2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

NeRFing it: Offline Object Segmentation Through Implicit Modeling

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

[1]Autonomous Systems Lab, ETH Zürich
[2]School of ITEE, The University of Queensland, Australia

***Abstract*— Most recently proposed methods for robotic perception are based on deep learning, which require very large datasets to perform well. The accuracy of a learned model is mainly dependent on the data distribution it was trained on. Thus for deploying such models, it is crucial to use training data belonging to the robot's environment. However, collecting and labeling data is a significant bottleneck, necessitating efficient data collection and labeling pipelines. This paper presents a method to compute high-quality object segmentation maps for RGB-D video sequences using minimal human labeling effort. We leverage the density learned by a Neural Radiance Field (NeRF) to infer the geometry of the scene, which we use to compute dense segmentation maps using a single 3D bounding box provided by a user. We study the accuracy of the computed segmentation maps and present a way to generate additional synthetic training examples observing the scene from novel viewpoints using the learned radiance fields. Our results show that our method is able to compute accurate segmentation maps, outperforming baseline and state-of-the-art methods. We also show that using the synthetic training examples improves performance on a downstream object detection task.**

I. Introduction

Robotic manipulation tasks require robots to detect and compute the pose of objects. The majority of perception methods used in robotics today are based on supervised learning [1]–[4] and require large amounts of labeled training examples to fit parameters [5]. Many approaches have been devised to learn in an unsupervised or self-supervised fashion to circumvent the need for human annotated data. However, these tend to not work as well as supervised alternatives and can usually benefit from annotated data [6].

For semantic segmentation, images are usually labeled by drawing polygons on individual image examples, which can be extremely time-consuming. The cost of producing labeled datasets quickly exceeds levels that can be sustained by most robotics applications. LabelFusion [7] presented a system to rapidly annotate RGB-D data by building dense reconstructions of the environment. It can be used to compute 6D pose labels for objects with a known model. While this approach works very well when object meshes are given, in most cases, such models are not available. Additionally, intra-category variation or the fact that objects deform, can make relying on object models impractical. Having a tool that can very quickly provide ground-truth labels in the case of unknown object models, would allow us to deploy more powerful supervised learning algorithms where it previously has not been possible.

To overcome the need for ground truth mesh models of objects, we introduce a pipeline to generate semantic segmentation maps, 6D poses and 3D bounding boxes of objects for handheld RGB-D video sequences. Our method leverages neural radiance fields (NeRFs) [8] to recover the 3D structure of the scene and uses the learned NeRF model to compute object segmentation maps and bounding boxes of objects in each RGB-D frame.

NeRFs learn an implicit representation of a scene using only RGB images and known camera poses. As active depth sensors are increasingly common on robotic platforms, we propose to use depth measurements as an additional supervision signal to the NeRF model. We show that depth supervision reduces the amount of shape artifacts present in the recovered 3D shape which is crucial for generating accurate segmentation maps. Using NeRFs to represent the scene has the additional benefit that we can synthesize new viewpoints of the scene. We leverage this to generate additional synthetic training data examples for the task. We study the quality of the synthesized training examples and show that these do in fact improve the performance of a learned model on a downstream object segmentation task.

To summarize, our contributions are as follows:

- A pipeline that computes scene reconstructions and high quality semantic segmentation maps for RGB-D image sequences using a depth-supervised formulation of NeRF.
- A method to generate additional training examples by synthesizing novel viewpoints and computing the target labels.

We evaluate our proposed pipeline on a large variety of different objects. We compare the quality of the produced labels against frame-by-frame annotated semantic segmentation maps and two different baseline methods. The first one using a ground truth mesh model of objects and the second using TSDF integration that does not require object models. Additionally, we compare against a state-of-the-art object annotation pipeline, Rapid Pose Labels [9]. We evaluate the effectiveness of our data augmentation scheme on a downstream object detection task.

II. Related Work

a) 2D Annotation and Active Learning: Annotated images are needed for learning tasks such as object detection, keypoint detection or semantic segmentation. Human labeling can be expedited by tools that allow directly drawing on

the image [10] or by reducing the problem of creating semantic masks to a keypoint annotation task [11]. Active learning [12] has also been incorporated to speed up the creation of ground truth datasets by applying pixelwise or viewpoint entropy [13], [14] to achieve equivalent performance using only a fraction of the training data. Such methods could very well be used in conjunction with our approach to make use of the different viewpoints we recover in the preprocessing step to further reduce the labeling burden.

b) 3D Data Annotation: While several 3D scene datasets exist [15]–[19] the goal of this work is to enable users to easily generate their own annotated datasets specific to their task. Existing tools can triangulate 2D annotations into 3D from multiple known viewpoints [20], [21]. Other methods directly label objects in 3D using known object models [7] or through explicit scene differencing as objects are incrementally introduced [22]. RGB-D annotation can also be sped up by leveraging structure in point cloud data, either by propagating labels over successive frames [23] or via GrabCut [24] based approaches [25]. Each of these methods present restrictions either in terms of how the data is collected (manual modification of the scene, reliance on accurate depth) or they do not provide the full spectrum of 3D annotations (no segmentation masks, depth map or scene mesh).

Rapid Pose Labels [9] is the current state-of-the-art system for labeling object poses, segmentation masks, and bounding boxes for raw RGB-D video where CAD models of the objects are not available. Nevertheless, it requires multiple scans of the same object and cannot be applied to articulated or deformable objects. It assumes that depth measurements are available for all labeled 2D keypoints, hindering its application on reflective or transparent objects. Additionally, the method requires post-processing the generated object point clouds, which requires extra work per object type. Our method is able to address each of these limitations and we compare against Rapid Pose Labels in our experiments.

c) Neural Radiance Fields: NeRF [8] introduced neural radiance fields as a way to encode the appearance of a scene into a neural network and realistically synthesize novel views of the scene. Since its inception, many variations have been introduced to improve object and scene recontruction using RGB only [26] or in combination with depth supervision [27], [28]. NeRFs have also been extended to infer semantic information [29], [30] and can also use this channel to improve geometry reconstruction [31]. Similar to [30] we make use of depth maps to supervise the NeRF model and produce high quality semantic segmentation maps with little human effort. However, our method does all learning and heavy processing in an offline step and does not require large amounts of computation while using the system making it suitable to run online onboard a robot.

III. Method

Our goal is to obtain high quality semantic segmentation maps, 3D bounding boxes and 6D poses of objects for each frame in a handheld RGB-D image sequence. Specifically, in the case where the objects vary from sequence to sequence and we do not have access to CAD models of the objects.

We propose a pipeline which consists of the following steps:

1) Compute camera poses for each frame.
2) Compute a 3D point cloud of the scene.
3) Learn a depth-supervised NeRF model of the scene.
4) Annotate objects with bounding boxes using a 3D graphical user interface.
5) Compute dense semantic segmentation maps using the learned NeRF model and the 3D bounding box labels.

A. Obtaining Camera Poses

As a prerequisite step, we have to compute camera poses for each RGB-D frame to propagate bounding box labels between frames and to train the NeRF model. In our experiments, we do this using hloc [32], [33], which we run on all the frames in our video sequence. We then scale the resulting trajectory by finding the scale factor that minimizes discrepancy between measured depth and points triangulated by hloc while filtering outliers in a RANSAC loop. However, we note that any other method can be used to compute metrically scaled camera poses. Given the camera poses and RGB-D frames, we can reconstruct a point cloud of the scene.

B. Learning a NeRF Model of the Scene

We use a NeRF [8], basing our implementation on JaxNeRF [34], to infer the geometry and appearance of a scene. The idea behind the NeRF algorithm, is to trace rays from known camera positions into the scene, and learn a radiance field that maps 3D points and viewing direction to color and density values. The radiance field is modeled as a multi-layer perceptron (MLP). Image pixel values are obtained using differentiable volumetric rendering.

Neural radiance fields have many advantages. They learn a continuous representation of the scene that does not require setting any parameters, such as resolution, per-scene or task. The level of detail captured is mainly constrained by the capacity of the model learning the radiance field and the captured images. The only parameters that need to be set are the near and far bounds used for sampling. As we are dealing with RGB-D sequences, these can be set automatically to match the minimum and maximum depth readings in the clips. The MLP we use for predicting the density has 8 layers with 256 neurons and the view direction conditioned RGB network has a single layer with 256 neurons.

Following [8], we define a photometric loss:

$$L_{photo} = \left\| \hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r}) \right\|_2^2, \tag{1}$$

where $C(\mathbf{r})$ is the ground truth and $\hat{\mathbf{C}}(\mathbf{r})$ the predicted color.

In our initial tests, we observed that due to the shape-radiance ambiguity [35], reconstructing the scene using only the RGB images and known camera poses does not yield very good results for many types of scenes. Planar surfaces would get reconstructed as uneven or density would get assigned to

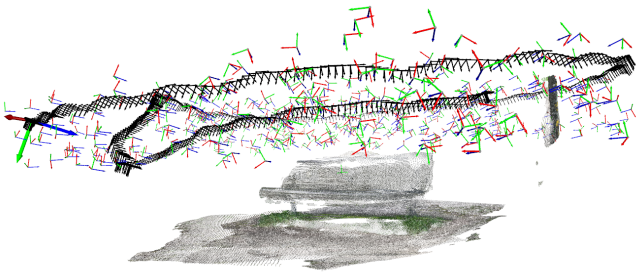

Fig. 1: Example of sampled viewpoints. The original trajectory is shown in black, the red (x-axis), green (y-axis) and blue (z-axis) axes are the sampled poses.

free space. We therefore add a depth loss to help the model disambiguate and learn from the captured depth maps where available:

$$L_{depth} = \delta(\mathbf{r}) \left\| \hat{D}(\mathbf{r}) - D(\mathbf{r}) \right\|_1, \tag{2}$$

where $\delta(\mathbf{r})$ is 0 where we don't have a depth measurement and 1 otherwise, $D(\mathbf{r})$ is the ground truth and $\hat{D}(\mathbf{r})$ the predicted depth. We optimize the model using gradient descent, by minimizing the combined loss:

$$L(\mathbf{r}) = L_{photo}(\mathbf{r}) + \lambda_d L_{depth}(\mathbf{r}), \tag{3}$$

where λ_d is a weighting parameter to balance the photometric and depth loss. In our experiments, we study the impact of this parameter.

C. Computing Object Labels

We use a graphical user interface to place 3D bounding boxes around objects of interest in the reconstructed point cloud of the scene. An example of this process is shown in the supplementary video. To compute segmentation masks, we render completed dense depth frames for each frame in our scan using the learned NeRF. We then convert each depth frame to a point cloud using the camera intrinsic parameters. Using the camera pose and object bounding boxes, we classify each point in the point cloud according to the object class and throw away points not belonging to any object. By re-projecting the object points back to the image frame, we obtain a dense segmentation mask of each object. We compute 2D bounding boxes for detection by computing the tightest bounding box containing the segmented object. Object poses can be computed by transforming the 3D bounding box pose into the camera coordinate frame.

D. Generating Synthetic Training Examples

NeRFs are able to synthesize high fidelity images from novel viewpoints, provided that the surfaces in the scene have been observed from a similar viewpoint and the free space between the novel viewpoint and the scene surface has been observed. We design an algorithm to automatically sample suitable poses and compute color, depth and segmentation mask triplets for these novel viewpoints. In our experiments, we investigate whether such synthetically generated training examples actually improve performance of a learned model on a semantic object segmentation task.

To automatically sample viewpoints from a scene, we first compute a bounding box of camera poses in an object's coordinate frame and then sample camera positions uniformly inside this bounding box. We filter out positions which are too close to a structure as measured by distance to the closest point in the point cloud. Next we compute a viewpoint orientation that points the camera at an object in the scene. We then shift the orientation by a random rotation uniformly sampled from $[-\pi/4, \pi/4]$ rad for the x and y axis and $[-\pi, \pi]$ rad for the z-axis, where the z-axis points forward, x to the right and y downwards in the image. Sampled frames are visualized in Figure 1.

IV. Experiments

We capture a number of indoor and outdoor scenes with a variety of objects to evaluate our method. Data was collected using Apple iPhone 12 Pro smartphones, which are equipped with a time-of-flight depth sensor. We train on images that have a resolution of 960×720 pixels and depth frames have a lower resolution of 256×192 pixels, which we upsample to match the color images.

A. Label Accuracy

To validate the quality of the labels produced by our proposed method, we evaluate the semantic segmentation maps of our approach with two baseline methods and compare them to manually annotated semantic segmentation maps that are obtained by drawing polygon shapes over the objects on individual images. The first baseline method involves segmenting the scene directly from the captured depth maps using the user provided bounding box. To produce the segmentation map, we first convert the depth map to a point cloud using camera intrinsic parameters. We then classify each point as being inside a bounding box or not and reproject them to the image frame to obtain the segmentation mask. The second baseline uses the mesh obtained through TSDF integration [36], [37] with a voxel size of 0.5cm and running marching cubes. We cut out the object from the mesh using the provided bounding box and render the cut out objects as a segmentation map.

Table I shows the mean intersection-over-union (mIoU) agreement across different scenes for the different methods against the manually annotated semantic segmentation masks; qualitative results can be seen in Figure 2. The NeRF-based method performs consistently better. As the depth maps captured by RGB-D sensors are noisy and have many pixels with no measurement, they produce significantly worse segmentation masks, especially for transparent and reflective objects. In some cases, the TSDF-based pipeline performs similarly to NeRF, especially for simple shapes where depth maps are of good quality, namely the fire hydrant and the park bench. Similarly, the TSDF baseline also struggles on transparent (wine glass, teapot) and reflective objects (cars, wine bottles). The NeRF approach does much better, though the masks are not quite as good in

the case of the transparent objects. Figure 3 shows some of the failure cases of the NeRF-based approach, where object segmentation masks are not correctly inferred.

We compare our method to the state-of-the-art automated 3D object annotation method, Rapid Pose Labels [9]. Similarly, Rapid Pose Labels is able to produce dense segmentation maps and poses for objects. We compare the methods on an oolong ice-tea bottle dataset containing 5 different scenes[1]. Using Rapid Pose Labels we were able to achieve a mIoU agreement of **0.801** against hand-labeled examples. Our method in turn achieves an mIoU score of **0.902** across the same scenes, a considerable improvement.

B. Depth Supervision

As depth sensors can have a large amount of noise and missing values, we study the effect of using depth maps to supervise the NeRF model. We fit the NeRF using different weights λ_d on the depth loss and qualitatively analyze how well the resulting depth maps approximate the geometry of the scene. In Figure 4 we visualize produced segmentation maps obtained with different levels of depth supervision. Table II shows the quantitative accuracy in terms of mIoU agreement across all the manually labeled examples.

C. Novel View Synthesis

As the learned NeRF is able to synthesise new viewpoints of the scene, we study the quality of the synthesized examples. To be of use in a downstream object segmentation, detection or pose estimation task, the produced color and depth would have to be qualitatively good and plausible. The segmentation masks would also have to stick to the object boundaries so as not to induce bias into the learned model.

We collect a dataset containing 19 scans of fire hydrants, which we split into 10 training scans and 9 test scans. We then annotate the scans using our method and create an additional synthetic dataset. We quantitatively verify the quality of the synthesized examples by labeling them by hand and comparing them against the produced masks.

Column 5 of Table I shows the accuracy of the computed segmentation masks for images synthesized from novel viewpoints by the NeRF model. Figure 5 shows examples of synthesized color images and segmentation masks. The generated images are generally of good quality, but some suffer from visual artifacts, mostly on unseen parts of the scene or when observing a surface from an out-of-sample viewing direction or far away surfaces that have not been properly observed or are beyond the sampling range.

To study downstream task performance using the synthetic examples, we train two semantic segmentation models: one using only the original scans and another also using the synthetic examples. We then compare the performance against manually labeled unseen examples to see if using the synthetic examples improve performance on this downstream task.

[1]Available here at the time of writing: https://github.com/rohanpsingh/RapidPoseLabels

Scene	Method mIoU Real Data			Synthetic
	Depth	TSDF	NeRF	Mask mIoU
bench1	0.4054	**0.8650**	0.8581	0.9140
bench2	0.3664	0.4712	**0.9280**	0.9130
car1	0.2139	0.6960	**0.9323**	0.9307
car2	0.2761	0.7207	**0.9471**	0.9481
cup	0.9220	0.7153	**0.9498**	0.9385
hat	0.9392	0.9209	**0.9524**	0.8956
hydrant_1	0.7468	**0.8967**	0.8861	0.8710
hydrant_2	0.8502	0.9354	**0.9551**	0.9093
hydrant_3	0.8207	0.7373	**0.9033**	0.9066
hydrant_4	0.8078	0.7835	**0.9246**	0.8827
keyboard	0.9241	0.9352	**0.9397**	0.9087
laptop	0.8160	0.8686	**0.8743**	0.8783
shoe_1	0.9312	0.8666	**0.9373**	0.9452
shoe_2	0.9299	0.9184	**0.9755**	0.9568
shoe_3	0.9614	0.9491	**0.9719**	0.9047
teapot	0.5799	0.5257	**0.8458**	0.8664
wine_glass	0.0381	0.0000	**0.7565**	0.7272
wine_bottle_red_1	0.6877	0.7960	**0.8739**	0.8456
wine_bottle_red_2	0.6916	0.7046	**0.8879**	0.8862
wine_bottle_white	0.6670	0.8456	**0.9006**	0.8894

TABLE I: Columns 2-4: mean intersection-over-union (mIOU) accuracy of segmentation masks for real examples, computed using the different methods. Column 5: mIOU accuracy of masks for synthetic examples.

scene $\backslash \boldsymbol{\lambda_d}$	0.0	0.001	0.01	0.05	0.1	0.25
cup	0.691	0.755	0.920	0.9498	0.946	0.948

TABLE II: Segmentation mIoU on the cup scene for different depth supervision weights λ_d.

Figure 6 compares the accuracy of two different object detection models, Mask-RCNN [38] and Yolo-v3 [39], that were trained on the fire hydrant dataset with different ratios of synthetic to real data. For the Mask-RCNN model which does both segmentation and detection, performance on both detection and segmentation is shown. Yolo only does object detection, so we only report detection performance. Detection accuracy for both models increases up until 60% synthetic data, before decreasing. Object segmentation performance peaks a little bit earlier, likely due to the less realistic sharpness and texture of the synthetic examples, which are much more important for segmentation than bounding box detection.

V. Discussion and Conclusions

We introduced a pipeline that uses NeRFs to create dense pixelwise labels for semantic segmentation of objects. We showed that the geometry recovered by a NeRF can be used to generate high quality segmentation masks which outperformed baseline methods. We showed that the learned NeRF can be used to generate additional data in the form of unseen viewpoints of the scanned scenes which can be used to train an object detector, further reducing the data collection burden. A promising direction for future work would be to investigate methods that could further diversify the generated examples, for example by relighting the scenes.

Fig. 2: Segmentation masks obtained using our method on different scenes. Segmentations are in yellow. Scenes are in the same order from top-left to bottom-right as in Table I.

Fig. 3: In some cases, geometry might not get recovered properly, as shown with the hole in the laptop. The hole in the handle of the teapot is not correctly inferred as free space. In the other two cases, some density is assigned outside of the object.

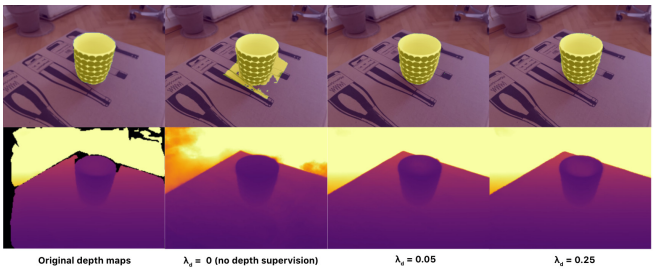

Fig. 4: The produced depth and segmentation masks on the cups scene with different levels of depth supervision given during NeRF training. The leftmost image shows the original depth maps. We see that the NeRF model trained without depth supervision produces artifacts in the geometry which end up affecting the segmentation masks.

Further research might focus on extending the method to scenes with moving objects or automating the placement of the bounding boxes, for example through active learning.

Our proposed method relies on accurate camera poses. Should the camera pose recovery step fail or produce bad results, the learned NeRF model will be severely impacted which leads to low-quality geometry reconstruction and thus poor segmentation masks. Estimating camera poses is an entire field of study in itself. However, this could be solved by mounting the camera on a manipulator and calibrating the system to obtain camera poses using proprioceptive sensor data, as done in [21].

As shown in the experiments, our method performs significantly better than the baseline methods on most objects, including transparent and reflective ones, but there are limits. As previously discussed, inferring the geometry of clear and fully transparent objects is still a challenge. Visual features in the images could be used to help the model cleanly segment and infer the density.

Our test scenes currently contain a single foreground object, often with other objects or clutter in the background. Since we use bounding boxes as a source of supervision, in more cluttered scenes, other objects might enter an object's bounding box, producing noisy labels. Further work could go towards filtering label noise in such scenarios. Another source of error in the segmentation masks comes from the scene geometry not being perfectly inferred and label edges not matching the object's boundary. This could be addressed by further refining the scene geometry and labels by allowing a user to refine the labels online, providing fixes to the generated 2D segmentation maps and using the additional supervision to improve on both the representation and the generated semantics.

References

[1] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

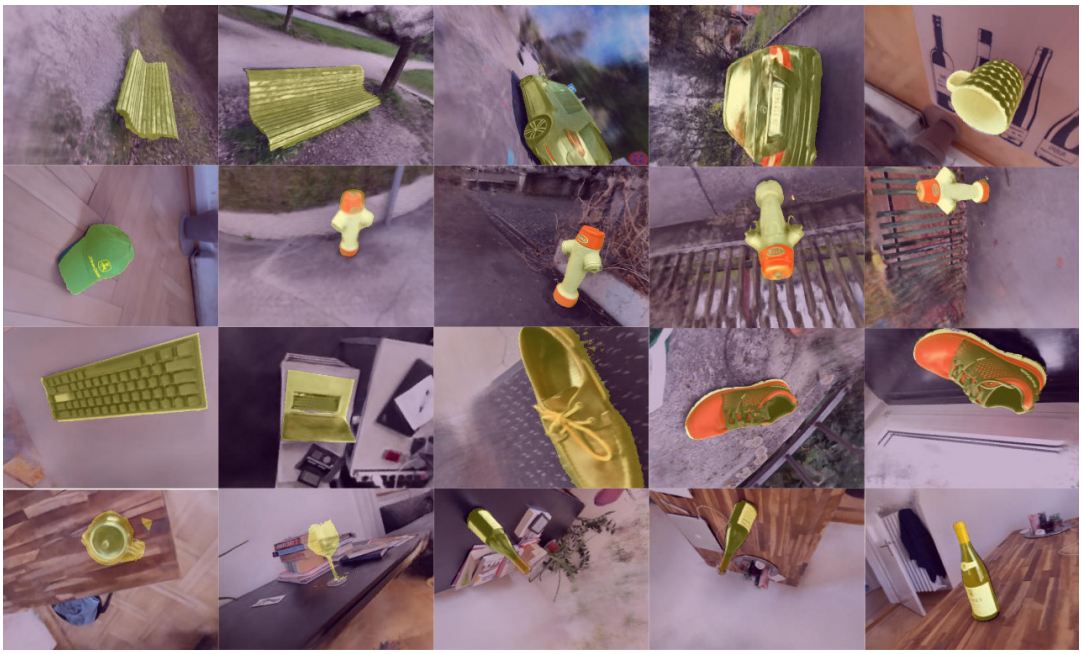

Fig. 5: Synthetic training examples and their associated segmentation masks, generated using the proposed approach.

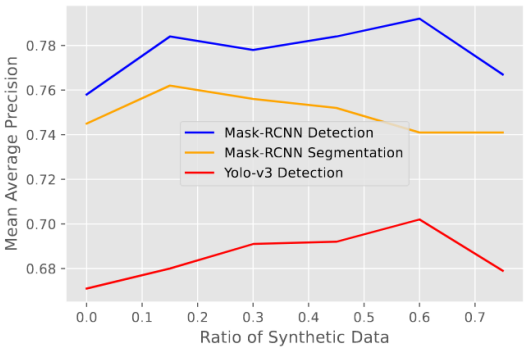

Fig. 6: Average precision on the held out test set for models trained with different ratios of synthetic to real data.

[2] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *The International Symposium of Robotics Research*, 2019.

[3] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Single-stage keypoint-based category-level object pose estimation from an rgb image," in *Proc. of the IEEE Int. Conference on Robotics & Automation (ICRA)*, 2022.

[5] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford *et al.*, "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, 2018.

[6] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[7] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake, "LabelFusion: A pipeline for generating ground truth labels for real RGBD data of cluttered scenes," in *Proc. of the IEEE Int. Conference on Robotics & Automation (ICRA)*, 2018.

[8] 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 *Proc. of the European Conference on Computer Vision (ECCV)*, 2020.

[9] R. P. Singh, M. Benallegue, Y. Yoshiyasu, and F. Kanehiro, "Rapid pose label generation through sparse representation of unknown objects," in *Proc. of the IEEE Int. Conference on Robotics & Automation (ICRA)*, 2021.

[10] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International Journal of Computer Vision*, 2008.

[11] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] B. Settles, "From theories to queries: Active learning in practice," in *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, 2011.

[13] G. Shin, W. Xie, and S. Albanie, "All you need are a few pixels: semantic segmentation with pixelpick," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] Y. Siddiqui, J. Valentin, and M. Nießner, "Viewal: Active learning with viewpoint entropy for semantic segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[17] G. Baruch, Z. Chen, A. Dehghan, Y. Feigin, P. Fu, T. Gebauer, D. Kurz, T. Dimry, B. Joffe, A. Schwartz *et al.*, "ARKitScenes: A diverse real-world dataset for 3D indoor scene understanding using mobile rgb-d data," in *Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[18] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3D objects from real depth images using mask r-cnn trained on synthetic point clouds," in *Proc. of the IEEE Int. Conference on Robotics & Automation (ICRA)*, 2019.

[19] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3D object detection and pose estimation," in *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018.

[20] X. Liu, R. Jonschkowski, A. Angelova, and K. Konolige, "Keypose: Multi-view 3D labeling and keypoint estimation for transparent objects," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] K. Blomqvist, J. J. Chung, L. Ott, and R. Siegwart, "Semi-automatic 3D object keypoint annotation and detection for the masses," in *International Conference on Pattern Recognition*, 2022.

[22] M. Suchi, T. Patten, D. Fischinger, and M. Vincze, "EasyLabel: a semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets," in *Proc. of the IEEE Int. Conference on Robotics & Automation (ICRA)*, 2019.

[23] H. A. Arief, M. Arief, G. Zhang, Z. Liu, M. Bhat, U. G. Indahl, H. Tveite, and D. Zhao, "SAnE: Smart annotation and evaluation tools for point cloud data," *IEEE Access*, 2020.

[24] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut" interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics (TOG)*, 2004.

[25] D. Stumpf, S. Krauß, G. Reis, O. Wasenmüller, and D. Stricker, "Salt: A semi-automatic labeling tool for rgb-d video sequences," *arXiv preprint arXiv:2102.10820*, 2021.

[26] 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.

[27] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[29] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, "In-place scene labelling and understanding with implicit scene representation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] S. Zhi, E. Sucar, A. Mouton, I. Haughton, T. Laidlow, and A. J. Davison, "ilabel: Interactive neural scene labelling," *arXiv preprint arXiv:2111.14637*, 2021.

[31] X. Fu, S. Zhang, T. Chen, Y. Lu, L. Zhu, X. Zhou, A. Geiger, and Y. Liao, "Panoptic NeRF: 3D-to-2D label transfer for panoptic urban scene segmentation," *arXiv preprint arXiv:2203.15224*, 2022.

[32] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[34] B. Deng, J. T. Barron, and P. P. Srinivasan, "JaxNeRF: an efficient JAX implementation of NeRF," 2020. [Online]. Available: https://github.com/google-research/google-research/tree/master/jaxnerf

[35] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[36] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proc. of the Conference on Computer Graphics and Interactive Techniques*, 1996.

[37] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality*, 2011.

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.