Improving 3D Scene Segmentation with Prior 3D Object Knowledge

Laszlo Szilagyi
Stanford University
laszlosz@stanford.edu

Abstract

Accurate 3D scene understanding is essential for robotics and augmented reality (AR), where high-quality instance segmentation and semantic scene graphs enable downstream reasoning and interaction. While recent methods such as ConceptGraphs [4] leverage vision-language models (VLMs) and large language models (LLMs) to segment RGB-D sequences and build open-vocabulary scene graphs, they are limited by incomplete viewpoint coverage, resulting in partial object reconstructions. This paper proposes a complementary approach that integrates prior knowledge in the form of known 3D object models to refine and complete partial reconstructions. The method identifies candidate object segments using semantic similarity from CLIP [9] embeddings and aligns reference objects via robust geometric registration pipelines based on FPFH [10] or PREDATOR [5] features, followed by RANSAC [3] and ICP [13]. Integrated into the ConceptGraphs pipeline, the approach shows improved global and per-object segmentation accuracy on the Replica [11] dataset, particularly for large and partially observed objects. This work demonstrates the effectiveness of incorporating object-level priors for more complete and accurate 3D scene representations, and lays the groundwork for injecting instance-specific semantics and affordances into scene graphs.

1. Introduction

3D semantic understanding is a foundational capability for robotics and augmented reality (AR) applications. Instance segmentation of 3D point clouds and the construction of semantic scene graphs are key components for building rich, structured representations of real-world environments. Recent advances have leveraged RGB-D data in combination with vision-language models (VLMs) and large language models (LLMs) to support both closed- and open-vocabulary semantic understanding. Notable examples include ConceptGraphs [4], Open3DSG [6], and OpenFunGraph [12], which demonstrate the potential of language-integrated perception systems.

Despite their promise, current methods face several limitations. The quality of 3D reconstruction and instance segmentation is often constrained by incomplete viewpoint coverage, leading to partial object reconstructions and geometric artifacts. Additionally, scene graph generation methods typically achieve only 60–80% accuracy in semantic and relational understanding, due in part to challenges in multi-view RGB-D capture and the use of general-purpose models trained on broad internet-scale datasets rather than environment-specific domains.

This project aims to address these limitations by incorporating prior knowledge of the environment into the mapping process. In many real-world robotics and AR scenarios, the types of objects likely to be encountered—such as specific pieces of furniture, appliances, or machinery—are known in advance. The central hypothesis is that integrating reference 3D models and associated semantic labels for known objects can improve the quality of the resulting point cloud and enhance instance-level recognition accuracy. Such object-level specificity also allows the inclusion of additional information—such as affordances and interaction capabilities—which can strengthen the expressiveness of the scene graph (e.g., identifying and annotating how to operate a particular coffee machine).

To explore this hypothesis, this project focuses on the perception layer, proposing a hybrid 3D mapping approach that combines generic RGB-D + VLM-based mapping (as implemented in ConceptGraphs [4]) with a reference-based registration module. This module leverages two point cloud registration pipelines—one based on FPFH [10] with RANSAC [3] and ICP [13], and another using the learning-based PREDATOR [5] model followed by RANSAC and ICP—to align known object models within the scene. The method is integrated into the ConceptGraphs stack and forms the foundation for follow-up work targeting downstream reasoning and planning.

2. Related Work

The main inspiration of this project, ConceptGraphs [4] builds an open-vocabulary 3D scene graph from a sequence of posed RGB-D images. The method uses generic instance segmentation models to segment regions from RGB images, extract semantic feature vectors for each, and project them to a 3D point cloud. These regions are incrementally associated and fused from multiple views, resulting in a set of 3D objects and associated vision (and language) descriptors. Then large vision and language models are used to caption each mapped 3D objects and derive inter-object relations, which generates the edges to connect the set of objects and form a graph. The resulting 3D scene graph provides a structured and comprehensive understanding of the scene and can further be easily translated to a text description, useful for LLM-based task planning.

A related project is OpenFunGraph [12], which builds functional 3D scene graphs. Unlike traditional 3D scene graphs that focus on spatial relationships of objects, functional 3D scene graphs capture objects, interactive elements, and their functional relationships. Similar to ConceptGraphs this method also leverages foundation models, including visual language models (LLAVA [7]) and large language models (ChatGPT [8]), to encode functional knowledge. The authors claim, that the method significantly outperforms adapted baselines, including Open3DSG and ConceptGraph, demonstrating its effectiveness in modeling complex scene functionalities. While both ConceptGraphs and OpenFunGraph provides great scene graph building performance, they are not leveraging prior wknoledge of the environment. Depending on the 3D environment mapping circumstances, large chunks of objects can remain unreconstructed. This is where I want to make improvements.

Scan2CAD [2] and SceneCAD [**?**] are two influential methods that address the problem of aligning 3D models to RGB-D scans of indoor environments. Scan2CAD focuses on matching individual objects in scans to CAD models from ShapeNet by learning geometric correspondences via a 3D convolutional neural network, followed by optimization to estimate 9DoF alignments. In contrast, SceneCAD goes beyond per-object alignment and jointly optimizes both object placements and the surrounding room layout using a graph-based reasoning module, producing a globally consistent and lightweight CAD reconstruction of the entire scene. Both methods aim for plausible and semantically consistent reconstructions using category-level CAD models, but do not attempt to identify or align specific instances of known objects. In this project, I pursue a more targeted goal: leveraging prior knowledge of known object instances—represented as labeled 3D point clouds—to improve the quality of the 3D map. By combining semantic filtering using CLIP [9] embeddings with robust geometric registration techniques (FPFH/PREDATOR, RANSAC, and ICP), my method enables instance-level alignment of specific reference objects. This allows not only for more complete and accurate reconstruction of known objects but also lays the groundwork for integrating object-specific semantics and affordances in downstream scene understanding.

For the pointy cloud registration, I have been relying on two main body of work: Fast Point Feature Histograms (FPFH) [10] and the PREDATOR [5] method. FPFH is used to generate local geometric descriptors for sampled keypoints in both the source and target point clouds. These descriptors encode surface normals and spatial relationships with neighboring points, providing a robust feature representation even in noisy environments. Once computed, FPFH descriptors are matched between the two point clouds to establish candidate correspondences. To estimate a robust rigid transformation, Random Sample Consensus (RANSAC) [3] is employed. RANSAC iteratively samples minimal sets of correspondences (typically three), computes transformation hypotheses, and selects the one that results in the highest number of inliers—i.e., correspondences that are spatially consistent under the proposed transformation. This process yields a coarse alignment that can be refined with the Iterative Closest Point (ICP) [13] algorithm.

PREDATOR is a deep learning-based model designed for pairwise registration of 3D point clouds, particularly effective in scenarios with low overlap between scans. The core innovation lies in its overlap-attention module, which enables early information exchange between the latent representations of the two point clouds. This mechanism allows the model to predict per-point overlap and matchability scores, effectively identifying regions that are both salient and common to both point clouds. By focusing on these overlapping regions, PREDATOR enhances the selection of interest points for matching. The architecture employs a combination of graph neural networks (GNNs) and transformer-based cross-attention mechanisms to refine feature descriptors conditioned on both point clouds. PREDATOR demonstrates promising results in low-overlap scenarios. I have leveraged an existing model, however the PREDATOR model can be fine trained and fine-tuned for specific scenes.

3. Method

To motivate the proposed improvement, we first revisit the ConceptGraphs [4] pipeline in more detail. ConceptGraphs incrementally constructs a 3D semantic scene graph from a sequence of RGB-D frames. Each object in the scene is represented by a 3D point cloud and a semantic feature vector (language embedding). The core processing steps, illustrated in Figure 1, are as follows:

1. **Class-agnostic 2D Segmentation:** Each RGB frame is processed by a generic segmentation model (e.g., SAM) to extract object masks. These regions are embedded using CLIP [9] to obtain semantic features, then projected into 3D and transformed into the global

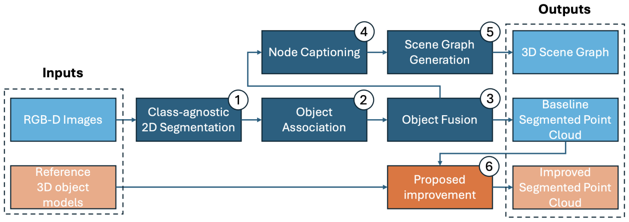

Figure 1. Overview of the ConceptGraphs method (dark blue) with input and output data (light blue). Note the placeholder for the proposed method (darg orange) and its input and outputs (light orange).

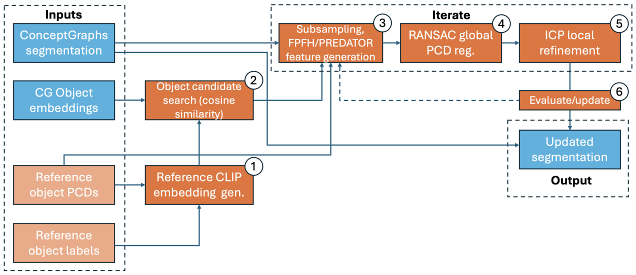

Figure 2. The flow of the proposed method. Notice the input data dependencies and the iteration loop.

map frame using depth and pose data.

2. **Object Association:** New segments are matched to existing objects using a similarity score combining geometric similarity (via DBSCAN clustering and nearest neighbor statistics) and semantic similarity (via cosine distance between CLIP features). Hungarian assignment is used to associate detections with objects.

3. **Object Fusion:** When a segment matches an existing object, its feature vector and point cloud are merged into the object representation. Feature embeddings are updated using a weighted average, and redundant 3D points are downsampled.

4. **Node Captioning:** For each object node, top-10 image crops are selected and passed to a vision-language model (LLaVA) with prompts like d̈escribe the central object in the image.̈ Candidate captions are refined using GPT-4 to produce final descriptions.

5. **Scene Graph Generation:** Edges between objects are inferred by computing 3D bounding box IoUs to form a dense graph. A minimum spanning tree (MST) is used for pruning. Edges are annotated using an LLM with relational prompts (e.g., Ä on B ̈) to infer spatial and semantic relationships.

In downstream applications such as robotic task planning, the final scene graph is converted into structured JSON representations containing each object's caption and 3D pose. These are passed to an LLM to interpret user queries and trigger corresponding robotic actions (e.g., grasping or navigation).

Note the placeholder of the proposed method in the baseline Conceptgraphs flow on 1 with its inputs and outputs.

3.1. Proposed Method: Object-Specific Point Cloud Refinement

This project focuses on improving the quality of the segmented 3D map by leveraging known reference objects. Specifically, a curated set of reference 3D object models and labels is used to register and refine the corresponding segments in the ConceptGraphs-generated map, enhancing completeness and accuracy. While reference objects could also be used to improve scene graph generation, this project focuses solely on the mapping aspect.

The proposed pipeline builds upon the existing ConceptGraphs scene graph and its object-wise language embeddings. The input includes the initial segmented map, per-object semantic embeddings, and a list of known reference object models and labels. As illustrated in Figure Y, the refinement process consists of the following steps:

1. **Reference Feature Generation:** For each reference object, a semantic embedding is generated using CLIP—either from the object label or from renderings of the reference 3D model.

2. **Candidate Matching:** Cosine similarity is computed between the reference embedding and all object embeddings in the scene graph. A similarity threshold (empirically set to 2.7) is used to identify candidate matches for each reference object.

3. **Feature Extraction:** For each candidate-reference pair, the corresponding point clouds are subsampled, and features are extracted using either FPFH or PREDATOR.

4. **Initial Registration (RANSAC):** Global registration is performed using RANSAC to align the candidate and reference point clouds based on their features.

5. **Refinement (ICP):** The alignment is refined using the Iterative Closest Point (ICP) algorithm.

6. **Fitness Evaluation and Iteration:** Open3D's `evaluate_registration` function is used to assess the alignment via the fitness metric, which measures the proportion of matched points within a threshold distance. If the fitness exceeds a set threshold (e.g., 0.7), the registration is accepted and the map is updated. Otherwise, additional iterations are attempted using alternative parameters. If all attempts fail, the original map object is retained.

3.2. Registration Methods: FPFH and PREDATOR

Both FPFH and PREDATOR are employed for global registration. FPFH is a fast, handcrafted descriptor encoding local geometric features via histograms of angular relationships. It is lightweight and effective for coarse alignment but lacks semantic understanding and struggles with low-overlap or noisy data. PREDATOR is a learned feature extractor using graph neural networks and attention mechanisms to reason about overlap and matchability between point clouds. It produces context-aware features and performs well under challenging conditions but is more computationally intensive.

Each method requires tuning of point cloud downsampling parameters. For the Open3D-based FPFH implementation (based on an Open3D example I have customized [1]), the `voxel_size` controls downsampling granularity. For the PREDATOR implementation that I have reworked, `voxel_down_sample` serves a similar role. Smaller voxel values preserve fine detail and are suitable for intricate objects, while larger values improve robustness and performance for large or noisy scenes. The optimal value depends on object size, shape complexity, and scan density. In this implementation, hardcoded configurations are used, though an adaptive approach could be explored in future work.

3.3. Implementation Strategy

Note, that the registration methods are non-deterministic, and depending on the object sizes, different parameters and methods perform better. Thus, my registration procedure iterates through multiple parameter combinations. FPFH-based registration is tested with `voxel_size` values of 0.02 and 0.1. PREDATOR-based registration is tested with `voxel_down_sample` values of 0.025 and 0.08. Each configuration is attempted up to 3 times. After each attempt, the fitness score is computed. If it exceeds 0.7, the transformation is accepted and the corresponding reference object is integrated into the scene. If no configuration meets the threshold, the original object remains unchanged.

4. Dataset

This project utilizes the **Replica Dataset** [11], a high-quality synthetic indoor dataset that provides ground-truth 3D point clouds, per-instance semantic segmentation labels, and rendered RGB-D image sequences. These features make it well-suited for evaluating the ConceptGraphs pipeline and the proposed refinement method. Figure X illustrates the `room0` scene from the dataset, showing both an RGB image and its corresponding semantic segmentation.

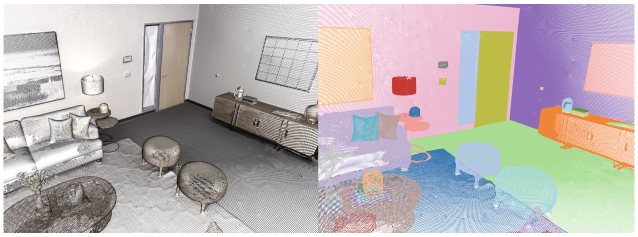

Figure 3. The `room0` scene from the Replica dataset with ground truth segmentation on the right.

To enable the evaluation of my method, the Replica dataset is preprocessed in the following steps:

1. **Baseline Scene Reconstruction:** First, I generate the baseline segmented 3D maps using the ConceptGraphs pipeline. These serve as the input to my method. Figure 4 shows example reconstructions and segmentations produced by ConceptGraphs for the “room0” scene. As shown, many objects are only partially reconstructed due to limited and suboptimal camera viewpoints—an issue frequently encountered in real-world robotics and AR scenarios.

2. **Ground-Truth Subset Filtering:** Due to limitations in ConceptGraphs’ ability to reconstruct large planar surfaces like walls, I focus the evaluation on reconstructing and segmenting foreground objects such as furniture and appliances. I preprocess the ground-truth point cloud by excluding structural elements like walls, doors, windows, and vents. The resulting filtered subset serves as a reference for evaluating segmentation performance. An example is shown in Figure 5.

3. **Reference Objects and Labels:** I manually select a subset of reference objects from each scene using the ground-truth segmented point cloud. These objects are centered and randomly rotated to provide a realistic registration challenge. Each selected object is also paired with a reference label (e.g., *chair*, *table*), which is used for language-based matching in the proposed pipeline. An example is shown in Figure 5.

To assess the robustness of the reconstruction process, I also experiment with varying levels of RGB-D frame subsampling. While the Replica dataset provides up to 2000 RGB-D frames per scene, ConceptGraphs typically operates on a default subset of 200 frames. In my experiments, I adopt this default as the baseline and additionally test subsampling rates of 1:2 and 1:4 (i.e., 100 and 50 frames, respectively). These variations allow me to evaluate how both the baseline ConceptGraphs method and the proposed reference-object-based enhancement perform under reduced visual input conditions.

Scenes `room0` and `room2` were selected for evaluation. Both contain suitable reference objects, while `room2` poses

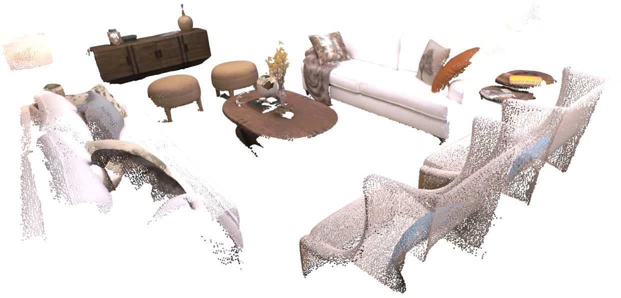

Figure 4. Mapping results from the ConceptGraphs method. Note the incomplete reconstructions.

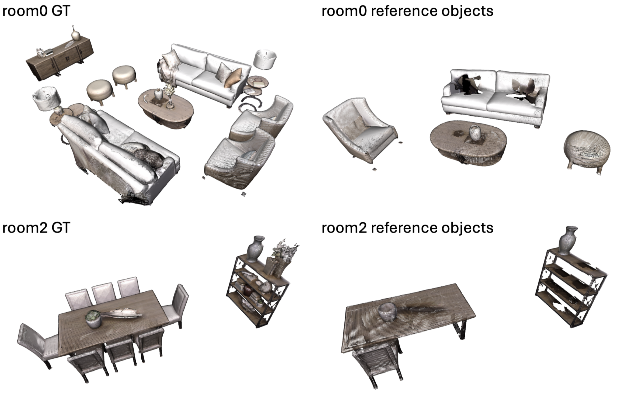

Figure 5. Ground truth scene (after removing the walls, ceiling and floor) and reference objects for the Replica room0 and room2 scenes. Note, that reference objects are centered and randomly rotated before use.

a greater challenge due to incomplete reconstructions—e.g., chairs often lack their lower parts—providing a useful contrast to the higher-quality results in `room0`.

5. Experiments

To evaluate the mapping and segmentation performance of both the baseline and the improved methods, I leverage the reference object point clouds and corresponding ground-truth segmentations. An evaluation script was developed to compute a set of metrics, including mean per-object IoU, global IoU, the number of unmatched ground-truth segments, and the number of orphan mapped segments.

5.1. Evaluation Metrics

The metrics are defined as follows:

- **Mean Per-Object IoU:** For each ground-truth object, the Intersection over Union (IoU) is computed with the mapped segment that has the highest overlap. Associations are one-to-one, and unmatched ground-truth objects receive an IoU of 0. The mean of these values is reported. This metric captures object-level accuracy but can be skewed by missed small objects.
- **Global IoU:** Computed as the total volume of intersection between all matched segments divided by the total union of ground-truth and predicted segments. This metric reflects overall scene-level reconstruction quality and is particularly sensitive to errors in large objects.
- **Unmatched Ground-Truth Segments:** These are ground-truth objects that are not associated with any predicted segment. They often correspond to small or occluded objects that were not separated during segmentation.
- **Orphan Mapped Segments:** Predicted segments that are not matched to any ground-truth object (i.e., they are not the best match for any object). These typically represent redundant or spurious segmentations. Although they could potentially be filtered using prior knowledge about known objects, such filtering is left for future work.

5.2. Experimental Setup

The experiments are structured as follows:

1. **Baseline Evaluation:** The default ConceptGraphs pipeline is evaluated using its standard input of 200 subsampled RGB-D frames. Additional evaluations are performed at 1:4 and 1:8 subsampling rates to assess robustness under reduced visual input. Baseline result are on 1.
2. **Combined Registration Method:** A hybrid approach is also evaluated, in which all configurations are attempted adaptively. This method is tested on all baseline input variants to assess overall performance. Evaluated using its standard input of 200 subsampled RGB-D frames. Additional evaluations are performed at 1:4 and 1:8 subsampling rates to assess robustness under reduced visual input. The combined method results are on 1.
3. **Single Registration Methods:** Each proposed registration method is evaluated independently compared to the baseline and combiend method:
   - **FPFH-based registration:** Tested with `voxel_size` values of 0.02 and 0.1.
   - **PREDATOR-based registration:** Tested with `voxel_down_sample` values of 0.025 and 0.08.

For each reference object, registration attempts are retried up to three times per configuration. The results are on 2.

Table 1 summarizes the results across all configurations and subsampling levels.

Table 1. Evaluation metrics across different input subsampling rates comparing the baseline ConceptGraph mapping to the combined method for Replica scenes `room0` (r0) and `room2` (r2). Metrics include mean IoU (mIoU), global IoU (gIoU), number of unmatched ground-truth objects (Unm.), and number of orphan segments (Orph.)

Method	**mIoU**		**gIoU**		**Unm.**		**Orph.**	
	r0	r2	r0	r2	r0	r2	r0	r2
Baseline	0.41	0.24	0.49	0.48	11	13	7	3
Baseline 1:4	0.40	0.24	0.47	0.50	10	14	3	3
Baseline 1:8	0.33	0.24	0.42	0.51	13	14	8	4
Combined	**0.49**	**0.34**	**0.76**	**0.65**	**10**	**13**	**3**	**3**
Combined 1:4	0.49	0.31	0.75	0.60	11	14	7	3
Combined 1:8	0.44	0.25	0.75	0.52	13	14	8	4

Table 2. Evaluation of individual registration configurations. Metrics include mean IoU (mIoU), global IoU (gIoU), number of unmatched ground-truth objects (Unm.), and number of orphan segments (Orph.).

Method	**mIoU**	**gIoU**	**Unm.**	**Orph.**
Baseline	0.41	0.49	11	7
Combined	**0.49**	**0.75**	**11**	**7**
FPFH 0.02	0.44	0.70	13	8
FPFH 0.10	0.45	0.69	13	8
PREDATOR 0.025	0.44	0.71	13	8
PREDATOR 0.08	0.44	0.72	13	8

Qualitative results along with visualized ground truth overlap are depicted on Figure 6 and Figure 7.

The results in Table 1 suggest that reference object registration significantly improves overall mapping and segmentation performance—particularly in terms of the global IoU (gIoU). This improvement is most evident when large reference objects are present, as gIoU is strongly influenced by the number of correctly reconstructed points. This is an important objective for general physical environment understanding. In contrast, improvements in mean per-object IoU (mIoU) are more limited in scenes that contain many small, non-reference objects. Notably, a lower mIoU is often correlated with a higher number of unmatched ground-truth segments, indicating that undetected small objects remain a challenge. Addressing this would require improving the baseline method's ability to detect and segment non-reference objects.

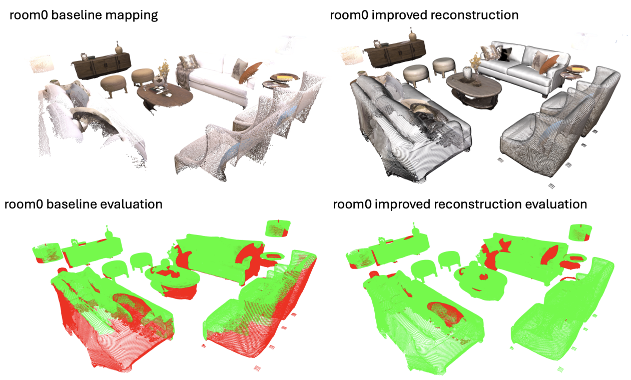
room0 baseline mapping
room0 improved reconstruction
room0 baseline evaluation
room0 improved reconstruction evaluation

Figure 6. Replica room0 qualitative results of the baseline and improved method on the top row. Notice how the incomplete point cloud mapping has been corrected by the improved method. On the bottom, the intersection with the ground truth is visualized.

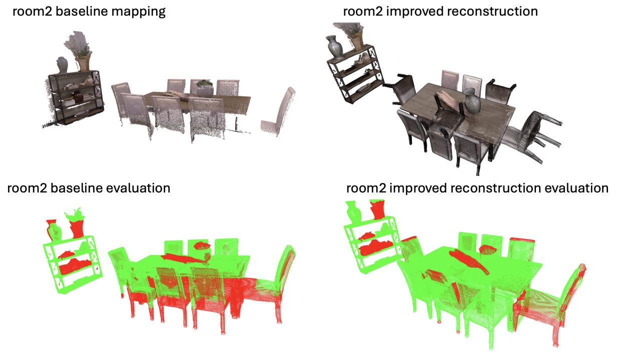
room2 baseline mapping
room2 improved reconstruction
room2 baseline evaluation
room2 improved reconstruction evaluation

Figure 7. Replica room2 qualitative results of the baseline and the partially improved method on the top row. Notice the large missing parts from the chairs and how that throws off the method displacing some of the chairs. On the bottom, the intersection with the ground truth is visualized.

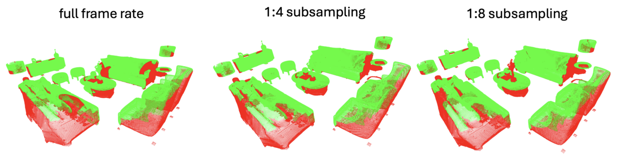

Figure 8. The effect of frame rate subsamplimg on the baseline method. Less drastic than expected.

Interestingly, both the baseline and the improved methods demonstrate a degree of robustness to frame subsampling. See visualization of Figure 8. While performance metrics do decrease with fewer input frames, even the 1:8 subsampling ratio results in only moderate degradation. Although some objects and segments are missed and reference-based registrations become more prone to failure, the overall drop in performance remains manageable.

While orphan segments are tracked, I have not yet implemented a filtering mechanism to suppress redundant or overlapping segments—especially those that duplicate reference objects. Developing such a filter is a promising direction for future work and could further enhance segmentation quality.

There is also a noticeable performance gap between scenes `room0` and `room2`, particularly when visualizing the outputs. The baseline performance in `room2` is relatively poor (e.g., mIoU of 0.24), and the reference-based registration methods struggle due to minimal geometric overlap between the partial reconstructions and the reference models.

Table 2 supports these observations. While all registration methods contribute to improvements over the baseline, each method fails to register certain objects in isolation. The combined method, which leverages multiple configurations, consistently achieves the best performance, validating its robustness in diverse scenarios.

6. Conclusion

In this project, I demonstrated that point cloud mapping and segmentation can be significantly improved through the integration of reference object point clouds, supporting the initial hypothesis. This form of exact object matching holds strong potential for enhancing downstream processes in scene graph construction frameworks such as ConceptGraphs—for example, by enabling the injection of object-specific metadata (e.g., affordances) directly into the graph.

That said, the current method would benefit from further tuning, particularly with regard to threshold selection and robustness across varied scenes. Domain-specific fine-tuning of components like PREDATOR could also yield notable performance gains. As future work, I plan to improve the adaptability and parameterization of the system and explore how richer textual metadata associated with reference objects could further enhance scene graph generation in ConceptGraphs.

References

[1] Open3d documentation, global registration.

[2] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner. Scan2cad: Learning cad model alignment in rgb-d scans, 2018.

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

[4] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[5] S. Huang, Z. Gojcic, M. Usvyatsov, and K. S. Andreas Wieser. Predator: Registration of 3d point clouds with low overlap. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

[6] S. Koch, N. Vaskevicius, M. Colosi, P. Hermosilla, and T. Ropinski. Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[7] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023.

[8] OpenAI. Gpt-4 technical report, 2024.

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[10] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, 2009.

[11] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[12] C. Zhang, A. Delitzas, F. Wang, R. Zhang, X. Ji, M. Pollefeys, and F. Engelmann. Open-Vocabulary Functional 3D Scene Graphs for Real-World Indoor Spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[13] J. Zhang, Y. Yao, and B. Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.