Panoptic Lifting for 3D Scene Understanding with Neural Fields – Supplementary Document –

Yawar Siddiqui[1,2] Lorenzo Porzi[2] Samuel Rota Bulò[2]
Norman Müller[1,2] Matthias Nießner[1] Angela Dai[1] Peter Kontschieder[2]
Technical University of Munich[1] Meta Reality Labs Zurich[2]

In this supplementary document, we discuss additional details about our method Panoptic Lifting. Specifically. in Section 1 we give additional details about our test time augmentation algorithm. A comparison of rendering performance of our method compared to the baselines is reported in Section 2. We also provide implementation details of our method and the baselines (Section 3), the data used for experiments in the main paper (Section 4), and limitations (Section 6). Finally, we report additional metrics, scene Segmentation Quality ($SQ^{\texttt{scene}}$) and Retrieval Quality ($RQ^{\texttt{scene}}$), in Section 5.

1. Test-time Augmentation for Mask2Former

In this section we describe the test-time augmentation strategy we adopt to obtain improved panoptic segmentation masks and per-pixel confidence scores from Mask2Former [2].

1.1. Test-time Augmentation

We run a pre-trained Mask2Former network on multiple augmented versions of each input image, using the following set of transformations: horizontal flip, rescale, contrast, RGB-shift, random gamma, random brightness & contrast, median blur, sharpen, and arbitrary combination of the previously mentioned augmentations. For each transformation, we intercept the Mask2Former outputs before its "panoptic fusion" stage, *i.e.* right after the transformer and pixel decoders (see Sec.3 of [2] for details). These outputs consist of a set of candidate segments, represented as 2D soft masks paired with probability distributions over the classes. After transforming the candidate segments back to the original image resolution and orientation, our next objective is to fuse them into a single, coherent panoptic segmentation.

1.2. Fusing Mask2Former predictions

We denote the candidate segments predicted from all augmented versions of the image as a set of pairs $(\mathsf{m}_i, \mathbf{p}_i), i = 1, \dots, N$, where $\mathsf{m}_i(x, y) \in [0, 1]$ is the predicted probability of pixel (x, y) to belong to segment i, and $\mathbf{p}_i = [p_i^1, ..., p_i^C]$ is the segment's predicted probability distribution over C classes. In the following, we describe a mechanism to combine these predictions into a single panoptic segmentation with associated confidences, following three steps: segment clustering, cluster aggregation and panoptic fusion.

Segment clustering. We build a graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, ..., N\}$, and $\mathcal{E} = \{(i, j) | i, j \in \mathcal{V} \wedge s(i, j) \geq \theta\}$. The matching function $s(i, j)$ is defined as a "soft-IoU"

$$s(i,j) = \frac{\sum_{x,y} \min(\mathsf{m}_i(x,y), \mathsf{m}_j(x,y))}{\sum_{x,y} \max(\mathsf{m}_i(x,y), \mathsf{m}_j(x,y))},$$

and θ is a matching threshold (e.g. $\theta = 0.5$). In other words, we add an edge between two segments if their soft-IoU is greater than θ. By finding the connected component of this graph, we partition the segments into clusters $\mathcal{K} \subset \mathcal{V}$.

Cluster aggregation. After clustering the segments, we define a new set of masks and class probabilities, this time associated with clusters instead of segments. We denote these as $\hat{\mathsf{m}}_\mathcal{K}(x, y)$ and $\hat{\mathbf{p}}_\mathcal{K} = [\hat{p}_\mathcal{K}^1, ..., \hat{p}_\mathcal{K}^C]$, respectively, and compute them by simply averaging the masks and probabilities of all segments belonging to each cluster

$$\hat{\mathsf{m}}_\mathcal{K}(x,y) = \frac{1}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \mathsf{m}_i(x,y),$$

$$\hat{\mathbf{p}}_\mathcal{K} = \frac{1}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \mathbf{p}_i.$$

Panoptic fusion. Given this new set of masks and class probabilities, we fuse them into a single overall panoptic prediction with an algorithm akin to the one used in the final stage of [2]. Specifically, we follow these steps:

1. For each cluster $\mathcal{K}$, we determine the most likely class $c_\mathcal{K}^* = \arg\max_c \hat{p}_\mathcal{K}^c$, and the corresponding probability $p_\mathcal{K}^* = \max_c \hat{p}_\mathcal{K}^c$.

2. We scale $\mathcal{K}$'s mask by $p_\mathcal{K}^*$ to obtain $\bar{\mathsf{m}}_\mathcal{K}(x, y) =$

Method	Time to render 2048 rays
PNF [5]	119.7 ms
DM-NeRF [11]	66.5 ms
Semantic-NeRF [12]	65.7 ms
Panoptic Lifting (Ours)	**13.1** ms

Table 1. Time taken to render a batch of 2048 rays on a NVIDIA RTX A6000 GPU.

Method	HyperSim [8]	Replica [10]	ScanNet [3]
Mask2Former [2]	50.52	50.10	43.6
Panoptic Lifting (Ours)	**66.84**	**63.79**	**60.4**

Table 2. Conventional PQ scores on novel views from the test set.

$p^*_{\mathcal{K}} \hat{\mathsf{m}}_{\mathcal{K}}(x, y)$.

3. We assign image pixels to clusters with the rule: (x, y) is assigned to $\mathsf{k}^*(x, y) = \arg\max_{\mathcal{K}} \bar{\mathsf{m}}_{\mathcal{K}}(x, y)$, and its confidence is set to $\mathsf{s}(x, y) = \max_{\mathcal{K}} \bar{\mathsf{m}}_{\mathcal{K}}(x, y)$.

At the end of this process, each pixel will have a class $c^*_{\mathsf{k}^*(x,y)}$ and a confidence $\mathsf{s}(x, y)$. Furthermore, pixels of thing classes can be partitioned into instances according to their cluster assignment $\mathsf{k}^*(x, y)$.

2. Rendering Performance

Tab. 1 compares the time taken to render a batch of 2048 rays for each method on an NVIDIA RTX A6000 GPU. Due to the hybrid representation from TensoRF, our model delivers a faster rendering performance compared to the baselines.

3. Implementation Details

3.1. Panoptic Lifting

Panoptic Lifting uses TensoRF [1] for modeling the scene density and radiance. Specifically, we use the Vector-Matrix (VM) decomposition, with number of density and appearance components set to 16 and 48 respectively. The starting grid resolution is set to 128^3 and goes upto 192^3 at the end of the optimization. 27 color features are decoded with a tiny 2 layer MLP with positional encoding with 2 components to encode the view direction and the features.

To model the semantic class distribution and surrogate identifiers we make use of two small view-independent MLPs. The semantic MLP has 5 layers with a width of 256 and outputs a probability distribution over the target classes for any given input position. The surrogate identifier is a 3 layer MLP which generates a distribution over max k identifiers (set to 50 in our experiments). Neither of these MLPs use positional encoding. We choose to go with MLPs instead of Vector-Matrix decompositions for semantics and surrogate identifiers for memory size constraints. Our model is trained with a batch of 2048 rays, with a learning rate of 0.0005 for MLPs and 0.02 for the TensoRF lines and planes.

3.2. Baselines

We use the publicly available Mask2Former [2] code and models, without any retraining or fine-tuning. For all methods that use Mask2Former instance labels (including ours), instance counts are renumbered to be distinct across frames. For Semantic-NeRF [12] and DM-NeRF [11], we use their publicly released code. Since DM-NeRF outputs the labels as abstract instance identifiers, we create a map from instance to class using the instance's majority class across the train set as its assigned class.

Since Panoptic Neural Fields [5] does not provide a public implementation, we re-implement it based on details from the paper. We do not use their prior-based initialization since it requires additional 3D datasets for the instanced classes. In the original implementation, PNF uses a monocular 3D detector, which is essential when dealing with dynamic objects varying across frames. However, since the task here deals with static scene, it is more fair to use a multi-view detector for getting the bounding boxes. We use a state-of-the-art multiview detector [9] pretrained on ScanNet for getting object bounding boxes for PNF in our experiments. Note that for getting a reasonable 3D detector performance, it is required that the camera poses are scaled and centered similarly to the original ScanNet training data. We perform these pose corrections for Replica [10], Hypersim [8] and in-the-wild scenes. Since this correction requires an estimate of scale, we use for pose correction the ground-truth depth from Replica and Hypersim, and NeRF optimized depth for scenes in the wild. We further show result with a variant of PNF that uses ground-truth detections, except for in-the-wild data where not ground-truth is available.

All models (including ours) are trained with Mask2Former [2] generated labels.

4. Data

Tab. 4 shows the scenes and their corresponding number of frames. The available posed images are split into 75% views for training and 25% intermediately sampled test views. Note that for each of the datasets, the ground-truth semantic and instance labels are only used for evaluation, and are not used for training or refinement of any models.

Since the original model (swin_large_IN21k) was trained on COCO [7], and the labels for evaluation come from different datasets, we map the Mask2Former predictions as

Method	HyperSim [8]		Replica [10]		ScanNet [3]	
	SQ^{scene} ↑	RQ^{scene} ↑	SQ^{scene} ↑	RQ^{scene} ↑	SQ^{scene} ↑	RQ^{scene} ↑
DM-NeRF [11]	62.06	55.45	58.68	47.68	53.26	46.13
PNF [5]	55.33	47.51	53.62	44.10	62.96	50.73
PNF [5] + GT Bounding Boxes	68.23	53.35	62.15	50.81	70.01	55.87
Panoptic Lifting (Ours)	**70.35**	**64.32**	**69.10**	**63.61**	**73.50**	**64.95**

Table 3. SQ and RQ metrics for on novel views from the test set.

Dataset	Scene	# Frames
HyperSim	ai_001_003	100
HyperSim	ai_001_008	100
HyperSim	ai_001_010	300
HyperSim	ai_008_004	63
HyperSim	ai_010_005	100
HyperSim	ai_035_001	200
ScanNet	scene0050_02	874
ScanNet	scene0144_01	678
ScanNet	scene0221_01	780
ScanNet	scene0300_01	929
ScanNet	scene0354_00	563
ScanNet	scene0389_00	708
ScanNet	scene0423_02	855
ScanNet	scene0427_00	659
ScanNet	scene0494_00	740
ScanNet	scene0616_00	758
ScanNet	scene0645_02	726
ScanNet	scene0693_00	866
Replica	office_0	900
Replica	office_2	900
Replica	office_3	900
Replica	office_4	900
Replica	raw	900
Replica	room_0	900
Replica	room_1	900
Replica	room_2	900
In the wild	office	1100
In the wild	bed_room	1100
In the wild	meeting_room	1100

Table 4. Scenes used for evaluations in our experiments. Note that the in the wild scenes are only used for qualitative evaluation (shown in the supplementary video) since ground truth labels are not available for a qualitative comparison with baselines.

Class	Type
wall	Stuff
floor	Stuff
cabinet	Stuff
bed	Thing
chair	Thing
sofa	Thing
table	Stuff
door	Stuff
window	Stuff
counter	Stuff
shelves	Stuff
curtain	Stuff
ceiling	Stuff
refridgerator	Thing
television	Thing
person	Thing
toilet	Thing
sink	Thing
lamp	Stuff
bag	Thing
otherprop	Stuff

Class	Type
wall	Stuff
floor	Stuff
cabinet	Stuff
bed	Things
chair	Things
sofa	Things
table	Stuff
door	Stuff
window	Stuff
counter	Stuff
shelves	Stuff
curtain	Stuff
ceiling	Stuff
refridgerator	Things
television	Things
person	Things
toilet	Things
sink	Things
lamp	Stuff
bag	Things
bottle	Things
cup	Things
keyboard	Things
mouse	Things
book	Things
laptop	Things
blanket	Stuff
pillow	Things
clock	Stuff
cellphone	Things
otherprop	Stuff

Table 5. Classes and their type (*stuff* or *thing*) for dataset experiments (left) and in the wild experiments (right).

well as the ground-truth labels across all the datasets used in our experiments to ScanNet 21 classes (Tab. 5 left). For in the wild scenes, we use 31 ScanNet classes listed in Tab. 5 (right).

5. Additional Results

Tab. 2 reports the conventional PQ scores between our method and Mask2Former [2]. As mentioned in the main paper, this does not take into account the instance consistency across the scene, since matching between ground-truth and predicted instances is done on a per-frame basis. We further report SQ`scene` and RQ`scene` in Tab. 3.

6. Limitations

Panoptic Lifting shows considerable improvements over the state of the art; however, several limitations remain. Our method uses predictions from a pre-trained panoptic segmentation model, and hence is limited to classes with which the original model was trained. In this context, it would be interesting to explore open world segmentation [4, 6] with self-supervised instance clustering. Further, unlike PNF [5], we cannot handle per dynamic objects and can only work with static scenes. Similar to other NeRF-based approaches, our method is currently run offline due to lengthy pre-processing for pose estimation, 2D segmentation inference, and neural field optimization; here, a promising avenue would be to integrate our approach with state-of-the-art SLAM approaches that run in real-time.

Furthermore, consistently mislabeled 2D instances will translate to incorrect 3D panoptics (*e.g.*, supplementary video at 3:07, stacked chairs are predicted as a single instance since 2D segmentation consistently predict them as such). Indeed, similar to existing NeRF-based works, we need many frames to get good depth estimates, in the absence of which panoptic fusion does not work well. We also do not incorporate recent improvements from the NeRF literature like robustness to camera pose noise, anti-aliasing, background modeling, *etc*. Consequently, our method directly leverages improvements of 2D panoptic segmentation and NeRF methods.

References

[1] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[2] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1, 2, 4

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 3

[4] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021. 4

[5] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 2, 3, 4

[6] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 4

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[8] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10912–10922, 2021. 2, 3

[9] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2397–2406, 2022. 2

[10] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 3

[11] Bing Wang, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022. 2, 3

[12] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 2