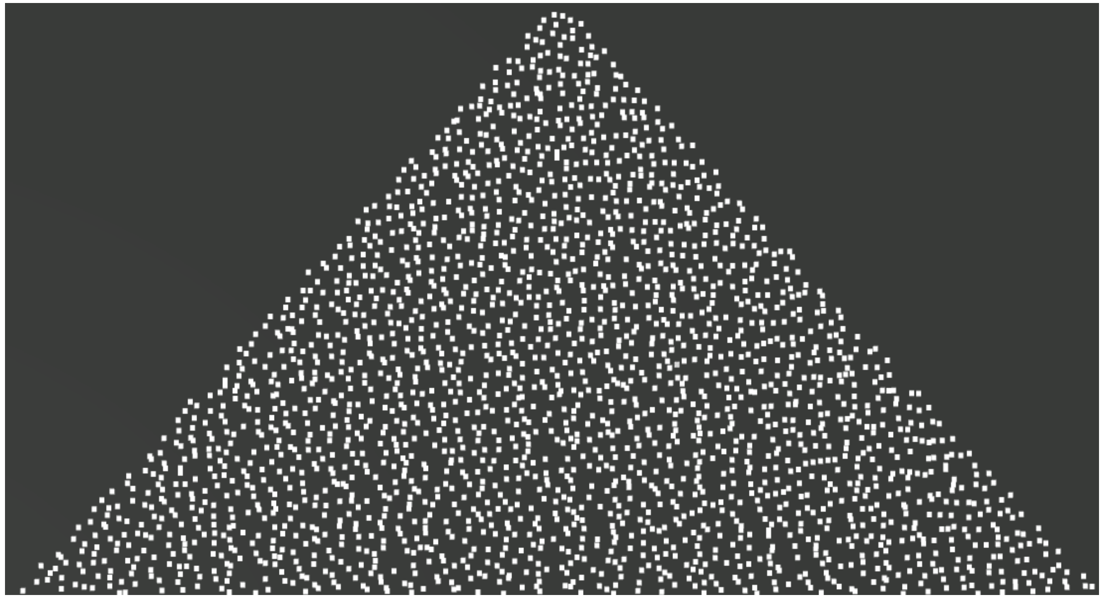
Technical Solution – Occluded Area Ratio

- For each ***triangle*** of the mesh, we compute a weight as occluded ratio.
  - We determine the ratio by calculating how many samplings on the triangle are occluded in terms of certain viewpoint.
  - With the per triangle occluded ratio, we can calculate occluded area of each triangle. Then sum up them to get total occluded area.



Occluded Area Ratio - Sample Triangles

- Halton sampling applied to get a ***uniform distribution*** on the triangle.
  - Compute ***random*** numbers r1 and r2 in the range [0, 1).
  - Compute ***barycentric*** coordinates using r1 and r2.
    - $\alpha = 1 - \sqrt{r1}$, $\beta = \sqrt{r1} * r2$, $\gamma = 1 - \alpha - \beta$.
    - Given vertices of triangle V1, V2 and V3, Sampling $P = \alpha * V1 + \beta * V2 + \gamma * V3$.

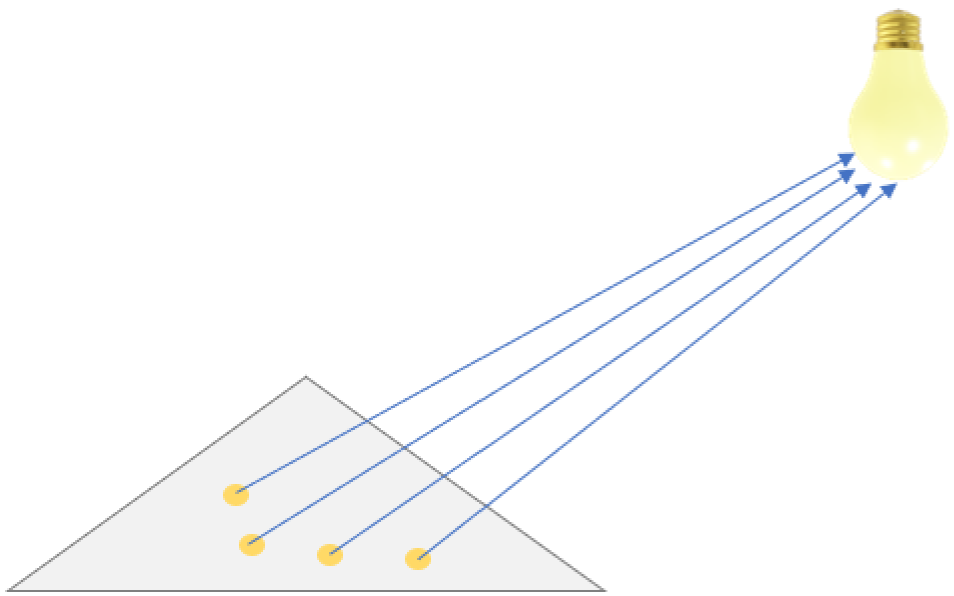


Sampled triangle

Occluded Area Ratio - Generate Rays

- Generate rays from samplings to viewpoint.

- When there are multiple viewpoints, we generate the ***same*** amount of rays from each sampling as the number of viewpoints.



Occluded Area Ratio - Ray Triangle Intersection

- ***Moeller–Trumbore*** ray-triangle intersection algorithm applied here.
  - Check if the ray is ***parallel*** to the triangle.
    - Compute ***normal*** of triangle, check by using dot product with ***direction*** vector.
  - Check if the ***ray-plane intersection*** lies outside of the triangle.
    - Each point on the plane can be represented as:
      - P = (1- u - v) * V1 + u * V2 + v * V3 = V1 + u * (V2 – V1) + v * (V3 – V1), where V2 – V1 and V3 – V1 are edge vectors E1 and E2 respectively.
      - u and v should within the range [0, 1] and (u + v) <= 1.
    - O - V1 = -t * D + u * E1 + v * E2, apply ***Cramer's rule*** to solve this equation to get value of t, u and v.

Occluded Area Ratio – Ratio Computation

- A sampling is visible if it can be seen by ***at least*** one viewpoint, otherwise, it's occluded.
  - Check the ***visibility*** of each sampling by comparing distance d1 from ***sampling to viewpoint*** and distance d2 from ***sampling to first hit intersection***.
  - If d1 < d2, sampling is visible to one of the viewpoints.

- For each triangle we compute occluded ratio based on the amount of occluded samplings and the total number of samplings.

- Compute occluded area for each triangle, then sum up to get total occluded area.
  - ***Occluded area ratio*** = ***total occluded area*** / ***total area.***

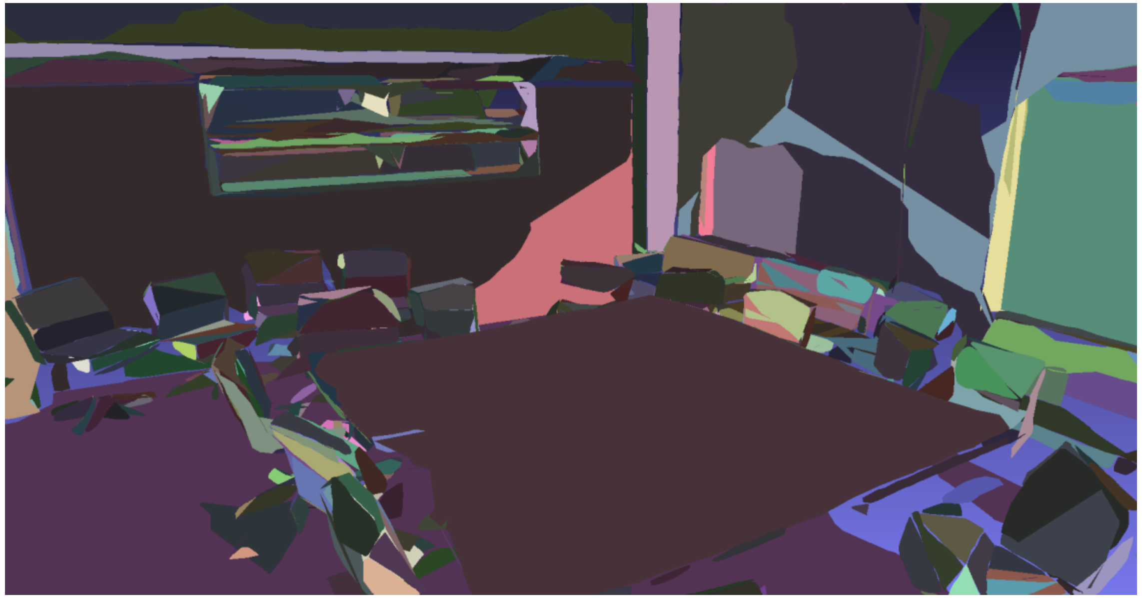
- We use this ratio to represent the occlusion level of a mesh.

Technical Solution – Estimate Mesh from Point Cloud

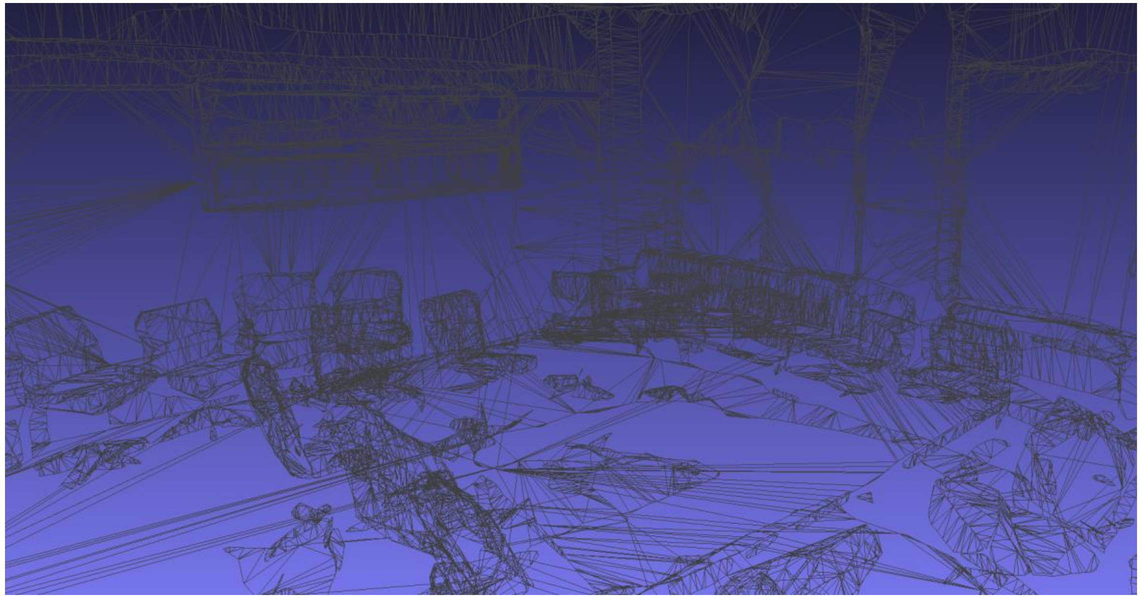
- Based on the paper " ***Finding good configurations of planar primitives in unorganized point cloud*** ".

- We can directly acquire a mesh from point cloud where the mesh has already been triangulated.

- Compute occluded area ratio for this mesh.



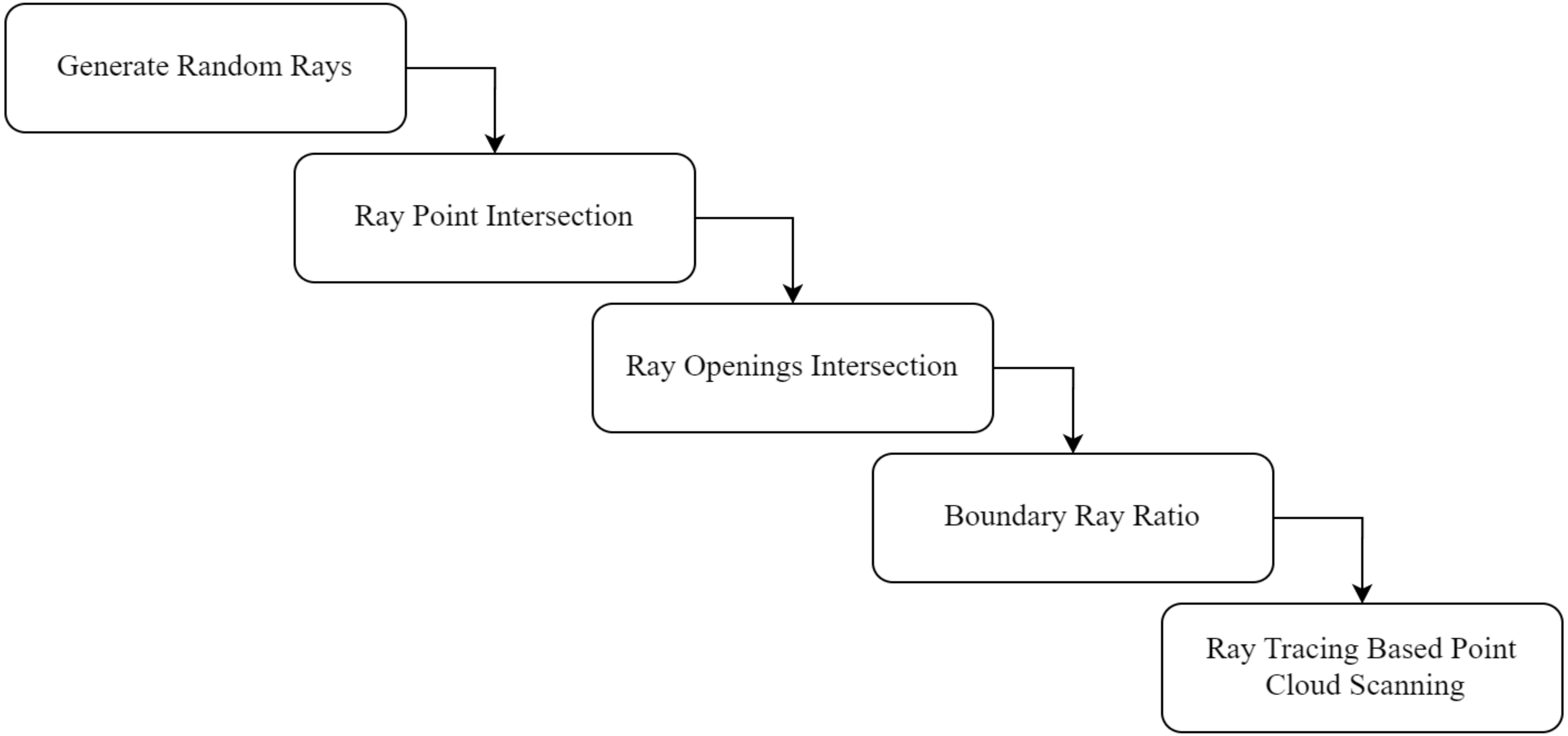
Estimated mesh



Wireframe

Technical Solution – Boundary Ray Ratio

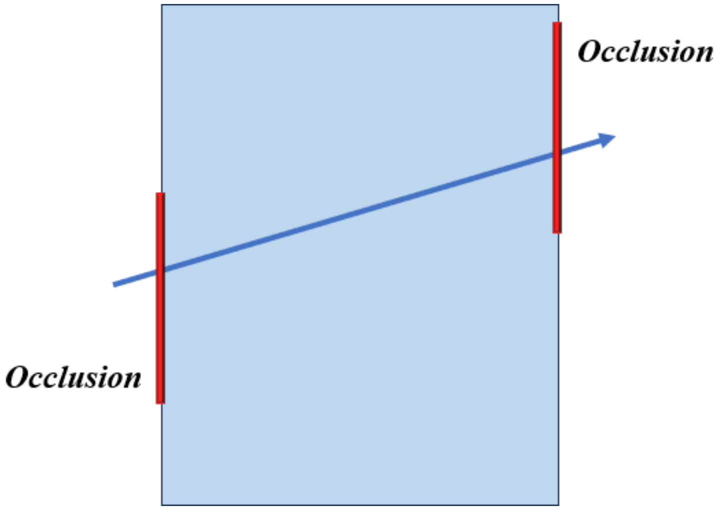
– We focus on rays intersect with boundary point.



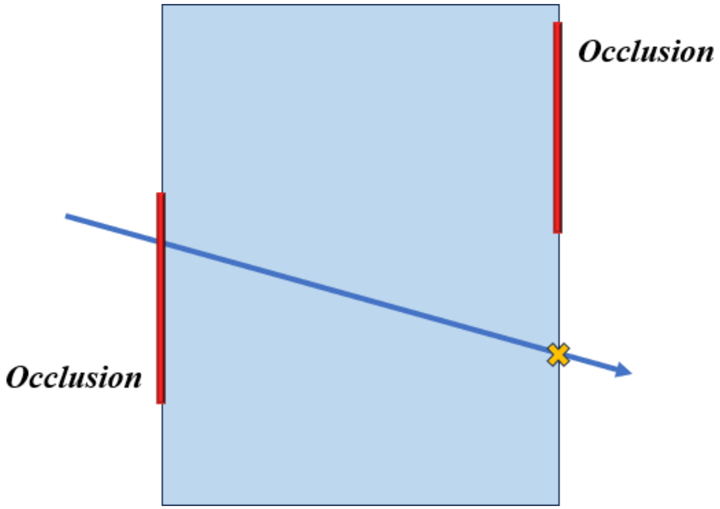
Boundary Ray Ratio – Generate Random Rays

- Cast a ray from outside of the scene(bounding box), in most cases the ray intersects ***2 faces***(except intersect with vertex) of the scene.

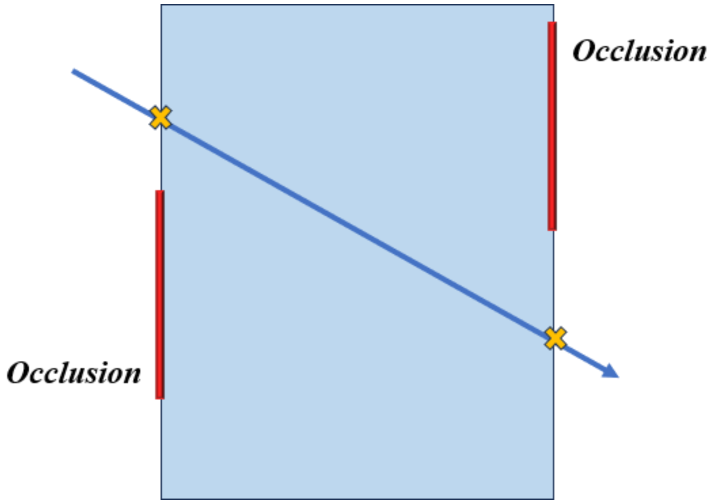
- Ray Classification: Non-boundary ray, 1-boundary ray and 2-boundary ray.
  - If a ray intersects occlusion on 2 faces, it's a non-boundary ray.
  - We are actually intersecting with points, when at least 1 boundary point is hit by ray, it's a boundary ray.



Non-boundary ray



1-boundary ray

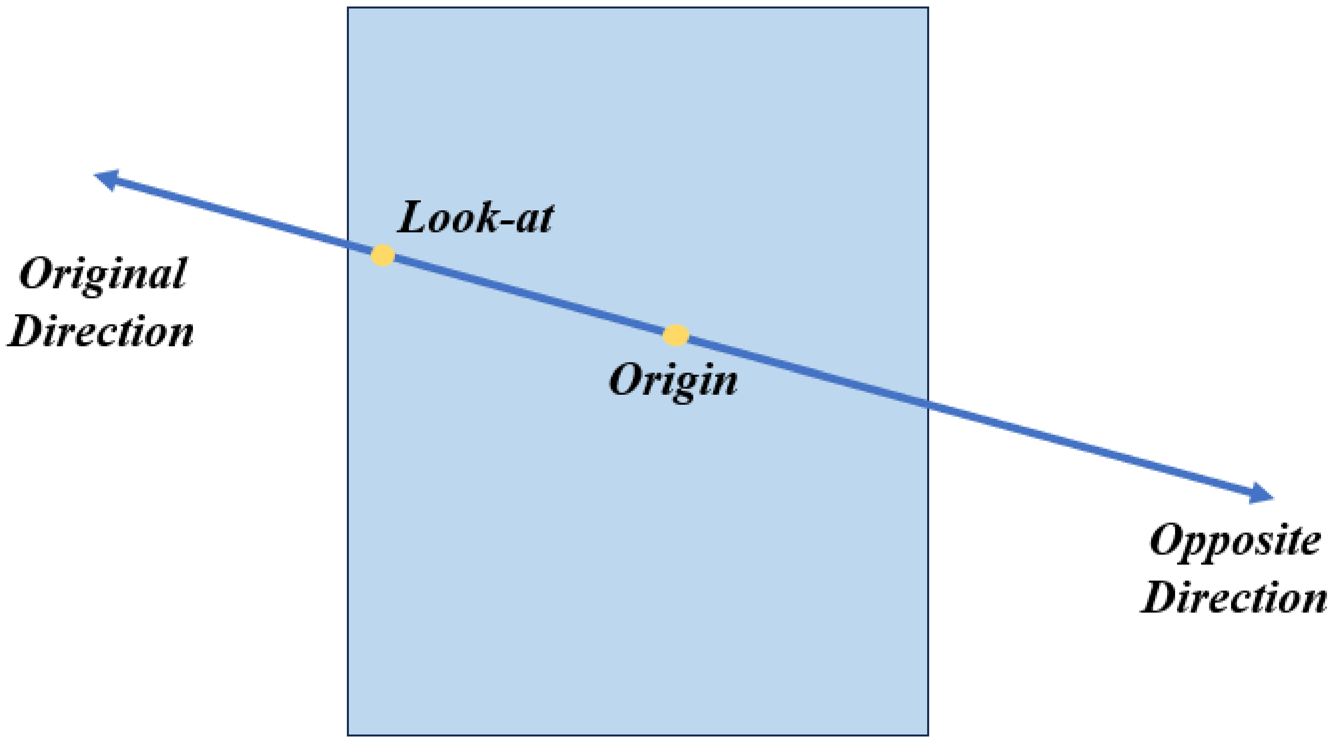


2-boundary ray

Boundary Ray Ratio – Generate Random Rays

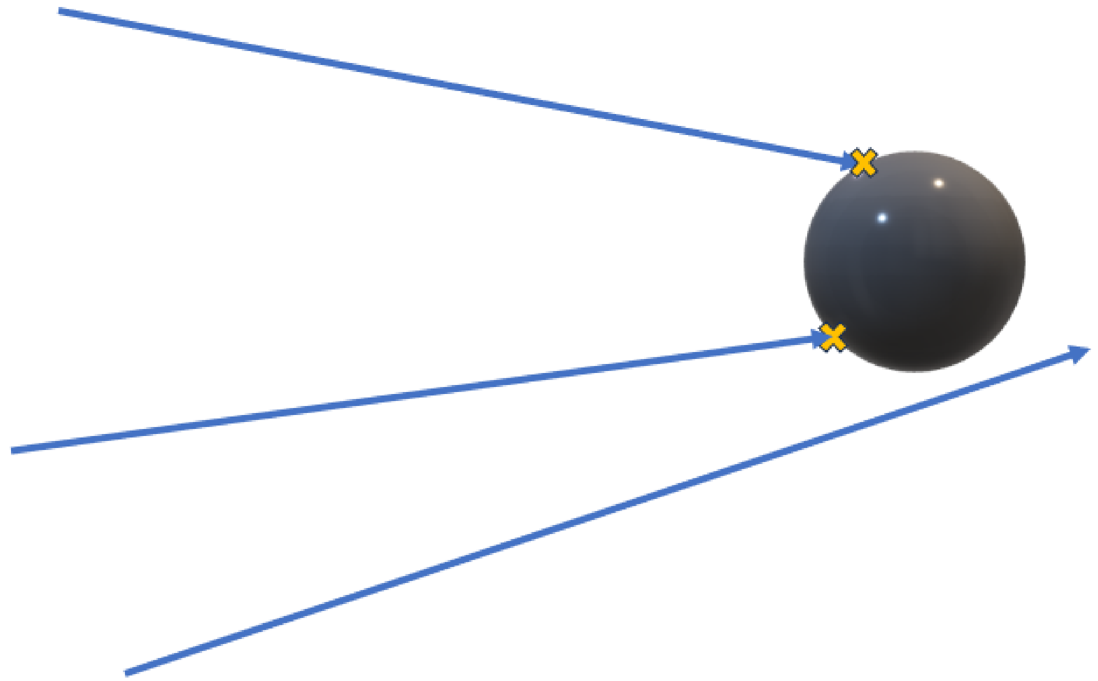
- It is more convenient for us to generate rays from inside of the bounding box.

- We check boundary in 2 directions, the original and opposite.



Boundary Ray Ratio – Ray Point Intersection

- Every point has a radius, thus we are using ray intersecting with sphere.
  - Check if origin if ray is inside the sphere.
  - Compute ***closest distance*** from ray to sphere.
  - Compare distance with radius.



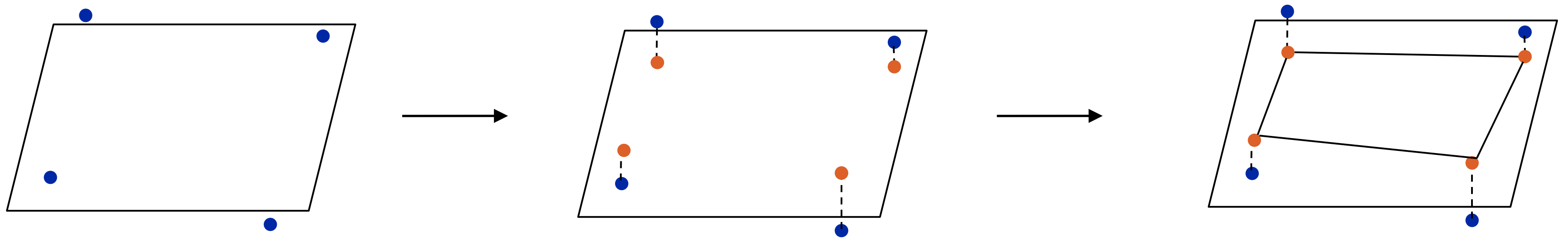
Boundary Ray Ratio – Ray Openings Intersection

– Detect openings by **picking** points in the interactive visualization tool.

Boundary Ray Ratio – Ray Openings Intersection

– Selected points can be estimated as a **polygon**

Estimate plane with RANSAC algorithm → *Project points used in first step onto plane* → *Compute convex hull of projected points to get a polygon*

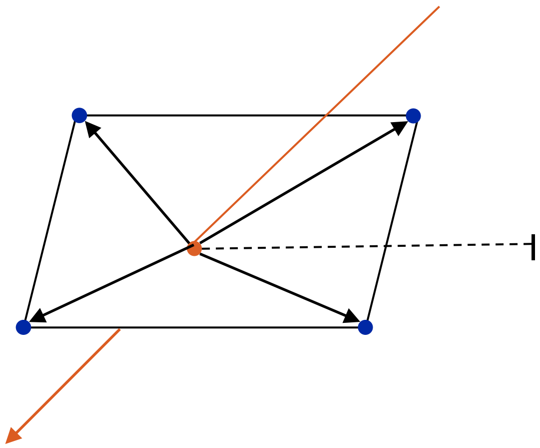


– If ray **intersects** polygon, it’s **not** an occlusion ray

– Compute **intersection** of ray and plane

– Compute **vectors** based on corners and intersection

– Compute **cross product clockwise**, if its sign always the same, intersection is **inside** polygon



Boundary Ray Ratio – Ratio Computation

- ***Non-boundary*** ray and ***1-boundary*** ray have intersection with occlusion, where non-boundary ray has no boundary intersection on both directions.

- Boundary ray ratio(occlusion level) is computed as follows:

$$Occlusion\ Level = \sqrt{\frac{\left(\frac{2}{3}\right) \cdot non\text{-}boundary\ ray\ count + \left(\frac{1}{3}\right) \cdot 1\text{-}boundary\ ray\ count}{total\ rays}}$$

- ***Higher*** weight assigned for non-boundary rays since they contribute more to occlusion level.

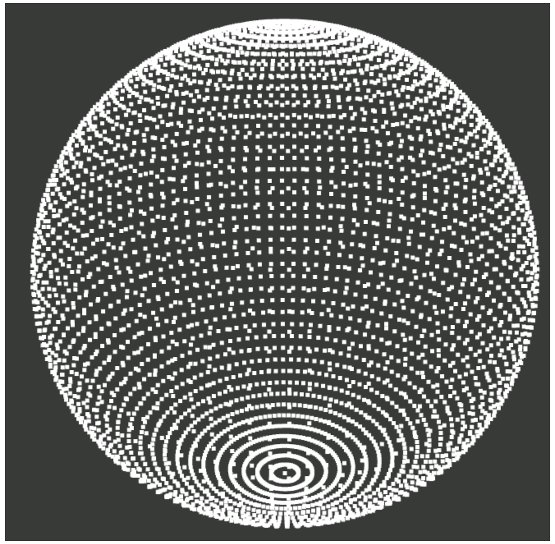
- To make this ratio comparable with ***occluded area ratio***, we apply square root here since the ray fills the ***volume*** of the space and the measurement grows at ***cubic*** rate while the area ratio grows at ***quadratic*** rate.

Boundary Ray Ratio – Ray Tracing Based Point Cloud Scanning

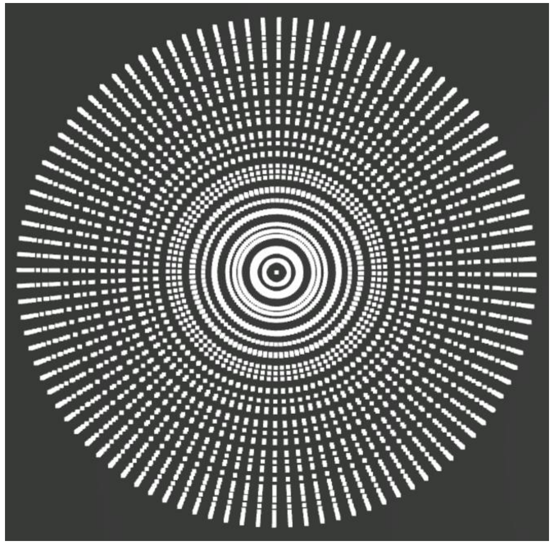
- To validate the ***boundary ray ratio*** is a reliable metric to represent occlusion level we would compare it with ***occluded area ratio***.

- Sufficient data is needed from the same scene to ensure the robustness of comparison.

- Apply spherical light source to scan point cloud to get sub-sampled data.
  - Sample points on surface to specify ray direction.
  - The scanning pattern is inspired by ***longitude*** and ***latitude*** of the earth.



Frontal view



View from above

Technical Solution – Evaluate Performance of Segmentation

- Classify each point in terms of ground truth semantic and predicted semantic into 4 categories:
  - True Positives(TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) .
  - This step will be conducted for all semantic classes.

- Compute precision and recall based on TP, FP, TN and FN.

- Compute ***F1 score*** from precision and recall.

Implementation - Software

- ***PCL*** and ***Eigen*** based C++ backend serves for **computation.**
- ***Three.js*** based frontend serves as user interface.
- Communication via ***WebSocket***.

User interface

Implementation – Octree

- Computation of ray-triangle, ray-point intersection is fast, but when there are 20k rays and 20k triangles/points in the scene, tedious ***iterations*** are time-consuming.

- ***Octree*** is applied to accelerate computation.
  - Ray-triangle intersection in computation of ***occluded area ratio***.
  - Ray-point intersection in computation of ***boundary ray ratio*** and ***point cloud scanning***.
  - Build based on PCL's octree class.

- Time complexity reduced from ***O(m*k)*** to ***O(m*logN)*** where ***m*** is the number of ***rays***, ***k*** is the amount of ***triangles/points*** and ***N*** represents total number of ***nodes*** in octree.
  - For a scene with 20k points there could be only 500 nodes in its corresponding octree.

Experimental Result

- Input data are mesh and point cloud with ground truth label.

- Validation
  - Validate that ***occluded area ratio*** of ***mesh*** will decrease if number of viewpoints increase.
  - Compare ***boundary ray ratio*** with ***occluded area ratio*** to validate that ***boundary ray ratio*** is a reliable metric to reflect the occlusion level of point cloud.

- Correlation
  - Compute ***F1 score*** for point cloud.
  - Compare scores of point clouds with different occlusion level to find correlation.

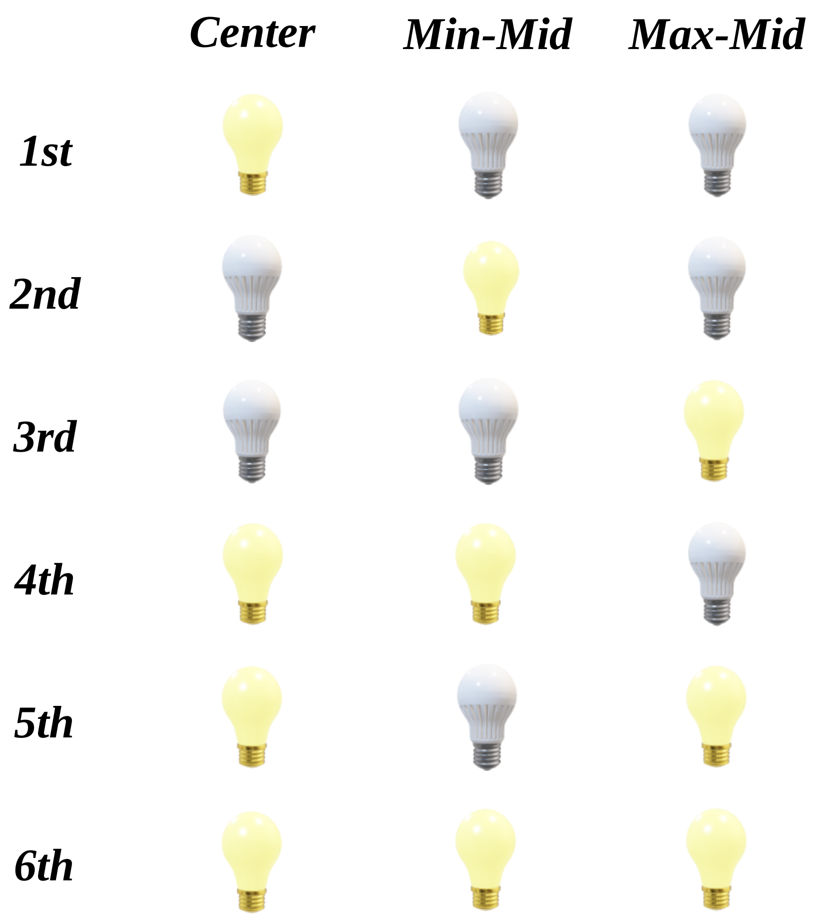
Validation – Viewpoint Pattern

- Three viewpoints used here:
  - Center of the scene's bounding box.
  - Midpoint of center and minimal point.
  - Midpoint of center and maximal point.

- ***6 patterns*** applied to place viewpoints(light sources) in the scene for all experiments.

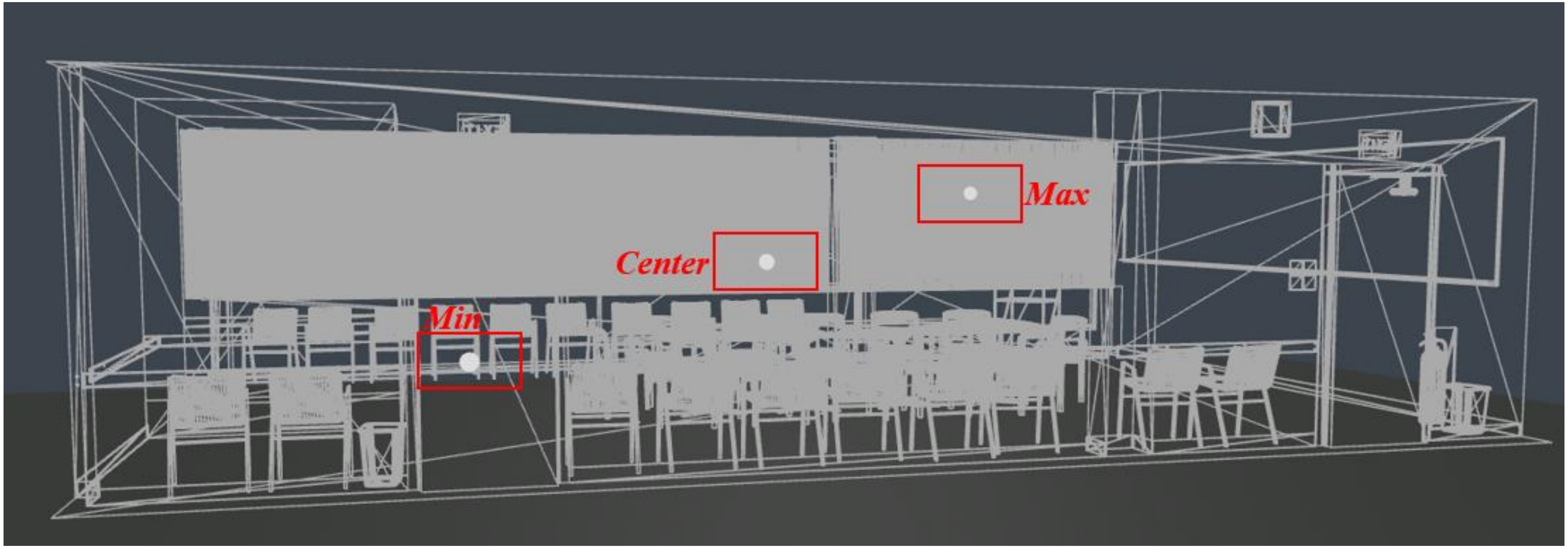
- In 1st pattern there is only one viewpoint in the center.

- In 6th pattern all viewpoints are considered.



Validation – Viewpoint Pattern

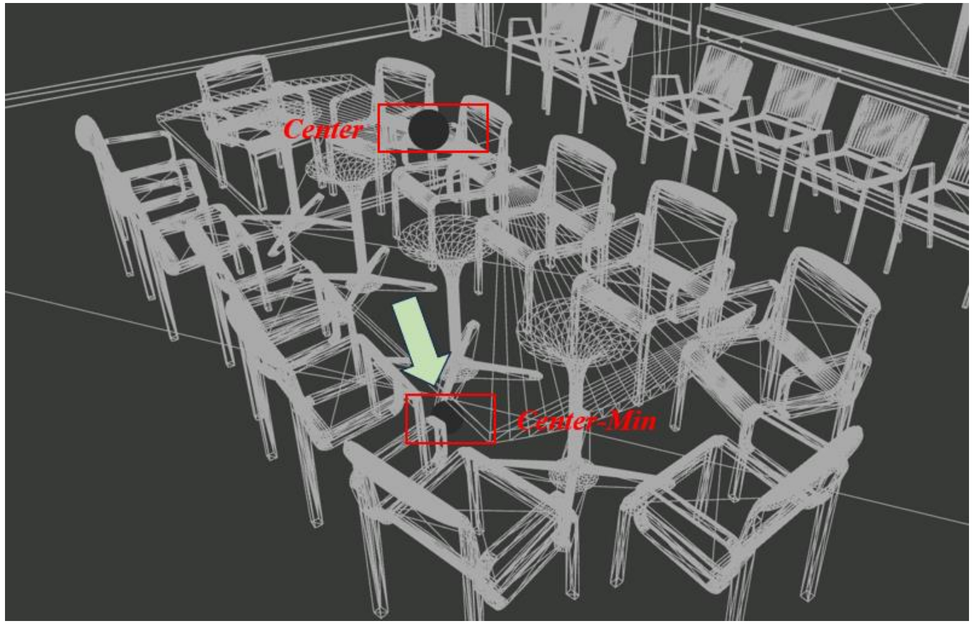
– Input mesh is a conference room.



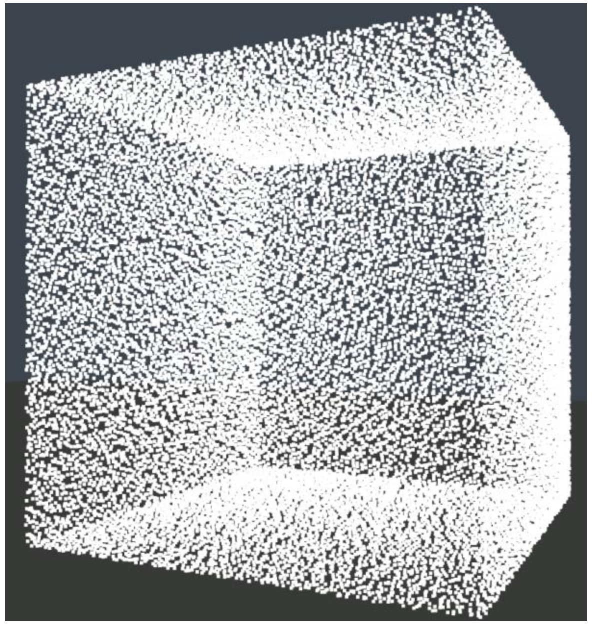
Validation – Occluded Area Ratio of Ground Truth Mesh

- Apply viewpoint patterns to compute ***occluded area ratio***.

- 2 corner cases are also considered:
  - Viewpoint ***under the table*** of the room where most samplings should ***not*** be visible.
  - Viewpoint in a mesh which represents a ***cube*** where all samplings should be visible.



Under Table

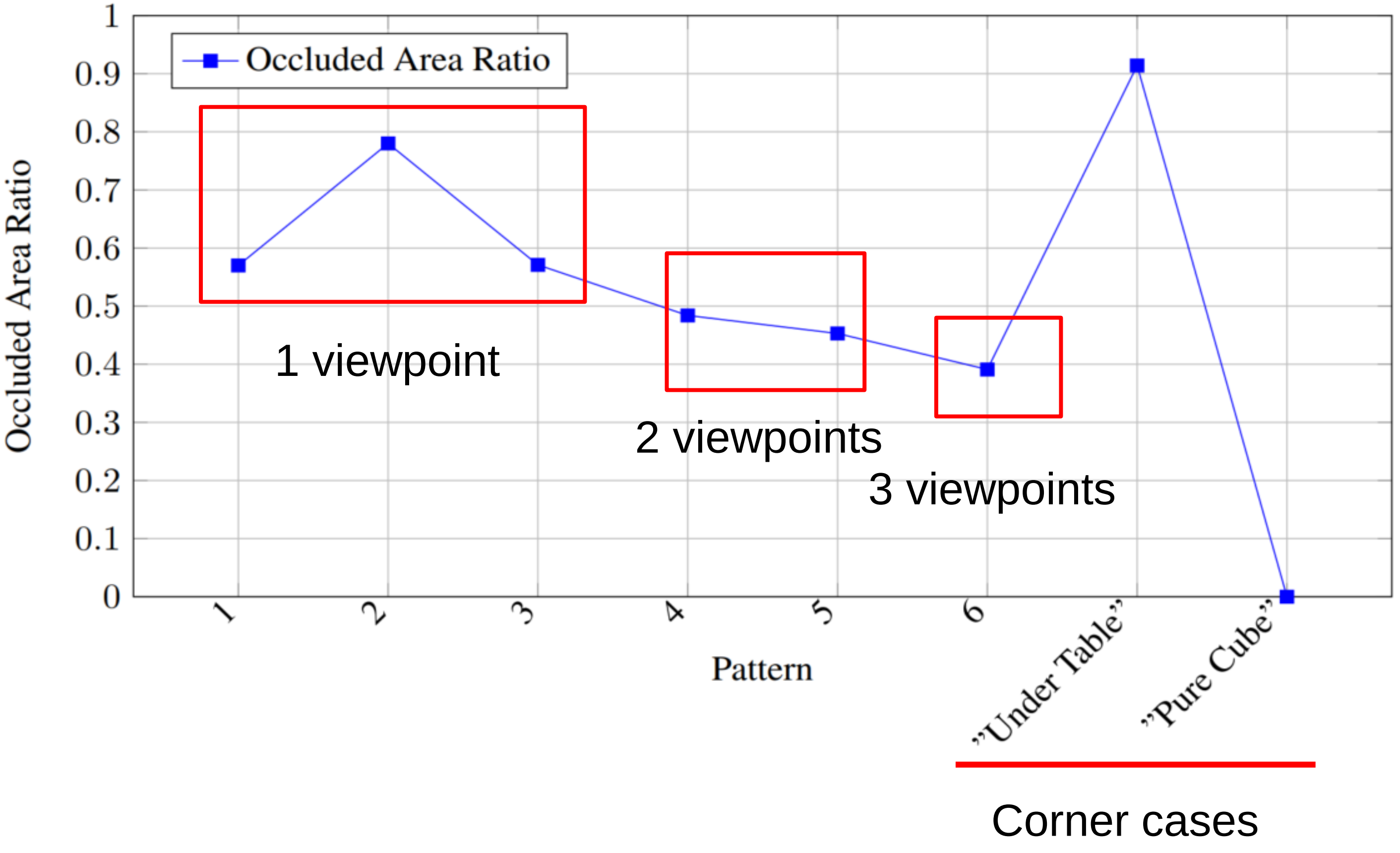


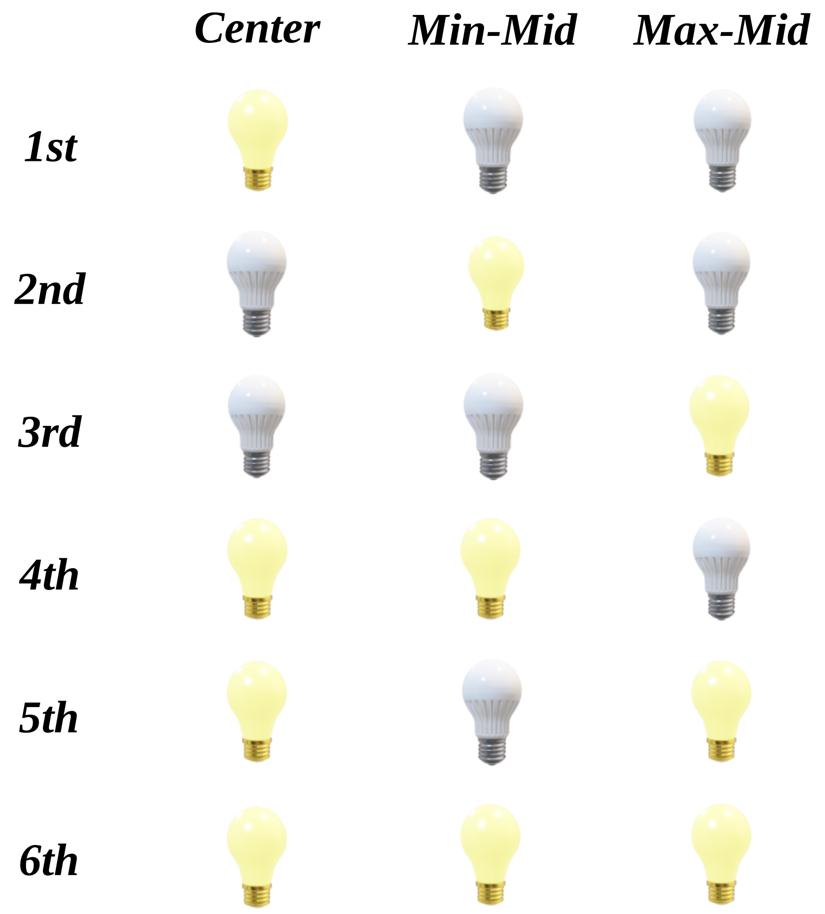
Cube

Validation – Occluded Area Ratio of Ground Truth Mesh

– 6 patterns + 2 corner cases used to compute ***occluded area ratio***.

– ***Lower*** occlusion level when there are ***more*** viewpoints in the scene.

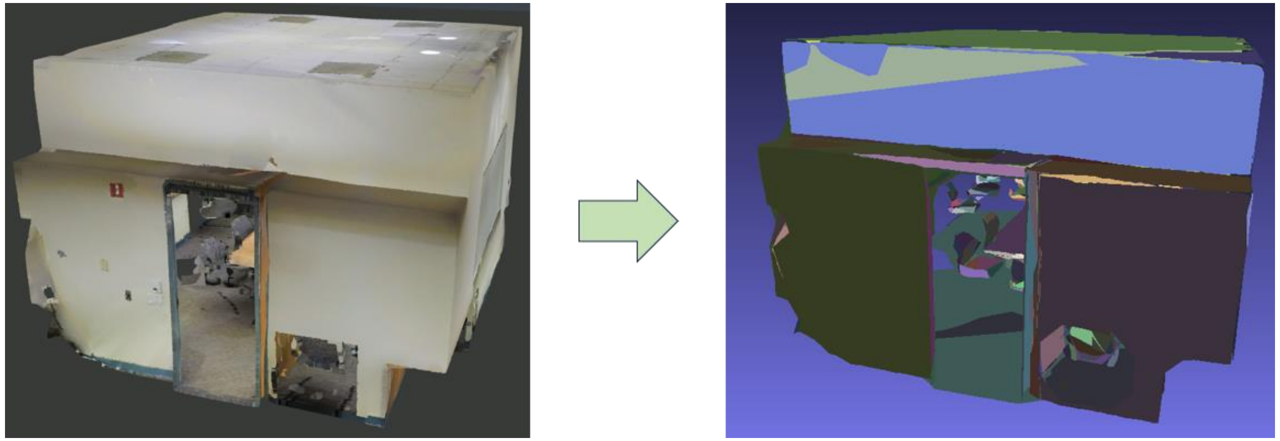




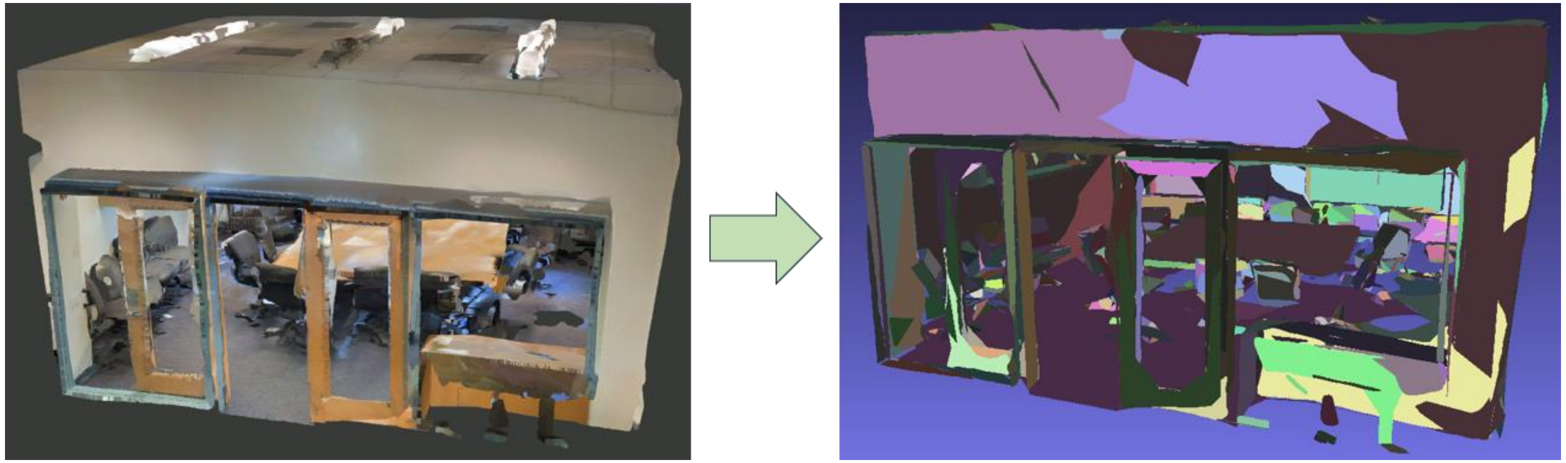
Validation – Occluded Area Ratio of Estimated Mesh

- 6 patterns used to compute ***occluded area ratio*** of ***estimated mesh***.

- Estimate mesh for 2 scenes, room 1 and room 2.



Room 1



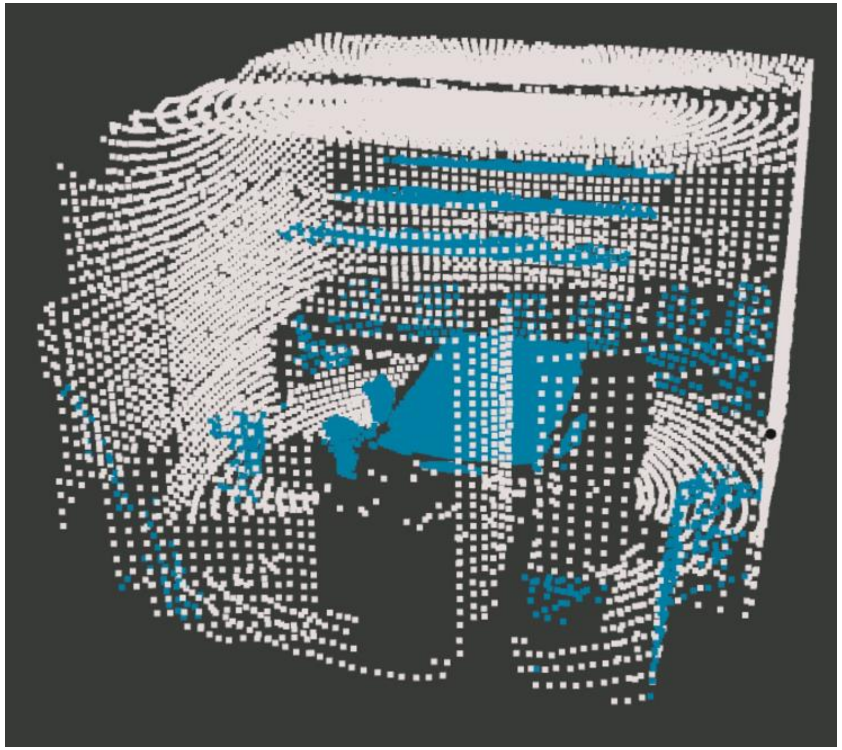
Room 2

Validation – Boundary Ray Ratio

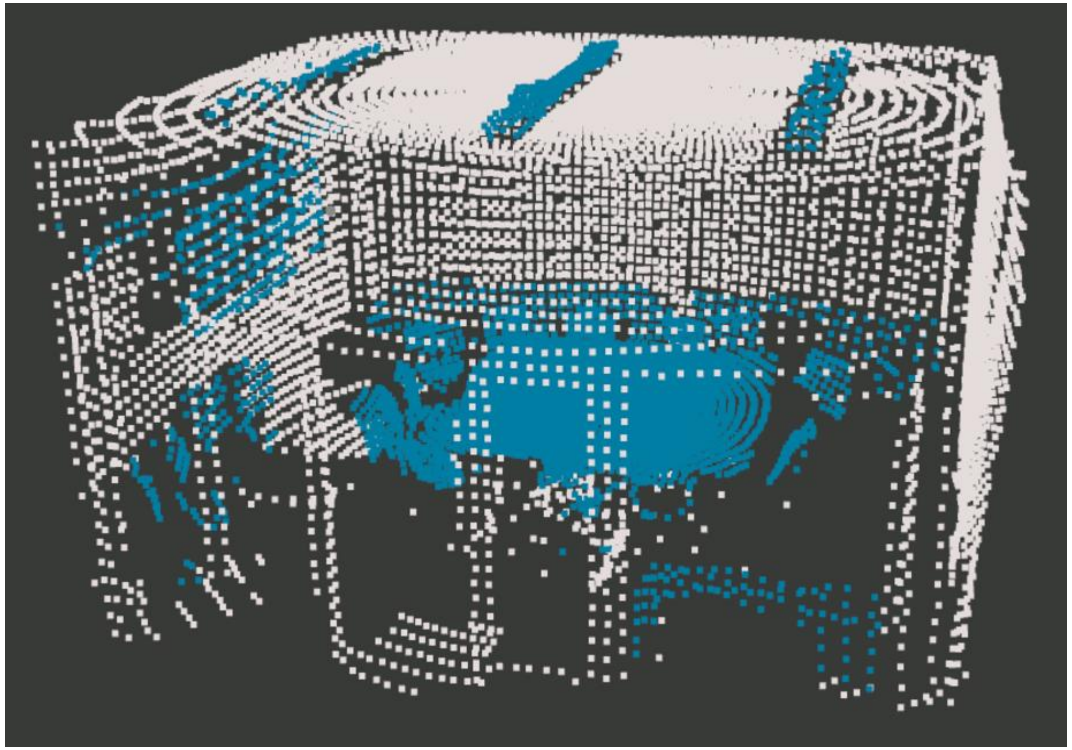
- Scan original point cloud to get sub-sampled data set, ***same patterns*** applied to place light sources.

- Use ***10k*** randomly generated rays to hit the cloud and classify ***boundary rays***.

- In visualization, points in ***white*** are ***boundary*** points, in ***blue*** are ***clutter*** points.



Scanned Room 1 with Light Source in Center

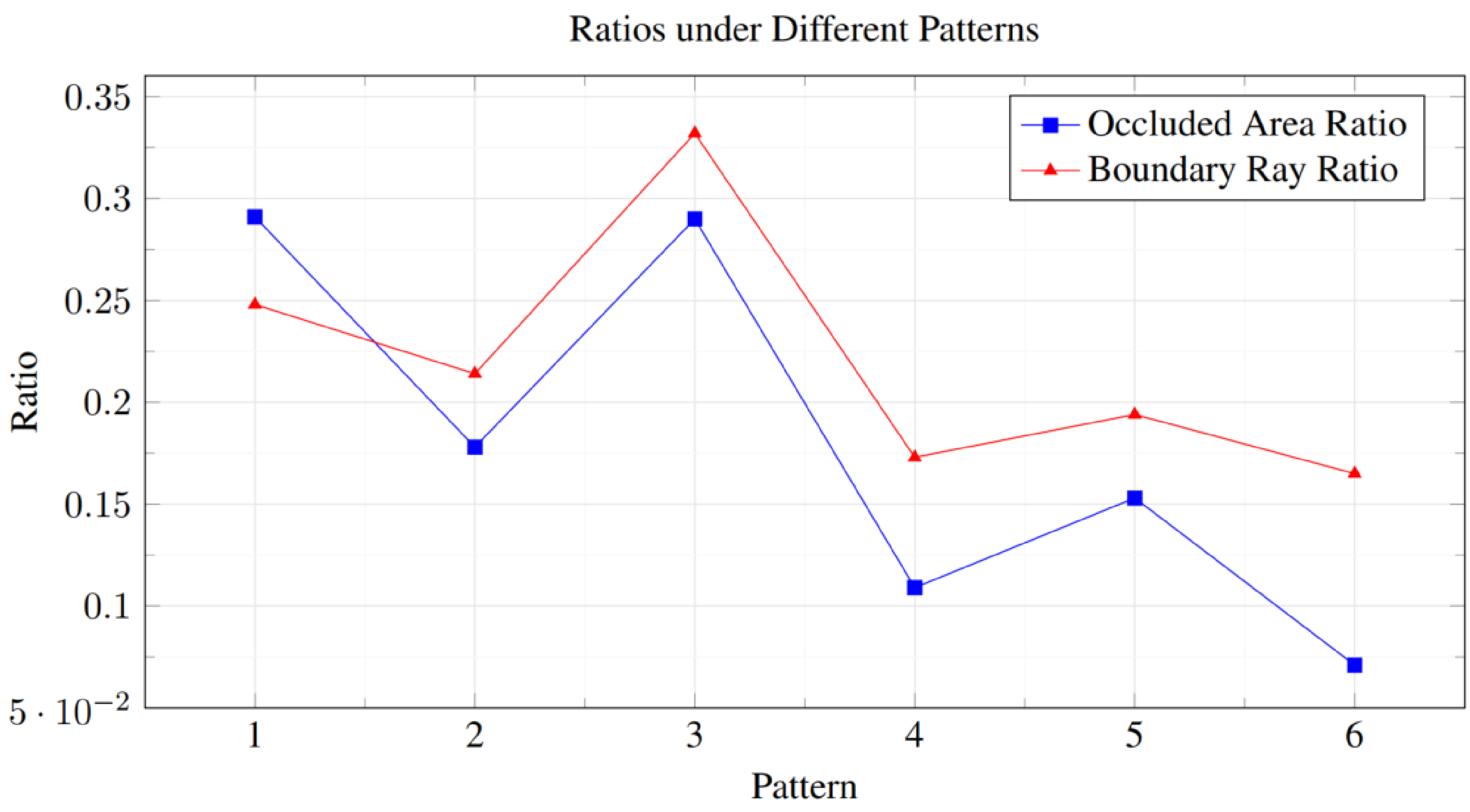


Scanned Room 2 with Light Source in Center

Validation – Result of Room 1

– Difference of value is small.

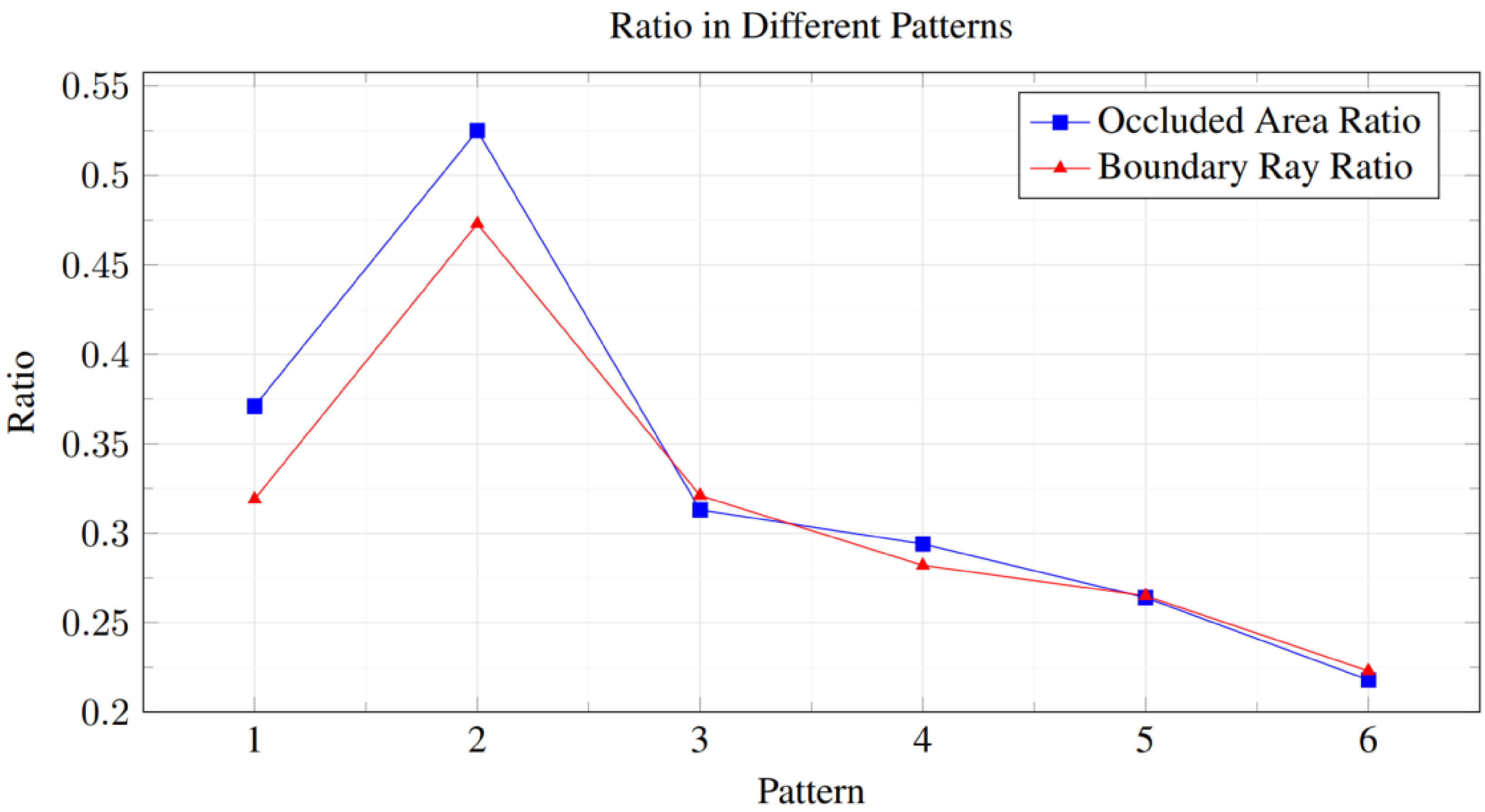
– Same trend when viewpoint pattern changed.



Validation – Result of Room 2

- Same as room 1.

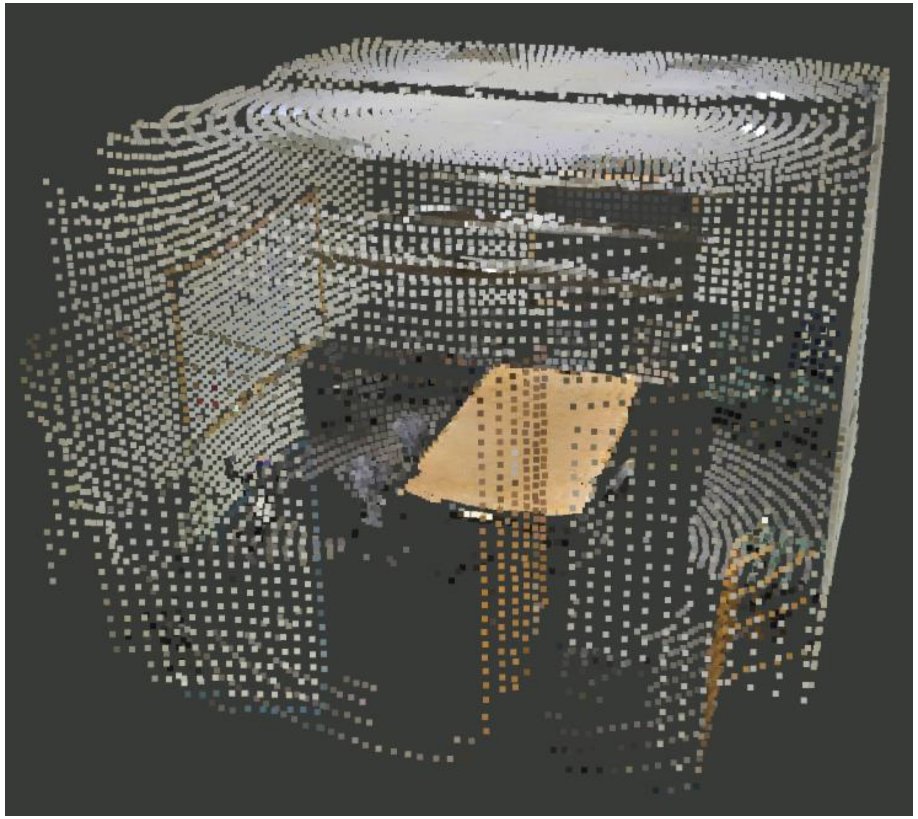
- Through these comparisons, we conclude that boundary ray ratio is a reliable metric to estimate occlusion level of point cloud.



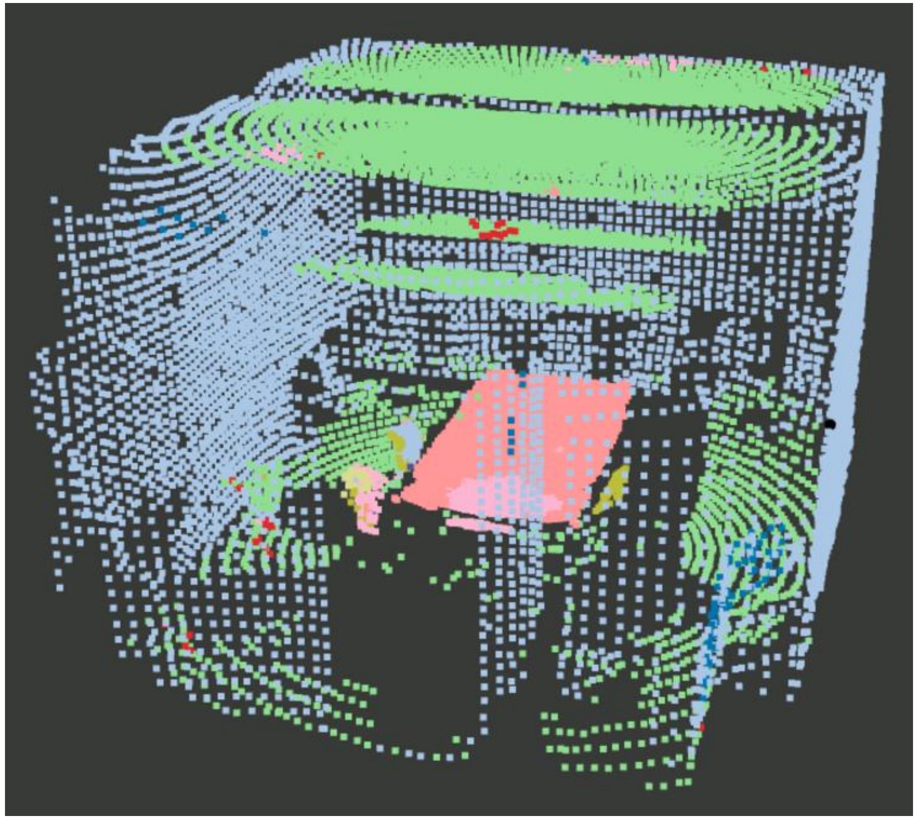
Correlation

– Input ***scanned*** point cloud into ***Minkowski Engine*** to get result of semantic segmentation.

– Compute ***F1 score*** for segmented cloud.



Scanned Cloud with Original Color

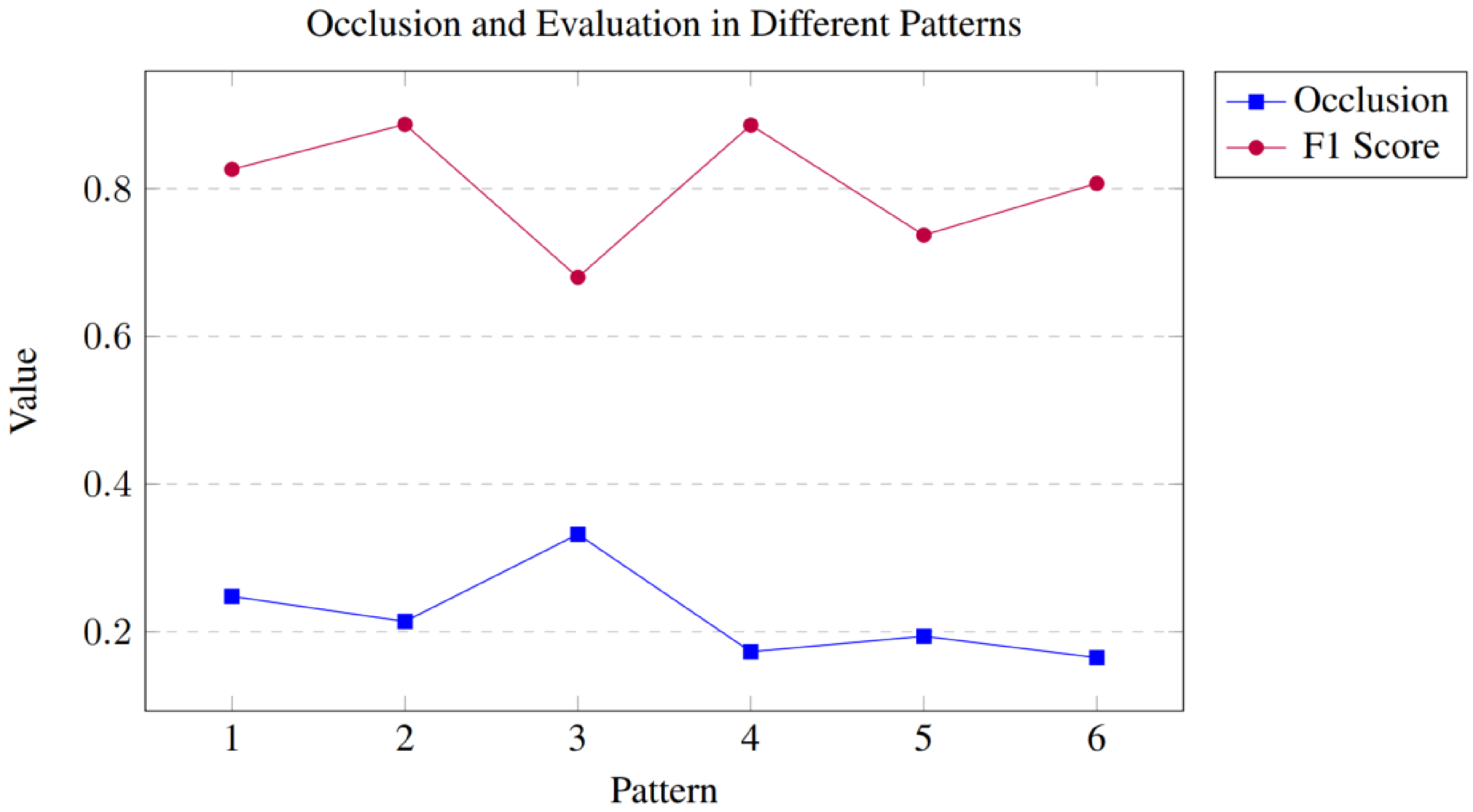


Segmented Cloud

Result of Room 1

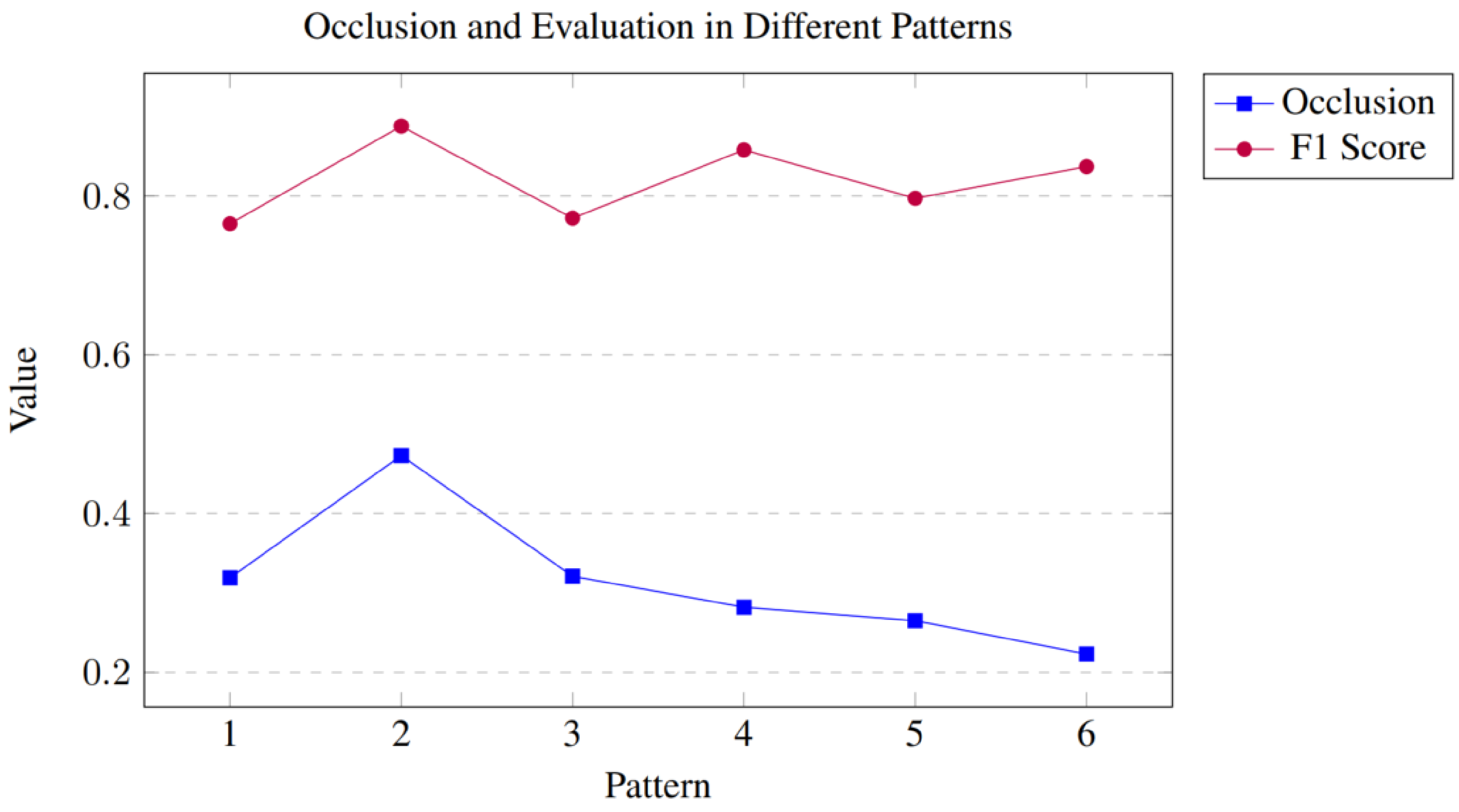
– Shows an ***opposite*** trend between ***occlusion level*** and ***F1 score***.

– In general, the two metrics are inversely proportional.



Result of Room 2

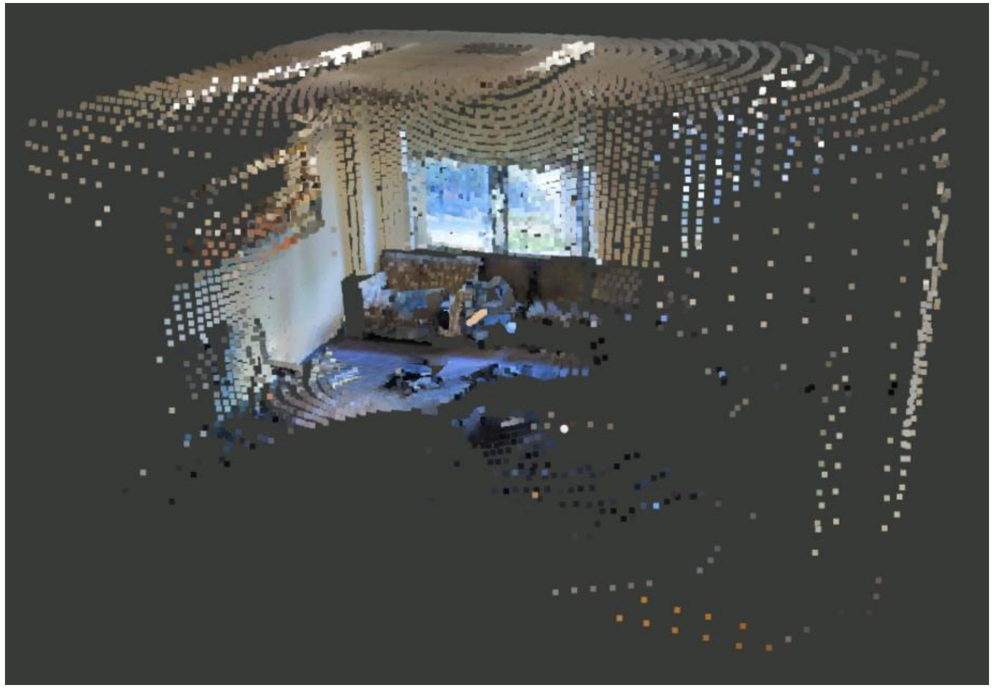
– In this case, ***no correlation*** can be found.

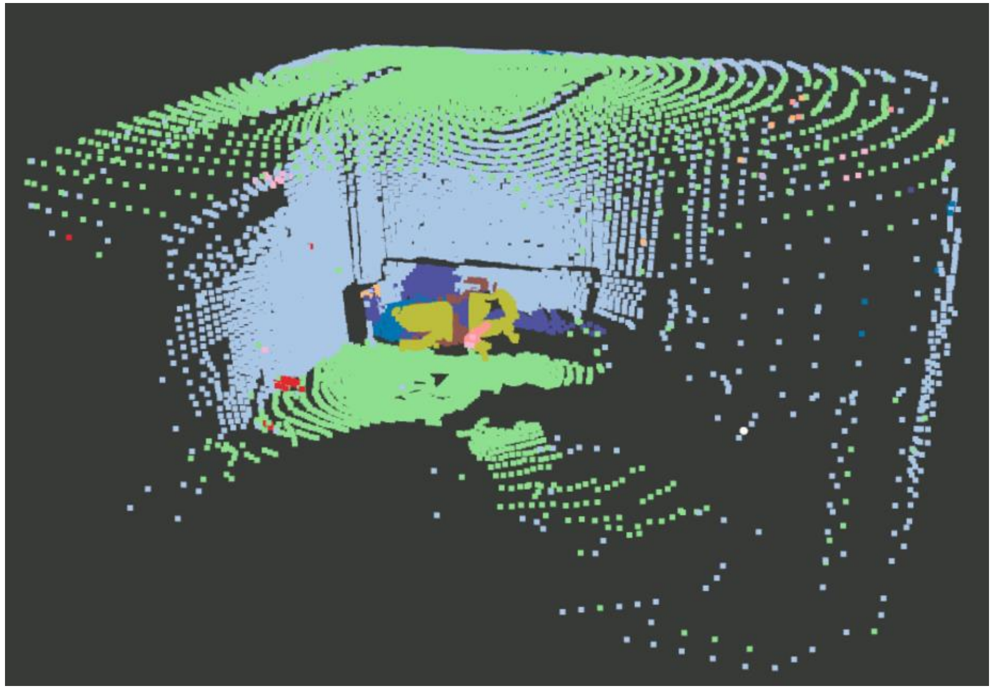


Discussion

- In the result of room 2, scanned point cloud under ***pattern 1*** where the light source placed in the ***minimal*** midpoint, has the ***highest loss*** of information but the ***highest F1 score***.

- Region close to light source is ***dense***, thus the segmentation might have better understanding there due to ***high richness*** of structural information.





Discussion

– Performance is also affected by ***structure*** of point cloud and ***density*** in certain regions. Occlusion alone is not enough to influence the result.

– More influential factors should be taken into consideration:
  – Variations in placement of light sources.
  – Diversity of interior items.
  – Spatial relationship between different objects and structures.

– Occlusion level has ***limited*** impact on performance of semantic segmentation.

Limitation

- We are not computing the ***real*** occlusion of point cloud.
- Our way of estimation is achieved through ray intersecting boundary points, which is highly ***dependent*** on the ***exterior structure*** of the scene.
- Due to ***limited diversity*** of data used in experiments, we cannot guarantee that the metric will always remain effective.

Future Work

- More data with different structures and complexities can be used in experiments.
- Propose more comprehensive metrics that emphasize the ***relationships*** among elements in the scene.
- Alternatively, we can develop an ***evaluation system*** to assess the completeness of point cloud, where influential factors could be assigned with different ***weight*** for computation.

Demo

– Video showcase.

– File explorer window is not displayed.

– https://www.youtube.com/watch?v=p31nPK5FAT0

Thanks for your attention!