Grounding Pretrained Features in 3D Representations

Kenneth Blomqvist[1], Francesco Milano[1], Jen Jen Chung[2], Lionel Ott[1] and Roland Siegwart[1]

Abstract— **Pretraining deep neural networks has become very popular and has led to the recent trend of foundation models. For perception, pretraining has mostly been constrained to 2D feature learning. 3D representation learning has yet to have its breakthrough moment. Data is more heterogeneous and harder to come by. 3D learning algorithms are still behind their 2D counterparts and the right 3D self-supervised learning objectives are yet to be discovered.**

In this paper, we take a look at a recent trend in 3D representation learning where features extracted from 2D images are grounded into a 3D representation through a 3D feature field. We discuss recent results, highlight some open problems in the field and suggest some potential avenues to solve these problems.

I. INTRODUCTION

With the rise of large-scale pre-training of foundation models in computer vision, methods have been proposed to ground the representations produced by such models into a 3D representation that could be leveraged by robots.

Directly learning 3D representations has proven hard, as learning algorithms that work directly on 3D representations are challenging to train and methods working on 2D vision tasks face many subtle issues when extending to 3D. One challenge with 3D representation learning, is that there are many very different ways of representing 3D data. A common choice is a global point cloud of the entire scene. A full reconstruction might not be available at runtime and large ones are hard to process. Others include voxelized representations, which can be expensive to process and have limited fidelity. Yet another option are partial representations, such as point clouds captured from individual viewpoint, which have to then be aggregated to get a persistent representation.

The dominance of 2D representation learning has a lot to do with the fact that we have extremely widely used standards for representing RGB images, such as JPEG. Images are widely supported by the billions of devices in every person's pocket and are shared at unprecedented rates on the internet. As these pictures are shared with some semantic context, we can train vision-language models [1] to infer semantic information from images. This vision-language pre-training paradigm has been shown to be much more effective than previously dominant ImageNet pre-training [2]. This is because the datasets involved are so much larger and they contain much more diverse examples since they aren't constrained by expert annotation time and can simply be collected automatically from the web. Another challenge facing 3D representations is that 3D sensors are still very much developing and differences in characteristics between one sensor and the next are much larger than between RGB cameras. An automotive LiDAR will produce very different output than the time-of-flight sensors embedded within high-end smartphones which again produce different output than structured-light based sensors. Learning on one of these does not necessarily transfer well to the other.

In robotics, 2D representations only take you so far. Robots operate in 3D environments and need to be able to reason about the geometry of a scene and plan through the representation to solve tasks. Some success has been achieved by training visuomotor policies [3], [4], bypassing the 3D representation problem. Such methods can work well, but even these methods might benefit from 3D representations, should useful ones become available.

Recently, neural implicit feature fields have emerged as a way to take 2D features and ground them in a 3D representation, encoded by a neural implicit function which is learned by minimizing a photo- and featuremetric loss with the optional addition of direct depth supervision. Feature fields were initially introduced by [5] and [6] as a method to decompose, segment and edit neural radiance fields. They have since been extended to automate interactive 3D segmentation [7], [8].

To enable CLIP-like semantic vision-language queries on point clouds, a similar quasi-feature field idea was introduced by CLIP-Fields [9]. [10] tackles the same problem, but learns to fuse features onto points using a 3D point cloud network. These point cloud based open vocabulary scene querying methods present some promising results, but require a separate step to estimate the point cloud. Neural implicit methods have been developed that can jointly learn the scene geometry and with a semantic feature field [11], [12]. The feature outputs, enable zero-shot segmentation, object detection, retrieval and more. [13] similarly explored grounding vision-language features on 3D representations, but by optimizing features in a parameterized voxel grid. On top of the feature grid, they define a set of operators to find, count and reason about objects in the scene. [14] uses a similar approach to learn multi-modal maps, combining LSeg features with audio and object information. Other work on grounding features onto 3D representations include [15]–[17].

The focus of this paper is to specifically highlight and discuss the open problems we have identified during our work in developing 3D feature fields. A summary of the general methodology and our experimental setup are provided

[1] Autonomous Systems Lab, Swiss Federal Institute of Technology in Zürich, Switzerland. kblomqvist@mavt.ethz.ch

[2] School of ITEE, The University of Queensland, Australia.

This project has received funding from EU Horizon 2020 program, project PILOTING H2020-ICT-2019-2 871542.

in Sections II and III; the interested reader is directed to [11] for full details. The core contribution of this paper is Section IV, which offers an in-depth analysis of the limitations, challenges and future possibilities of these methods.

II. Vision-Language Feature Fields

At the core of feature fields is a NeRF-like [18] neural implicit representation with an additional feature output in the MLP. The representation consists of two parameterized parts, a positional encoder and a multi-layer perceptron, which maps encodings to color, density and feature outputs. The multi-layer perceptron maps the encoded position to density σ, color $\mathbf{c}$ and a feature output $\mathbf{f}$.

The feature outputs are learned in the same away as the color using volumetric rendering and a transmittance function computed from the density output of the MLP. To produce rendered quantities from the MLP outputs, we define a rendering function R:

$$R(\mathbf{r}, h) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i \delta_i))h(\mathbf{x}_i), \tag{1}$$

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \tag{2}$$

where h is a function outputting a vector or scalar quantity for points $\mathbf{x}_i$ within the volume, T_i is the transmittance function, δ_j is the distance between samples and σ_i is the predicted density for encoded point samples $\mathbf{x}_i$ along a ray $\mathbf{r}$. We use R to produce aggregated outputs for each ray:

$$\begin{aligned} \hat{\mathbf{c}}(\mathbf{r}) &= R(\mathbf{r}, \mathbf{c}), \\ \hat{d}(\mathbf{r}) &= R(\mathbf{r}, z), \\ \hat{\mathbf{f}}(\mathbf{r}) &= R(\mathbf{r}, \mathbf{f}), \end{aligned} \tag{3}$$

Here, z is the depth along the ray.

To encode positions, in our framework, we use hybrid positional encoding [7]. Hybrid positional encoding combines the low frequencies of frequency encoding [18] with the hierarchical volumetric grid of parameters introduced by [19]. The volumetric parameters are learned jointly with the MLP parameters to fit a scene.

Finally, to learn the representation, we minimize a combined photometric, depth and feature error, by minimizing the L^2 distance between rendered color $\hat{\mathbf{c}}$ and ground truth color $\bar{\mathbf{c}}$, L^1 loss betweeen rendered depth $\hat{d}$ and ground truth depth $\bar{d}$ and L^2 loss between rendered feature $\hat{\mathbf{f}}$ and extracted features $\bar{\mathbf{f}}$. We minimize this loss by randomly sampling mini-batches of pixel locations and corresponding rays from input images, mimimizing parameters using stochastic gradient descent.

As such a framework is capable of making use of arbitrary pixel-aligned feature maps, to create vision-language feature fields, we choose to use learned features for which the similarity with text prompts can be computed through a simple dot product. For example, LSeg [20] is suitable for this purpose. Once trained, the feature field can be used for downstream tasks such as object discovery, segmentation and planning tasks. In particular, since the vision-language features of LSeg apply open vocabulary training, feature fields trained with these targets can work with natural language class prompts at runtime, by assigning feature field features to the closest text prompt embedding encoded with the LSeg text encoder.

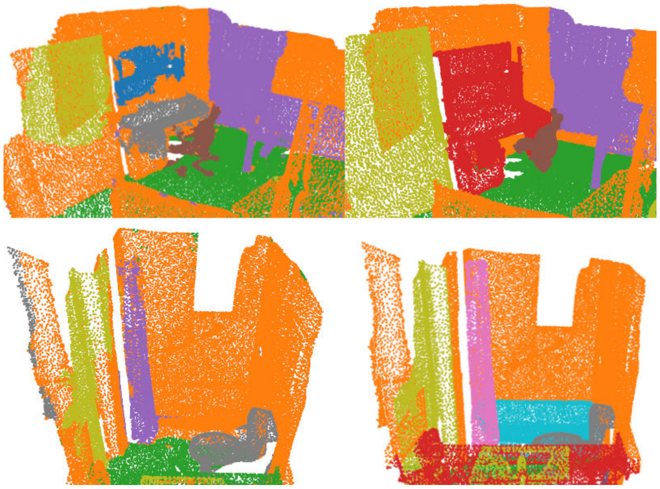

Fig. 1. The left column shows predictions and the right side ground truth annotations for point clouds in the ScanNet validation set.

III. Experiments

We run some experiments using LSeg features [20] on the validation set of the ScanNet dataset [21] using the 20 evaluation classes to test the 3D open-vocabulary segmentation performance of the resulting representation. We first train our representation on each scene for 20 000 iterations on color, depth and extracted features. We then evaluate against the ground truth point cloud by querying our representation at each point and comparing against the ground truth label.

Table I shows the mIoU agreement with the ScanNet validation set segmentation labels. As can be seen, the method performs well on many of the classes, but completely fails on others. Specifically, the "desk" class is almost always classified as "table" and similarly, "shower curtain" as the "curtain" class. Figure 1 shows a qualitative example of this misalignment between the features and the dataset semantics. The bathtub visible in the bottom right and labeled as "other furniture", is also missed by our approach.

IV. Open Problems

A. *Learning good pixel-aligned features*

A key component of learning good feature fields is having high quality pixel-aligned features to supervise on. Not only do these features have to capture the desired object or semantic properties, they ideally would be viewpoint invariant to facilitate the convergence of the learned feature field.

In the case of open-vocabulary scene understanding, the current state-of-the-art pixelwise methods, are fine-tuned on closed-set datasets, reversing a lot of the learning that

mIoU top classes				mIoU bottom classes				mIoU all
Floor	Bed	Wall	Chair	Picture	Desk	Other furniture	Shower curtain	Mean
75.8	66.3	61.5	60.2	2.4	1.5	0.1	0.0	47.4

TABLE I

Mean intersection over union on ScanNet validation set for best and worse classes.

was done on the open-set pre-training task [17], [22]. As can be seen in Table I, while the LSeg [20] features do reasonably well on most classes, they completely fail on others. Additionally, similar classes are easily confused, which is likely due to the bias induced by the ADE20K [23] dataset on which LSeg is fine-tuned. Some of the confusion between classes might be possible to fix by aligning the query embeddings for each class on some labeled examples.

Methods such as CLIP [24] learn their visual models through global vector valued embeddings which are correlated against natural language feature embeddings. While this type of supervision is readily available from datasets scraped from the internet, they do not learn dense visual features which could be directly segmented or grounded in a feature field. Some methods have opted to solve this problem via multi-scale fusion and effectively convolving the learned CLIP visual encoder over the image at multiple scales [12]. This has proven to work reasonably well, but does come with a high upfront preprocessing cost and doesn't currently lend itself to real-time methods. Others have studied how to learn more fine-grained outputs using a CLIP style contrastive objective [25], [26]. Such methods have yet to be demonstrated for higher output resolutions, but with some further improvements, might lead to scalable self-supervised methods which yield pixel-aligned features that are robust in the long tail of classes that are not present in curated, closed-set datasets. Combining self-supervised objectives with knowledge distilled from more specialized models [27] could be a path forward. [1] provides a comprehensive survey on the current state of vision-language pretraining.

Non-feature-based open-set semantic segmentation methods such as X-Decoder [28], which uses a transformer to produce the final segmentation given visual and text encodings, could potentially also be adapted to 3D representations.

B. *Learning downstream layers*

Foundation models have proven useful for other 2D computer vision tasks, especially for zero and few-shot learning. Feature fields present the opportunity to apply these few-shot learning methods in the 3D domain. At its most simple, this means learning downstream layers on top of the 3D feature field to tackle few-shot object detection, segmentation, object re-identification and other tasks from a few sparse examples. Such methods could either operate on individual features of the feature field or they could make predictions over a neighborhood of features. [29] already learned grasp detection on top of NeRF. Segmenting instances of objects reliably from each other remains unsolved. Panoptic Lifting [30] defines surrogate ID outputs to a radiance field which are learned through a remapping of the possibly noisy 2D instance labels to achieve 3D instance segmentation. For open-set vision-language feature fields, object instance is query dependent, complicating the representation learning and run-time 3D inference task. For example, the instance segmentation is different if the scene is queried for "couch" vs. "sofa cushion".

C. *Having the 3D supervise the 2D*

Recently we have seen how the representation learned by neural implicit methods can be used to supervise object descriptor learning [31] and stereo matching [32]. Feature maps rendered from 3D feature fields could be used as a regression target to either learn 3D-aware or viewpoint-invariant versions of the original 2D feature maps. Alternatively, they could be used to distill the information into more efficient, embedded versions of the heavy foundation models used to learn the original 3D representations.

D. *Real-time deployment on robots*

The core challenges for deploying feature fields onboard real robots are largely related to computation. While current state-of-the-art NeRF implementations are able to run in real-time on high-end workstations [19], they consume a lot of computational resources. When adding additional feature heads and models to extract features from incoming images, the cost increases and takes further floating point operations to learn.

E. *Learning representations incrementally*

This brings us to the related issue of how to build scene representations incrementally. Current neural implicit implementations all jointly optimize both volumetric parameters and the MLP parameters. As the MLP parameters are constantly updated, the full buffer of keyframes used for the scene have to be kept around and sampled from. As the MLP parameters change over the course of the optimization, the volumetric parameters change their meaning. This could be solved by pretraining the rendering network MLP on a few scenes, and only optimizing the volumetric parameters on subsequent scenes. This would mean that once a certain region of the volume has been optimized to convergence, the images and feature maps used could be discarded to cap the memory use.

There are cases where retrieving previously collected views of a part of the scene might be useful, such as when the loop is closed and the robot returns to observe an area of the scene which was previously viewed. In such a case, retrieving the frames which observed that part of the scene from a different viewpoint could yield a better representation.

Another possible approach would be to not directly optimize the latent volumetric parameters through the MLP, but actually predict them or otherwise optimize them given the recently collected viewpoints. This could yield benefits such as being able to infer unobserved geometry and semantics, but might result in lower fidelity scene representations and could cause hallucinations. At the very least, this could be done for unobserved parts of the scene, to guess what might be there before observing it, as has been done in traditional volumetric mapping [33].

F. Integrating with SLAM

Typical neural implicit scene implementations assume ground truth camera poses, which in practice are often computed using COLMAP [34]. In a real-time system, these have to be estimated on the fly. We have obtained reasonable results by naïvely integrating with state-of-the-art sparse visual-inertial SLAM systems, but some drift inevitably happens causing error to accumulate in the map [11]. Neural implicit SLAM systems have been developed [35], [36] which directly estimate camera poses through a neural implicit representation. Using the features which are learned in the feature field (semantic or otherwise) could help refine poses in addition to the typical photometric and geometric losses, akin to what is done in [37]. Hybrid systems combining the benefits of traditional sparse feature-based and neural implicit SLAM could also be a promising direction.

G. Dealing with dynamic scenes

Dynamic scenes remain a challenge for neural implicit representations. In order to deal with changes in the scene, neural implicit representations either have to forget outdated information or changes need to be explicitly tracked. Optimizing the volumetric representation on a sliding window of frames might be feasible for coarse changes in the scene, but tracking more real-time effects such as humans or dynamic objects remains difficult. Some works have explored composing multiple neural radiance fields [38]–[40], which combined with real-time pose estimation could yield a high fidelity object-level SLAM system. The pose estimation itself could be done using a feature field representation of the objects of interest. Something like this is already explored by [41]. Furthermore, in a neural implicit object-level SLAM system, dealing with noisy segmentation and object detection remains challenging [42].

H. Planning through feature fields

An important component of high-level task and motion planning from natural language instructions, is disambiguating objects from one another [43]–[45]. Many high-level planning frameworks assume a rich and accurate scene representation is available [46].

The rich semantic scene understanding capabilities promised by feature field-based methods offer unprecedented opportunities for high-level planning. Being able to query a scene using arbitrary text queries could help solve open-ended planning problems. Some early results have already been shown for planning from natural language prompts [47], and language informed navigation [9], [48]–[53]. Previous natural language grounded planners have made use of object detectors [47], [54]. To our knowledge, no previous effort has yet tackled planning through semantic feature fields.

V. CONCLUSIONS

As we have detailed in this short report, neural implicit feature fields are an exciting and emerging tool that enables a number of downstream applications. We presented some results we have obtained using our vision-language feature field framework. We highlighted many of the shortcomings we identified in our work as well as in related methods. We discussed several promising and exciting future directions and open problems in the field. We also believe that there are likely to be many more applications that we could not foresee here and we look forward to reading about them.

The general framework of feature fields that we detailed can be used with any type of pixel-aligned features. We showed that this framework provides very good zero-shot scene understanding performance in the form of segmentation. Given the pace of innovation in the visual representation learning community in the past years, we expect orders of magnitude better pixel-aligned visual-language models to be become available in the next couple of years. We therefore predict that the static case of semantic scene understanding will be very close to solved within a couple years time. We also believe that foundation model approaches that distill information from massive weakly supervised datasets will surpass current supervised methods which are learned on these small scale 3D datasets.

REFERENCES

[1] J. Zhang, J. Huang, S. Jin, and S. Lu, “Vision-language models for vision tasks: A survey,” *arXiv preprint arXiv:2304.00685*, 2023.

[2] N. Mu, A. Kirillov, D. Wagner, and S. Xie, “Slip: Self-supervision meets language-image pre-training,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pp. 529–544, Springer, 2022.

[3] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[4] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, “Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1549–1556, 2019.

[5] S. Kobayashi, E. Matsumoto, and V. Sitzmann, “Decomposing NeRF for Editing via Feature Field Distillation,” in *NeurIPS*, 2022.

[6] V. Tschernezki, D. Larlus, and A. Vedaldi, “Neuraldiff: Segmenting 3d objects that move in egocentric videos,” in *2021 International Conference on 3D Vision (3DV)*, pp. 910–919, IEEE, 2021.

[7] K. Blomqvist, L. Ott, J. J. Chung, and R. Siegwart, “Baking in the Feature: Accelerating Volumetric Segmentation by Rendering Feature Maps,” *arXiv preprint arXiv:2209.12744*, 2022.

[8] K. Mazur, E. Sucar, and A. J. Davison, “Feature-Realistic Neural Fusion for Real-Time, Open Set Scene Understanding,” *IEEE ICRA*, 2023.

[9] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, “CLIP-Fields: Weakly Supervised Semantic Fields for Robotic Memory,” *arXiv preprint arXiv:2210.05663*, 2022.

[10] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, “OpenScene: 3D Scene Understanding with Open Vocabularies,” *arXiv preprint arXiv:2211.15654*, 2022.

[11] K. Blomqvist, F. Milano, J. J. Chung, L. Ott, and R. Siegwart, "Neural implicit vision-language feature fields," 2023.

[12] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," *arXiv preprint arXiv:2303.09553*, 2023.

[13] Y. Hong, C. Lin, Y. Du, Z. Chen, J. B. Tenenbaum, and C. Gan, "3d concept learning and reasoning from multi-view images," *arXiv preprint arXiv:2303.11327*, 2023.

[14] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," *arXiv preprint arXiv:2303.07522*, 2023.

[15] H. Ha and S. Song, "Semantic Abstraction: Open-World 3D Scene Understanding from 2D Vision-Language Models," in *Conference on Robot Learning*, 2022.

[16] Y. Hong, Y. Du, C. Lin, J. Tenenbaum, and C. Gan, "3d concept grounding on neural fields," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7769–7782, 2022.

[17] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Omama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, *et al.*, "ConceptFusion: Open-set Multimodal 3D Mapping," *arXiv preprint arXiv:2302.07241*, 2023.

[18] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *ECCV*, 2020.

[19] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," *arXiv preprint arXiv:2201.05989*, 2022.

[20] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven Semantic Segmentation," *arXiv preprint arXiv:2201.03546*, 2022.

[21] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," in *IEEE/CVF CVPR*, 2017.

[22] Y. Ding, L. Liu, C. Tian, J. Yang, and H. Ding, "Don't stop learning: Towards continual learning for the clip model," *arXiv preprint arXiv:2207.09248*, 2022.

[23] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene Parsing through ADE20K Dataset," in *IEEE CVPR*, 2017.

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," in *International Conference on Machine Learning*, 2021.

[25] L. Yao, R. Huang, L. Hou, G. Lu, M. Niu, H. Xu, X. Liang, Z. Li, X. Jiang, and C. Xu, "Filip: fine-grained interactive language-image pre-training," *arXiv preprint arXiv:2111.07783*, 2021.

[26] K. Ranasinghe, B. McKinzie, S. Ravi, Y. Yang, A. Toshev, and J. Shlens, "Perceptual Grouping in Vision-Language Models," *arXiv preprint arXiv:2210.09996*, 2022.

[27] J. Yang, R. Ding, Z. Wang, and X. Qi, "Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding," *arXiv preprint arXiv:2304.00962*, 2023.

[28] X. Zou, Z.-Y. Dou, J. Yang, Z. Gan, L. Li, C. Li, X. Dai, H. Behl, J. Wang, L. Yuan, *et al.*, "Generalized Decoding for Pixel, Image, and Language," *arXiv preprint arXiv:2212.11270*, 2022.

[29] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *6th Annual Conference on Robot Learning*.

[30] Y. Siddiqui, L. Porzi, S. R. Buló, N. Müller, M. Nießner, A. Dai, and P. Kontschieder, "Panoptic Lifting for 3D Scene Understanding with Neural Fields," *arXiv preprint arXiv:2212.09802*, 2022.

[31] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "NeRF-Supervision: Learning Dense Object Descriptors from Neural Radiance Fields," *arXiv preprint arXiv:2203.01913*, 2022.

[32] F. Tosi, A. Tonioni, D. De Gregorio, and M. Poggi, "Nerf-supervised deep stereo," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[33] L. Schmid, M. N. Cheema, V. Reijgwart, R. Siegwart, F. Tombari, and C. Cadena, "Incremental 3d scene completion for safe and efficient exploration mapping and planning," *arXiv preprint arXiv:2208.08307*, 2022.

[34] J. L. Schonberger and J.-M. Frahm, "Structure-From-Motion Revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

[35] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMAP: Implicit Mapping and Positioning in Real-Time," in *IEEE/CVF ICCV*, 2021.

[36] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "NICE-SLAM: Neural Implicit Scalable Encoding for SLAM," in *IEEE/CVF CVPR*, 2022.

[37] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys, "Pixel-perfect structure-from-motion with featuremetric refinement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5987–5997, 2021.

[38] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. J. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser, "Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation," in *IEEE/CVF CVPR*, 2022.

[39] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint, "Learning multi-object dynamics with compositional neural radiance fields," in *Conference on Robot Learning*, pp. 1755–1768, PMLR, 2023.

[40] X. Kong, S. Liu, M. Taher, and A. Davison, "vMAP: Vectorised Object Mapping for Neural Field SLAM," *arxiv preprint arXiv:2302.01838*, 2023.

[41] R. L. Haugaard and A. G. Buch, "Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6749–6758, 2022.

[42] J. Abou-Chakra, F. Dayoub, and N. Sünderhauf, "Implicit object mapping with noisy data," *arXiv preprint arXiv:2204.10516*, 2022.

[43] V. Cohen, B. Burchfiel, T. Nguyen, N. Gopalan, S. Tellex, and G. Konidaris, "Grounding language attributes to objects using bayesian eigenobjects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1187–1194, IEEE, 2019.

[44] P. Pramanick, C. Sarkar, S. Paul, R. dev Roychoudhury, and B. Bhowmick, "Doro: Disambiguation of referred object for embodied agents," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10826–10833, 2022.

[45] J. Thomason, M. Shridhar, Y. Bisk, C. Paxton, and L. Zettlemoyer, "Language grounding with 3d objects," in *Conference on Robot Learning*, pp. 1691–1701, PMLR, 2022.

[46] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. Wang, "Vision-and-language navigation: A survey of tasks, methods, and future directions," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7606–7623, 2022.

[47] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, *et al.*, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[48] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual Language Maps for Robot Navigation," *arXiv preprint arXiv:2210.05714*, 2022.

[49] D. Shah, B. Osinski, B. Ichter, and S. Levine, "LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action," *arXiv preprint arXiv:2207.04429*, 2022.

[50] V. Blukis, R. Knepper, and Y. Artzi, "Few-shot Object Grounding and Mapping for Natural Language Robot Instruction Following," in *Conference on Robot Learning*, 2021.

[51] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. Ryoo, A. Stone, and D. Kappler, "Open-vocabulary Queryable Scene Representations for Real World Planning," *arxiv preprint arXiv:2209.09874*, 2022.

[52] S. Tan, M. Ge, D. Guo, H. Liu, and F. Sun, "Self-supervised 3D Semantic Representation Learning for Vision-and-Language Navigation," *arXiv preprint arXiv:2201.10788*, 2022.

[53] G. Georgakis, K. Schmeckpeper, K. Wanchoo, S. Dan, E. Miltsakaki, D. Roth, and K. Daniilidis, "Cross-modal map learning for vision and language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15460–15470, 2022.

[54] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "LLM-Planner: Few-Shot Grounded Planning for Embodied Agents with Large Language Models," *arXiv preprint arXiv:2212.04088*, 2022.