MastSAM: Solving Multi-view Inconsistency Segmentation by Sequence Matched 3D Coordinates

*Note: Sub-titles are not captured in Xplore and should not be used

Anonymous Authors

***Abstract*—With the emerging importance of understanding 3D environments, such as spatial intelligence and 3D foundation models, researcbers have sought to distill knowledge from off-the-shelf 2D foundation models such as CLIP and SAM. However, these 2D foundation models often produce inconsistent information across different views. To tackle this issue, we present MastSAM. This method leverages Mast3R's ability to map 2D pixel coordinates from image pairs into a shared 3D space. By doing so, MastSAM enables consistent tracking of corresponding points across multiple views so that 2D foundation models such as SAM can output multi-view-consistent segmentation. Our main contributions are as follows: 1) Clearly defining the multi-view inconsistency problem in 2D foundation models. 2) Proposing a novel solution to minimize the multi-view inconsistency problem using MastSAM.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Recent advances in 2D foundation models such as CLIP and SAM have greatly simplified traditional 2D vision tasks. Meanwhile, 3D awareness, perception, and understanding have emerged as central areas of focus in computer vision. However, several challenges arise in 3D, with data scarcity and diversity being at the forefront. Compared with 2D images, 3D data are more difficult to collect. Although large-scale outdoor scene datasets [1], indoor scene datasets [2], [3], object-level datasets [4], [5], and part-level datasets [6] suffice for demonstration-level experiments, they still lack generality for open-vocabulary settings. Furthermore, the diversity of 3D representations such as meshes, point clouds, and occupancy grids makes it difficult to devise a universal model architecture.

A common approach is to distill information from 2D foundation models (e.g., CLIP or SAM) into 3D. Early work by [7] uses a radiance-field representation to learn language embeddings for each patch, while [8] distills knowledge from [9] to achieve 3D-aware segmentation. However, due to limitations of radiance fields' implicit representations, direct 3D manipulation is not possible. Instead, one must instead render images back to 2D to produce language embeddings and segmentation masks. With the emergence of Gaussian Splatting [10]—an explicit and compact representation—several works [11]–[13] have demonstrated that lifting 2D knowledge into 3D can significantly improve 3D perception.

Despite these developments, multi-view inconsistency remains a major challenge. As shown in Fig.1, such inconsistencies occur in both object-level and scene-level segmentation, ultimately causing semantic corruption in 3D. For example in Fig.1 (a), the bear's nose, eyes, and feet are segmented separately in one view, but merged into a single mask in another. This issue is also evident in simpler video segmentation tasks where SAM often fails to maintain consistent masks across adjacent frames, leading to increasing errors with larger viewpoint changes. Fig.1 shows two adjacent frames from the Davis dataset [14]—used here as inputs for both our MastSAM model and the original SAM model [9].

Fig. 1. (a) Multi-View Inconsistency Example (b) Video Sequence Case: A simpler form of multi-view inconsistency appears in video sequences, where adjacent frames exhibit small camera pose changes. (c) Results of the original SAM: The figure shows SAM's outputs on the two frames. Same-colored regions share the same mask ID. SAM fails to consistently match objects across views, splitting or merging them differently in each frame. (d) Results of MastSAM: Our MastSAM method assigns the same mask ID (color) to the same object across frames, effectively correlating masks across different views.

Identify applicable funding agency here. If none, delete this.

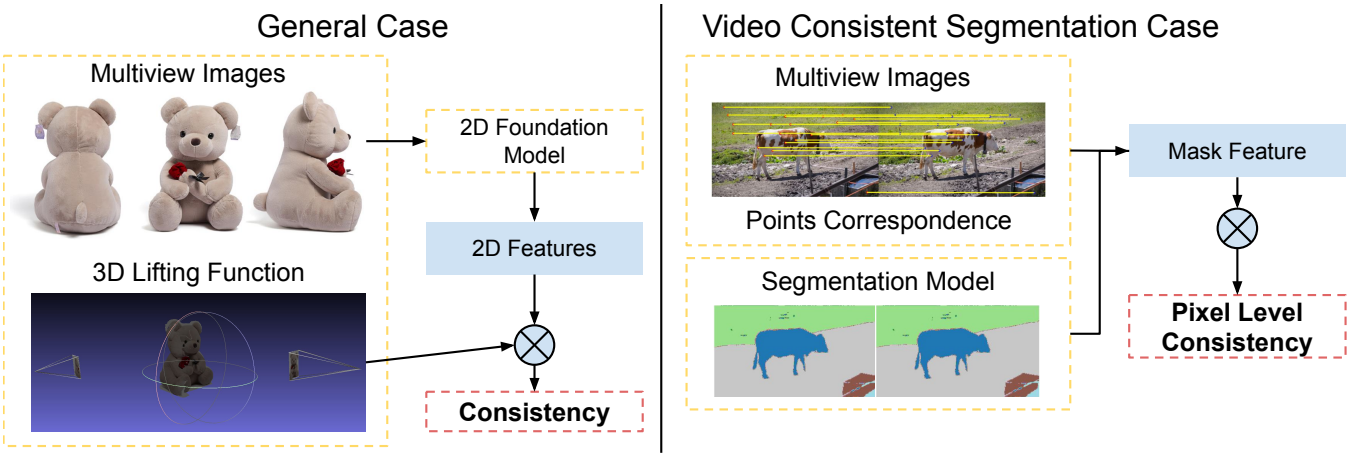

Fig. 2. The input of the measurement metrics are multi-view images, and a 3D lift function that can match images to 3D location, as well as the 2D foundation model we want to measure. The basic idea is that the output of 2D foundation model points to the same 3D location should be the same. The cross function here calculate the average cosine similarity between output features that at the same 3D coordinates.However, in reality, it would be impossible to calculate pixel by pixel due to the computational burden. In current setting, we then simplify the calculation granularity from per pixel calculation to per-mask. However, this per-mask calculation is a good approximation to segmentation problem. We each mask from adjacent picture from a $H \times W$ representation to a consistent feature representation. And calculate the feature vector's similarity

To address this, we formally define the multi-view inconsistency problem and propose a baseline solution. Our method, MastSAM, leverages Mast3R [15] to establish point-to-point correspondences. We initialize masks using SAM's Auto-Mask-Generator by selecting representative points from each mask. We then track these points in subsequent frames according to Mast3R's guidance. During this procedure, we iteratively introduce new masks in previously unmasked regions. As shown in Fig.1, MastSAM effectively mitigates multi-view inconsistency. Our key contributions are threefold:

- **Formal Definition:** We formally define multi-view inconsistency in 3D segmentation.
- **New Metric:** We propose the first metric to quantitatively measure multi-view consistency.
- **MastSAM Algorithm:** We present MastSAM and demonstrate its upper-bound performance on the proposed metric.

In the following sections, we will first define the problem and introduce related work. Then we will introduce our method, show metrics, and evaluate both qualitative and quantitative experiment results.

II. Multi-view Inconsistency

In this section, we formally define the multi-view inconsistency problem. We start with an intuitive explanation, and then mathematically define it. As the name suggests, there are several 2D images where the information corresponding to the same 3D location are inconsistent among all of the 2D views. As shown in the Fig.2.

Formally, we have a image set $\mathcal{I}$, where $\mathcal{I} = I_1, I_2, ..., I_n$. We also define a pixel space $\mathbb{P}$ for each image i as shown in Equ.1. Where W_i and H_i is the height and width of image i.

$$\mathbb{P}_i = \{(x, y) | x \in 1, 2, \ldots, W_i, y \in 1, 2, \ldots, H_i\} \quad (1)$$

We then define a function $\mathcal{F}$ that can generate a function f_i that maps pixel space to consistent coordinates space, i.e. $\mathbb{P} \to \mathbb{R}^3$ as shown Equ.3

$$f_i : \mathbb{P}_i \to \mathbb{R}^3, f_i(x, y) = (x', y', z') \quad (2)$$

$$\mathcal{F} : I \times \mathcal{I} \to (\mathbb{P}_i \to \mathbb{R}^3), F(i, \mathcal{I}) = f_i \quad (3)$$

Given above equation, we define the multi-view consistency function h_w as a mapping from a pixel space to a score, i.e. $\mathbb{P}_w \to \mathbb{R}$. It specifically define as shown in Equ.6. Notice that the multi-view consistency is used to measure the power of a model instead of measuring the dataset we use. Therefore, we currently regard function $\mathcal{F}$ as a perfect function that can output right 3D coordinates.

$$\epsilon_w(x, y) = \left\{(i, j, k) \in \mathcal{I} \times \mathbb{P}_i \;\middle|\; \|f_w(x, y) - f_i(j, k)\| \leq \epsilon\right\} \quad (4)$$

$$h_w(x, y, M) = \frac{\sum_{(i,j,k) \in \epsilon_w(x,y)} \frac{M_i(j,k) \cdot M_w(x,y)}{\|M_i(j,k)\|_2 \cdot \|M_w(x,y)\|_2}}{|\epsilon_w(x, y)|} \quad (5)$$

$$H(\mathcal{I}, \mathcal{M}) = \frac{1}{|\{I, \mathbb{P}_i\}|} \sum_{(w,x,y) \in \{I, \mathbb{P}_i\}} h_w(x, y, M) \quad (6)$$

As we shown in the above equation Equ.6, we first find all other pixel in different images within a reasonable range to the input pixel on the corresponding 3D space. This reasonable range means that those pixel are point at almost the same 3D location in 3D space. And then, for each pixel we calculate the cosine similarity with the input pixel, Finally the weighted average is the result of the consistency score of that pixel using current 2D foundation model M.

Although the above metrics would reliable and accurate to measure multi-view inconsistency, it would be impossible to calculate the metrics due to computational complexity.

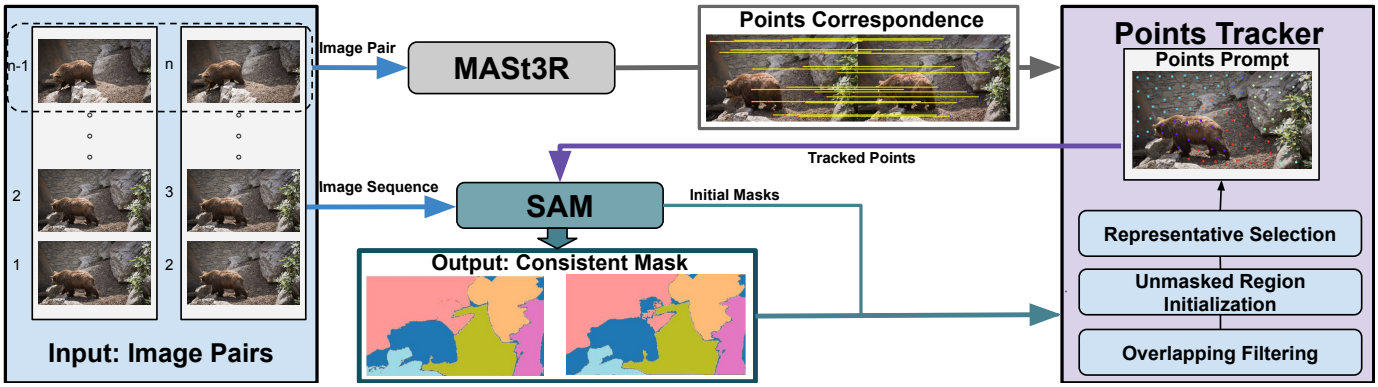

Fig. 3. Overview of our approach. Given a pair of image sequences, one sequence is input to SAM, and an image pair is input to MASt3R. SAM produces a set of image masks, which is processed by the Prompt Points Tracker. The output of MASt3R is a set of correspondence points, which is also input to the Prompt Points Tracker. The Prompt Points Tracker applies overlapping filtering, decreasing the number of masks. The Point Tracker then initializes the unmasked regions, increasing the number of masks back to dynamic equilibrium from the previous step, and selects representative points to ensure multi-view consistency. Each group of points is then input back into SAM to produce the final segmented image, aiming to address the multi-view consistency problem. The next frame of masks is sent to the Points Tracker to generate consistent masks for subsequent frames.

Therefore, an easier approximation might vary from a different down-stream task. The previous multi view inconsistency measurement metrics seem to work fine but there are some issues, especially using the metrics as a loss function, known as the degeneration problem. When all output of a model become the same, the inconsistency becomes zero, but loses the original function of model. Therefore, it would be unreasonable to use these metrics. They should be used together with other downstream tasks, such as segmentation accuracy. In the experiment session, we will explain how we implement an approximation of above metrics to down stream tasks such as SAM and CLIP.

III. Related work

A. Image Matching

Image matching aims to establish correspondences between pixels across different images of the same scene, capturing global spatial consistency. Previous works have utilized keypoint-based matching, connecting sparse, locally invariant features between images. Traditional methods, such as those based on epipolar geometry [16]–[18], enforce spatial constraints by leveraging geometric relationships between camera views. Handcrafted approaches like SIFT [19], [20] achieve image matching through robust local feature extraction. Modern methods like SuperGlue [21] enhance image matching using graph-based attention, improving robustness under challenging conditions. However, their reliance on keypoint-based matching limits their effectiveness in handling extreme viewpoint changes.

In contrast, recent advances such as DUSt3R [22] and MASt3R [15] use dense matching, establishing correspondences for all pixels. In addition, they treat image matching as a 3D problem, centered around camera pose and scene geometry. DUSt3R redefines pairwise reconstruction as the regression of 3D pointmaps [22], achieving robustness to viewpoint and illumination changes without relying on explicit matching supervision. Building on DUSt3R, MASt3R significantly improves the accuracy of pairwise matches and the reciprocal matching speed by adding a feature matching module that aligns dense local features in 3D space [15]. By using dense matching and grounding spatial consistency in a 3D framework, MASt3R addresses key limitations of traditional and modern methods, making it a robust and efficient solution for image-matching tasks.

B. Image segmentation

Segment anything(SAM) [9] is a zero-short image segmentation model that inputs points or bounding boxes and outputs a corresponding segmentation mask. Built on the Vision Transformer (ViT) architecture [**?**] and trained on the large-scale SA-1B dataset [9], SAM demonstrates remarkable segmentation performance for single-image tasks.

Extending SAM's [9] capabilities to video segmentation, recent approaches have emerged. SAM2 [23] introduces frame-by-frame segmentation using a sparse attention mechanism to efficiently track temporal changes . While this approach adapts well to frame variations, it relies heavily on the quality of initial prompts and lacks robust mechanisms for continuity tracking across frames. Similarly, the Track Anything Model (TAM) [24] addresses the need for consistent segmentation by combining SAM's high-quality segmentation with XMem's [25] memory mechanism. TAM integrates SAM [9] for precise mask initialization and refines with XMem for semi-supervised video object segmentation. On the other hand, the matching Anything by Segment Anything (MASA) [26] model focuses on tracking multiple moving objects through bounding boxes, utilizing SAM and various data transformers [26]. Despite these advancements, current video segmentation models face several limitations. Their performance tends to decline over time, especially during long video sequences, as segmentation accuracy heavily depends on the quality of the initialization. Poor-quality initial masks can significantly impair the results. Most critically, these works treat each frame as an independent unit, providing individually optimized outputs with a

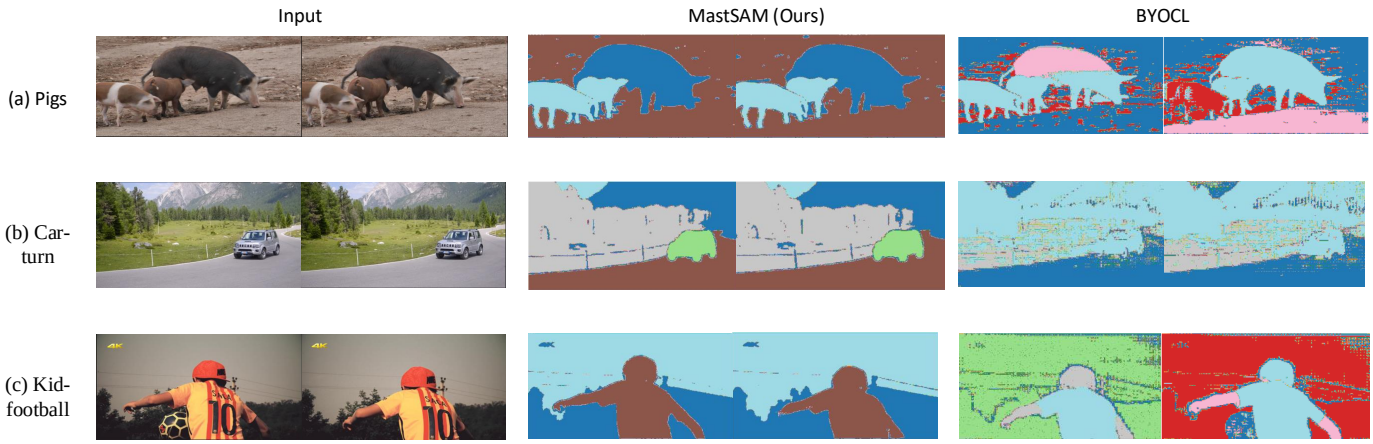

Fig. 4. This figure presents the comparison between Mast-SAM and BYOCL in different data sequences. For each data sequence, the left-most image is the original image, the middle image is the output of Mast-SAM, and the right-most image is the output of BYOCL. As one can see that the MastSAM can correlate previous image and current image together. This is shown by the same color on correlated mask. Although BYOCL method can sometime correlate the consistency well, but it lose its ability to correct segment the original mask.

limited understanding of overall continuity. This frame-centric approach often fails to capture global spatial relationships, making them susceptible to challenges like object occlusion, rotation, or viewpoint transformation.

To address these limitations, our work combines SAM [9] with MASt3R [15], leveraging MASt3R's ability to robustly capture spatial continuity across frames. By grounding segmentation in a 3D perspective, our approach ensures robust performance even when the camera poses changes, bridging the gap between frame-by-frame optimization and spatially consistent video segmentation.

IV. Methods

Our method consists of three main components: MASt3R, SAM, and the Points Tracker. These modules interact with each other in a sequence designed to ensure multi-view consistency and produce accurate, dynamically adjusted segmentation masks across all image frames.

A. Input and Pre-processing

The input comprises of sequential image pairs drawn from a multi-frame dataset, such as [14] [2], which is representative of typical video. This approach can be easily adapted to [3] and other 3D scan datasets. Each pair is structured for simultaneous processing by two pipelines. One pipeline is aimed at correspondence detection by MASt3R and the other pipeline is focused on mask consistency produced by SAM. It is significant to note that the dataset is processed for uniform resolution and aligned using intrinsic and extrinsic camera parameters for efficient downstream processing.

B. MASt3R for Points Correspondence

MASt3R takes each input image pair and identifies sparse, but accurate correspondences across the frames. It leverages feature extraction through transformer-based models and cross-image attention mechanisms. Features between images are aligned, matching scores are computed, and correspondences are filtered based on confidence thresholds. These correspondence points are one of the two essential inputs for initializing the Points Tracker.

MASt3R outputs correspondence maps, which are then further refined by combining reciprocal matches, ensuring point consistency.

C. SAM for Mask Consistency

Simultaneously, the SAM module processes each image frame in the image sequence to generate segmentation masks. This allows SAM to adapt the segmentation task to produce consistent masks for regions of interest in the image sequence. SAM accounts for the dynamic nature of the sequence by integrating initial masks and point maps provided by the Points Tracker in subsequent iterations. This feedback loop ensures that the masks are constantly evolving with the updated tracked points.

D. Points Tracker for Multi-view Refinement

The Points Tracker serves as the critical intermediary for ensuring multi-view consistency and accurate propagation of masks across the sequence. Its core components include:

- **Overlapping Filtering:** The Points Tracker module applies a post-processing step to remove overlapping masks using geometric constraints and consistency checks, ensuring each segment remains well-defined and unique.
- **Representative Selection:** Correspondence points identified by MASt3R are refined through clustering (K-Means) and confidence filtering to select representative points for multi-view alignment.
- **Unmasked Region Initialization:** Regions outside the initial mask coverage are dynamically identified and initialized, ensuring that new regions entering the view are incorporated. This step is crucial to ensure we have an equilibrium in the number of input and output masks,

especially after the Overlapping Filtering step reducing the number of masks in the module.

These steps optimize camera parameters, 3D depth maps, and point alignments iteratively to ensure robust multi-view consistency.

E. Pipeline Integration

The outputs from the MASt3R and SAM modules feed into the Points Tracker, which generate refined masks that are then re-inputted into SAM. This iterative process allows for feedback-driven mask updates, with MASt3R ensuring that tracked points maintain temporal and spatial consistency. Leveraging the synergy between segmentation and correspondence tracking, this pipeline effectively addresses challenges in dynamic multi-view sequences.

The final output is a set of consistently aligned segmentation masks and a reconstructed 3D point cloud for the scene. These outputs enable robust segmentation and analysis of multi-view dynamic datasets.

V. Experiment and Results

A. Introduction to experiment

To extensively evaluate MastSAM, we performed experiments on two open-source datasets: the Davis benchmark [14] and Mose.

The ground truth annotation in both datasets only refer to one mask of the major object, whereas MastSAM automatically outputs an image possessing masks of every object. This discrepancy made direct comparisons between MastSAM and the ground truth annotations impractical. To address this, we developed the following evaluation protocol to ensure a comprehensive and fair assessment of MastSAM's performance.

- **Ground Truth Replication:** We replicated the ground truth (GT) annotation for each annotation image as many times as the number of masks generated by MastSAM.
- **Best Mask Selection:** For each predicted mask, we calculated the Intersection over Union (IoU) with the corresponding GT annotations. The mask with the highest IoU value among all predicted masks for each image was selected.
- **Metrics Computation:** Using the best mask for each image, we computed the following evaluation metrics: IoU, F1 score, precision, and recall. These metrics collectively offer a comprehensive evaluation of the regional and boundary precision of MastSAM.

Furthermore, we adopted the parallel evaluation method to facilitate running speed and GPU space. Four GPUs were used following the 'queue' logic. Once a GPU was set free, a new unprocessed sequence would be sent to this GPU. With this logic, MastSAM was able to segment 100 images consistently within 15 minutes.

B. experiment results

1) Quantitative Results: Compared to BYOCL, our method demonstrates significant improvements across all metrics, highlighting its superior segmentation performance. Although the IoU and F1 score of our method are slightly lower than those of SAM1, our approach achieves higher temporal consistency between consecutive frames, as evidenced by a more balanced Precision and Recall. The specific quantitative results are shown in the tableI below:

TABLE I
PERFORMANCE METRICS OF DIFFERENT METHODS ON THE DAVIS DATASET.

Method	IoU	F1	Precision	Recall	Consistency
BYOCL	0.3906	0.527	0.4621	0.6671	XX.XX
Ours	0.4553	0.5228	0.5226	0.5988	92.32
THM	0.6787	0.786	0.9891	0.6965	NA

These results underscore the effectiveness of our method in addressing segmentation consistency across frames. By leveraging MastSAM's ability to ensure multi-view consistency, our approach achieves a robust trade-off between accuracy and temporal stability, which is essential for video segmentation tasks.

2) Visualization Results: As shown in Fig.4 Fig.5, our method qualitatively much better compare to BYOCL.

C. Ablation

TABLE II
ABLATION BETWEEN NO UNMASKED REGION INITIALIZATION AND FULL CAPACITY

Method	IoU	F1	Precision	Recall
Ours w/o initialization	0.3678	0.4211	0.5682	0.3780
Ours	0.4553	0.5228	0.5226	0.5988

In this part, we show the essence of our design logic. Since our method's natural of expanding mask to the mask nearby, it will lose its ability to segment a meaningful result. This is due to all mask will be ultimately aggregate into one results. To prevent this degenerate situation, we import a counter measure by iteratively segment unsegmented area using original SAM model. In this way, we stabilize the segmentation mask number. Our ablation study shown in the Tab.II display our assumption clearly.

VI. Conclusion

In this work, we formalize the concept of multi-view inconsistency in 3D segmentation and propose the first dedicated metric for quantifying consistency across views. We introduce MastSAM, an algorithm that achieves theoretical upper-bound performance on our metric, demonstrating its potential for robust multi-view analysis. However, our current framework has limitations: (1) While MastSAM provides strong empirical results, further experiments on diverse benchmarks are needed to generalize its efficacy; (2) The computational complexity of our pipeline necessitates task-specific adaptations for downstream applications, which may limit plug-and-play usability; (3) Our experiments focus on video sequences, leaving open questions about performance in static multi-view settings (e.g., ScanNet, indoor scenes). Future work will address these challenges by optimizing computational efficiency and extending evaluations to broader 3D understanding tasks and datasets.

Fig. 5. In here, we propose more comparison regarding the segmentation result instead of complexity result.However, our major focus in current paper is on how to solve the consistency problem, the downstream task result should be the secondary consideration

REFERENCES

[1] B. Xiong, N. Zheng, and Z. Li, "Gauu-scene v2: Expanse lidar image dataset shows unreliable geometric reconstruction using gaussian splatting and nerf," *arXiv preprint arXiv:2404.04880*, 2024.

[2] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[3] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "Scannet++: A high-fidelity dataset of 3d indoor scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12–22.

[4] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 142–13 153.

[5] M. Xu, P. Chen, H. Liu, and X. Han, "To-scene: A large-scale dataset for understanding 3d tabletop scenes," in *European Conference on Computer Vision*. Springer, 2022, pp. 340–356.

[6] A. Delitzas, A. Takmaz, F. Tombari, R. Sumner, M. Pollefeys, and F. Engelmann, "Scenefun3d: Fine-grained functionality and affordance understanding in 3d scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 531–14 542.

[7] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 729–19 739.

[8] C. M. Kim, M. Wu, J. Kerr, K. Goldberg, M. Tancik, and A. Kanazawa, "Garfield: Group anything with radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 530–21 539.

[9] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[10] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[11] B. Xiong, X. Ye, T. H. E. Tse, K. Han, S. Cui, and Z. Li, "Sa-gs: Semantic-aware gaussian splatting for large scene reconstruction with geometry constrain," *arXiv preprint arXiv:2405.16923*, 2024.

[12] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan, "Language embedded 3d gaussians for open-vocabulary scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5333–5343.

[13] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 051–20 060.

[14] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[15] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," in *European Conference on Computer Vision*. Springer, 2025, pp. 71–91.

[16] Y. Bhalgat, J. F. Henriques, and A. Zisserman, "A light touch approach to teaching transformers multi-view geometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4958–4969.

[17] Y. He, R. Yan, K. Fragkiadaki, and S.-I. Yu, "Epipolar transformers," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2020, pp. 7779–7788.

[18] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 757–774.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[21] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[22] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 697–20 709.

[23] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[24] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng, "Track anything: Segment anything meets videos," *arXiv preprint arXiv:2304.11968*, 2023.

[25] H. K. Cheng and A. G. Schwing, "Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model," in *European Conference on Computer Vision*. Springer, 2022, pp. 640–658.

[26] S. Li, L. Ke, M. Danelljan, L. Piccinelli, M. Segu, L. Van Gool, and F. Yu, "Matching anything by segmenting anything," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 963–18 973.