Supplementary Material for
NAVI: Category-Agnostic Image Collections with High-Quality 3D Shape and Pose Annotations

Varun Jampani* **Kevis-Kokitsi Maninis*** **Andreas Engelhardt** **Arjun Karpur**
Karen Truong **Kyle Sargent** **Stefan Popov** **Andre Araujo**
Ricardo Martin-Brualla **Kaushal Patel** **Daniel Vlasic** **Vittorio Ferrari**
Ameesh Makadia **Ce Liu†** **Yuanzhen Li** **Howard Zhou**

Google

Contents

*Equal contribution
†C. Liu's current affiliation is Microsoft

Preprint. Under review.

Alignment initialization Alignment result

Figure A1: **3D-2D alignment tool**. The user is able to rotate and translate the 3D object directly on the screen. We provide ways to improve the annotation experience such as a restricted 'easy' mode when the object appears upright, saving a backup pose that can be recovered, enabling/disabling texture, and others. For a full demo, please refer to `alignment_tool.mov`.

1 Additional dataset details

1.1 3D↔2D alignment tool

Our interactive alignment tool was developed using the three.js library [31] that allows the user to interact with a 3D object directly on the browser. The objective is to directly produce alignments on images by rotating and translating the object.

For rotating the object, we used the intuitive mouse movements of Orbit Controls [15] that allow the user to rotate an object in 3D using 2D drag-and-drop movements. Orbit Controls constrains the 'up'-axis of the 3D shape ('easy' constrained mode), which makes the task much easier when the objects are in 'upright' position on the images. For adjustments, and for objects that do not appear in 'upright' position, the user has the option to remove the 'up-axis' constraint, and allow for all possible 3D poses ('difficult' unconstrained mode). Our tool has the option to switch between the two modes, which enables the user to first bring the object to a close-enough pose using 'easy' mode, and then switch to 'difficult' mode for the final adjustments. For translating the object, we used the panning functionality of [15].

Taking into account feedback from the annotators, we developed the option to save a backup pose, and revert back to it in case they need to restart the process (eg. when the backup pose is better than the current pose). We further developed simple keyboard shortcuts that improve the annotation experience, such as disabling/enabling texture on the 3D shape, and changing its opacity. Figure A1 illustrates our annotation interface. The camera parameters (focal length) are initialized from the ExIF metadata of the images, and can also be adjusted from within the tool. For a full demo please refer to `alignment_tool.mov` included in this supplementary.

We observed that our interactive tool gives the user full control over the alignments, and lead to higher-quality poses than alternative implicit alignment tools that optimize the pose from 3D↔2D point correspondences [29]. This allowed us to obtain annotations of higher quality than existing datasets that use general crowd-sourcing.

1.2 Dataset statistics

Table 1 presents the general statistics of the NAVI dataset. It contains 10515 alignments in total over 36 complicated object shapes, divided into in-the-wild images (2298) and multiview scenes (8217). Each object is aligned on 65 in-the-wild images on average. There are 267 unique multiview scenes, some of which were also captured by different cameras (324 multiview captures in total).

1.3 Data license, access details, and intended usage

The dataset is released under the CC-BY license. The accompanying code that shows the dataset use is released under the Apache License 2.0. The authors of the dataset bear all responsibility in case of violation of rights.

All contents of this submission (code, paper, data) can be accessed from our project page: https://navidataset.github.io. The data is hosted on Google Cloud by Google Research. The authors will maintain and update the dataset.

# Alignments (total)	10515
# Alignments (wild)	2298
# Alignments (multiview)	8217
# Objects	36
# Scenes	324
# Scenes (unique)	267

Table 1: **General statistics of NAVI.**

Extensive documentation of the dataset and how to use it for the various tasks can be found in the accompanying Github repo: https://github.com/google/navi. Users are invited to use the included Jupyter notebook NAVI Dataset Tutorial.ipynb for a quick start.

The released dataset consists of multiple folders with images (`jpg`), scans (`obj`, `mtl`, `glb`) and annotations (`json`) that connect them. Users can download the dataset at https://storage.googleapis.com/gresearch/navi-dataset/navi_v1.tar.gz (29GB).

The intended usage of the dataset is to enable benchmarking and systematic development of the 3D vision tasks presented in the main paper as well as this supplemental: 3D from multiview image collections, 3D from in-the-wild image collections, correspondence estimation, and 3D from a single image.

2 3D from multiview image collections

2.1 COLMAP initialization details

For the multiview collections included in the evaluation we randomly selected 5 scenes with more than 10 images for each of the objects. On those scenes COLMAP [27] successfully registered 73% ($\pm$ 32%) of the views, on average. We only evaluate on the views of the validation set that were successfully registered by COLMAP, which gives the experiments initialized from COLMAP cameras an advantage.

2.2 Visual results

Figure A2 illustrates qualitative results for novel view synthesis from multiview reconstruction using InstantNGP [21]. We compare the camera poses obtained by COLMAP with the ground-truth camera poses of NAVI. We observe some artifacts like the oversmoothed and noisy contours, especially on the COLMAP variant. These artifacts can be attributed to slightly offset camera poses, as well as the relatively small number of images in certain scenes for NeRF-like methods. We add a small amount of distortion loss as proposed by [1] to reduce the risk of floater artifacts. Additional regularization might be beneficial for some scenes to further improve results.

3 3D from in-the-wild image collections

3.1 Comparison with existing in-the-wild image collections

Table 2 compares existing published image collections used for 3D reconstruction from in-the-wild data to NAVI. Most existing collections include only a few number of scenes and, if at all, sparse 3D annotations. NAVI is the only known dataset of that kind with real-world image collections of objects in the wild (with varying environments and cameras) that also includes verified 3D$\leftrightarrow$2D alignments using 3D scanned meshes.

(a) GT Novel View Image (b) COLMAP cameras (c) NAVI cameras

Figure A2: **Novel view synthesis from multiview reconstructions**. We compare test examples from runs on selected scenes initialized with COLMAP reconstructed camera poses to NAVI GT pose initialization. For each configuration an InstantNGP [21] instance is optimized.

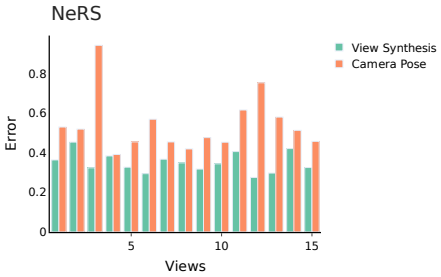
Dataset	# Objects	# Scenes	# Images	Camera Poses	3D Annotations	3D↔2D Alignment
LASSIE [36]	6	6	180	-	Keypoints	✗
E-LASSIE [37]	6	6	270	-	Keypoints	✗
NeRD [3]	8	8	396	Synthetic-GT	-	✗
NeRF-W [18]	-	6	7658	-	-	✗
SAMURAI [5]	8	8	560	-	-	✗
NeROIC [12]	3	3	132	COLMAP	-	✗
NAVI (Ours)	36	36	2298	Near-GT	Scanned mesh	✓

Table 2: **In-the-wild image collections.** We compare statistics for available image collections featuring in-the-wild data, meaning images of the same object taken in different environments and lighting settings.

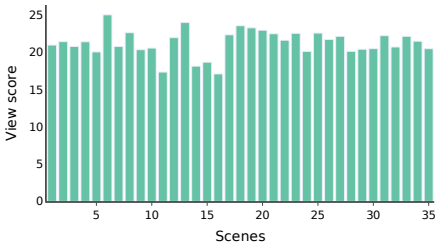
3.2 Details of evaluated techniques

In the following, we briefly summarize the different techniques we analyzed for 3D from in-the-wild image collections on NAVI. For more details, please refer to the respective paper.

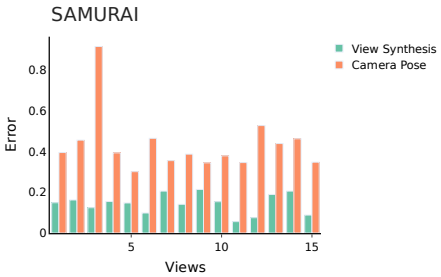
- **NeROIC [12]** proposes a multi-stage approach to reconstruct geometry and material properties from online image collections of objects. Camera poses are initialized with a COLMAP-based pipeline and fine-tuned during the first reconstruction stage which is followed by a normal extraction stage to estimate high-quality surface normals. Finally, material properties and illumination are estimated to enable relighting in addition to novel view synthesis.
- **NeRS [38]** introduces Neural Reflectance Surfaces that constrain reconstructions using a surface-based representation. Starting from manually annotated rough initial poses and a template mesh the objects are decomposed into a surface mesh, illumination and surface reflectivity as albedo and shininess. We define the dimensions of an initial cuboid that approximates the object's bounding box for each scene as suggested by [38].
- **SAMURAI [5]** enables reconstruction of a NeRF representation and decomposes appearance into illumination and a physically based BRDF based on a differentiable renderer for pre-integrated lighting [4]. Camera poses are initialized as quadrants from manual annotation and then refined using a multiplex and a coarse-to-fine scheme. We obtain the initial directions from the GT poses and use them to initialize both NeRS and SAMURAI.
- **GNeRF [19]** employs an adversarial approach and a pre-trained inversion network for camera pose and shape optimization from completely unknown cameras. GNeRF is the only method presented here that does not account for the different lighting settings which is also reflected in worse performance on more challenging scenes.



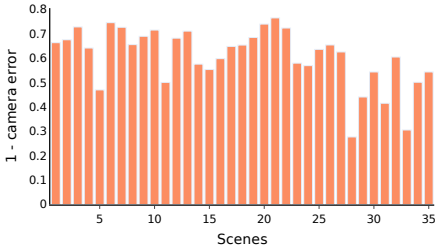
(a) Camera and view rendering errors over the test views for NeRS [38]



(b) Mean view rendering score over all in-the-wild-scenes averaged from SAMURAI and NeRS results.



(c) Camera and view rendering errors over the test views for SAMURAI [5]



(d) Mean inverse camera error over all in-the-wild scenes averaged from SAMURAI and NeRS results.

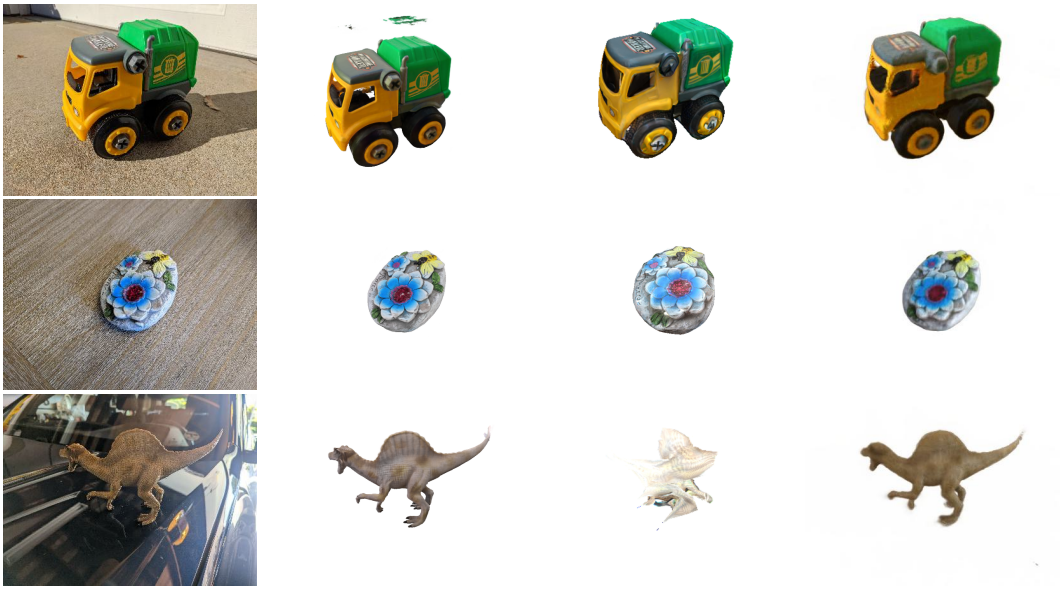
Figure A3: **Error distribution over scenes and views**. Left we show the distribution of normalized view errors (inverse PSNR and SSIM) and camera pose errors (translation and rotation) over the test views of the "Keywest Showpiece" in-the-wild scene for two reconstruction methods, NeRS and SAMURAI. Lower values are better. On the right the normalized view score (SSIM and PSNR) and the normalized inverse camera error (translation and rotation) are given for all in-the-wild scenes in NAVI. Higher values are better. We report the average of SAMURAI and NeRS results.

3.3 Additional results

Error distribution: Figure A3b and A3d visualize the combined mean scores of PSNR and SSIM for the view synthesis task from SAMURAI and NeRS for all in-the-wild scenes. We observe that the difficulty of the NAVI scenes varies, as indicated by the fluctuating scores. By analyzing the results, we notice that the most challenging scenes across methods are the ones featuring symmetric objects, such as water guns and a hand drill. Our intuition is that symmetric objects of complicated shapes pose an additional challenge for these methods.

Figure A3a and A3c show the normalized view error and camera pose error over the views in one example scene. It is a common artifact that some cameras still show a high error after the optimization, probably being trapped in a local minimum. Interestingly, both NeRS and SAMURAI have a similar camera pose error distribution despite the differences regarding their 3d representation and camera optimization strategy. It's also important to note that large errors of individual cameras do not necessarily result in a failed shape reconstruction as long as there are enough correctly aligned views.

View synthesis: Figure A4 presents additional view synthesis results from NeROIC [12], NeRS [38] and SAMURAI [5] for selected test views. NeROIC is tailored towards high quality view rendering results which can be achieved with good initial poses. When initialized further away from the GT poses the lack of additional camera regularization leads to failing reconstructions. NeRS is able to robustly reconstruct all objects in most settings but shows limited quality for more complex shapes. Poses are generally less precise and scaled differently compared to the ground truth. SAMURAI is tuned towards large camera updates in the beginning of the training and therefore is able to reconstruct even from large initial pose offsets. Rendering quality is limited by the used illumination model.



(a) GT Novel View Image (b) NeROIC (c) NeRS (d) SAMURAI

Figure A4: **Novel view synthesis with in-the-wild 3D reconstruction**. Sample novel view synthesis results of different techniques for 3D reconstruction from in-the-wild image collections.

Method	Initialization	Mean Distance↓	IoU↑
NeRS [38]	Quadrants	5.86	45.1%
SAMURAI [5]	Quadrants	3.65	63.7%

Table 3: **Shape evaluation from in-the-wild images.** The scanned 3D shapes provided by NAVI enable the evaluation of the reconstructed shape independently of the camera poses and rendering quality. We compare the point cloud extracted on the surface of the predicted mesh to points sampled on the ground truth mesh by using the Chamfer distance. We also measure Intersection-over-union (IoU) on fixed-resolution occupancy grid generated from the predicted and GT meshes.

Results often show some smoothness, also a result of small pose errors that can not be compensated by the neural representation due to the explicit rendering step.

Direct 3D Shape evaluation

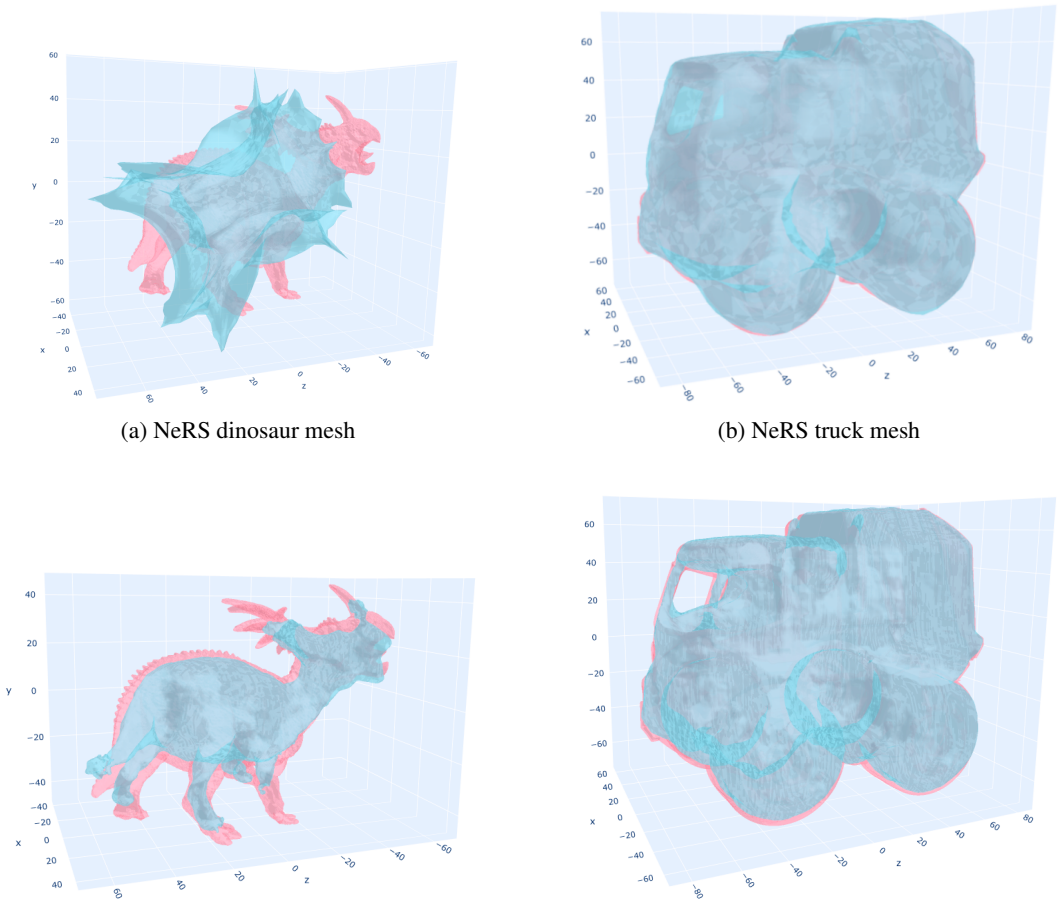
NAVI enables directly evaluating 3D shape reconstruction by using the provided 3D scans that are aligned on the images.

For this experiment we use NeRS that generates a mesh as part of its optimization and SAMURAI that provides a mesh extraction pipeline. Generally, it is possible to generate point clouds from a NeRF representation which could also be compared to the GT mesh.

The reconstruction output of these methods is arbitrarily transformed by a rigid transformation due to the optimization setup. To align the predicted mesh with the GT mesh we first adjust its scale. We then perform fast global registration [39] on downsampled point clouds followed by refinement via point-plane ICP [25]. Figure A5 shows examples of the mesh alignment process.

For evaluation we use the mean Chamfer distance of the two set of vertices, and the 3D intersection over union (IoU) of the voxelized meshes. Results are presented in Table 3.

We observe that the relative reconstruction quality in terms of the shape metrics corresponds to the novel view synthesis results. Looking at the aligned meshes we can derive further insights. NeRS, which is initialized with a cuboid, fails to reconstruct objects with shapes that are very different from its initialization (see dinosaur in Figure A5a for NeRS vs. Figure A5c for SAMURAI). On the other



(a) NeRS dinosaur mesh

(b) NeRS truck mesh

(c) SAMURAI dinosaur mesh

(d) SAMURAI truck mesh

Figure A5: **Comparison of predicted and GT mesh after alignment**. The reconstructed mesh (blue) is aligned to the GT mesh (pink) and overlaid with a 50% blend. We show two methods: NeRS (top) and SAMURAI (bottom).

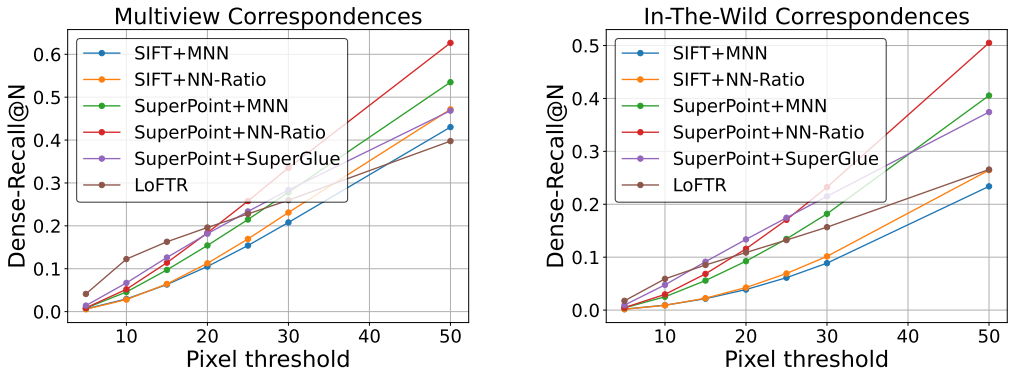
hand, for objects with shapes closer to a cuboid NeRS tends to predict the bulk of the object more accurately (see less pink regions in Figure A5b), while SAMURAI is able to add finer details that are missing from the NeRS reconstruction (see Figure A5d).

4 Correspondence estimation

4.1 Details of experimented techniques

In the following, we briefly summarize the different correspondence techniques we benchmarked with NAVI. See the respective papers for more details.

- **SIFT + MNN/NN-Ratio [16]**. We use SIFT local features with heuristics-based matchers to represent the traditional baseline used in many Structure-from-Motion pipelines. Specifically, we use two popular variants of nearest neighbor search: mutual nearest neighbor and Lowe's ratio test.
- **SuperPoint + MNN/NN-Ratio [8]**. We replace traditional SIFT feature extraction with a learned detect/describe method, SuperPoint. We use traditional matchers to predict correspondences and rely on the improved descriptiveness of SuperPoint's features.
- **SuperPoint + SuperGlue [26]**. We replace traditional matchers with the popular SuperGlue sparse learnable feature matcher, which relies on a graph neural network and attention modules to predict correspondences from the input keypoint set. SuperGlue is trained on multiview image pairs from outdoor scenes, and we perform no additional object-centric finetuning.



(a) (b)

Figure A6: **Dense-Recall@N,** where the pixel radius N ranges from 5-50 pixels. Plots are provided for both the *multiview* and *in-the-wild* sets.

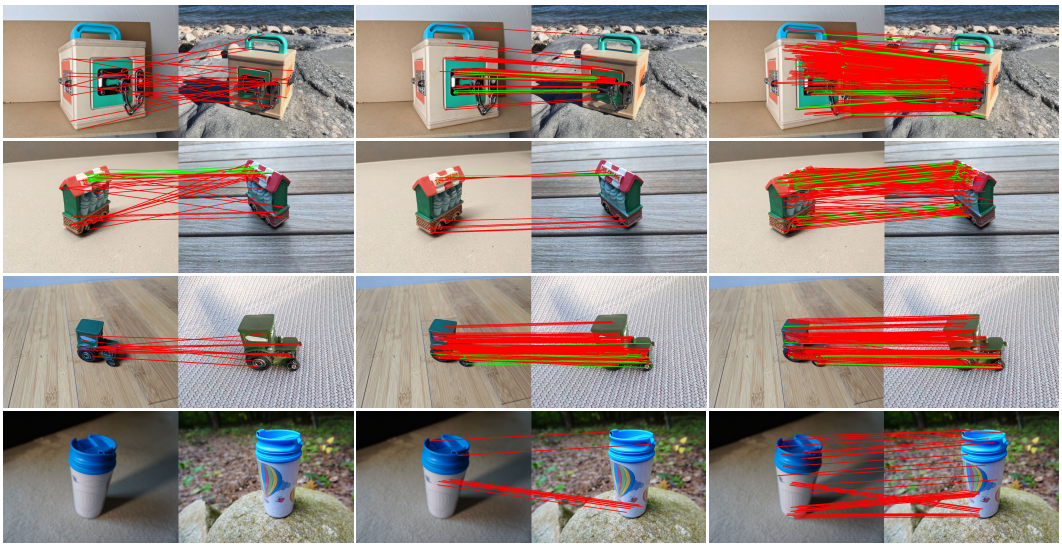


Figure A7: **Sample correspondence results** of different techniques where the correct (within 3 pixels) and incorrect matches are shown in green and red respectively.

- **LoFTR [28]**. Dense learnable matchers are often used to overcome repeatability issues in sparse local feature detection and matching. LoFTR relies on a coarse-to-fine transformer to propose a wider set of correspondences across entire images. We use the Kornia [24] implementation of LoFTR, which is also pretrained on outdoor scene pairs.

4.2 Additional results

Dense recall at varying thresholds. Fig. A6 presents Dense-Recall@N results for the same set of correspondence estimation techniques, but with varying pixel radius'. We sweep radius values between 5 and 50 pixels and show that methods vary in dense recall performance at different threshold values. LoFTR and SuperPoint+SuperGlue outperform most methods in low-radius scenarios, but traditional matchers (SuperPoint+MNN and SuperPoint+NN-Ratio) see stronger performance for higher pixel radius'.

Additional visual results. Fig. A7 presents additional qualitative results for three correspondence estimation techniques. We believe these visualizations show that popular correspondence estimation techniques still have significant headroom for the task of fine-grained, object-centric correspondence estimation.

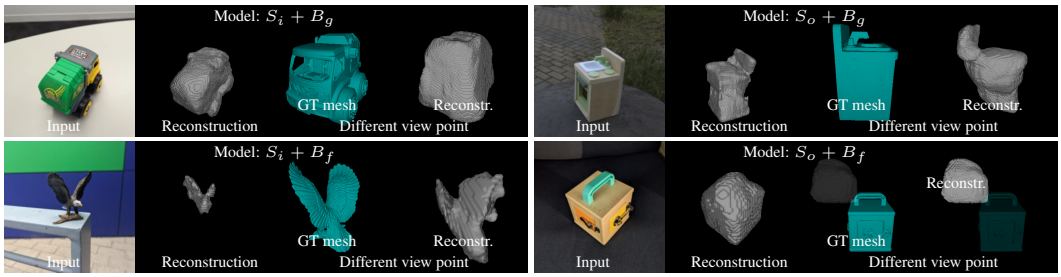


Figure A8: **Sample single image 3D reconstructions using CoreNet [23]**. In all cases, the reconstructed geometry aligns well with the input image. When splitting along objects (S_o), reconstructions contain errors in unobserved parts. In addition, CoReNet cannot resolve the depth/scale ambiguity for $S_o + B_f$ and it reconstructs objects at a wrong depth. Both are evident when viewing reconstruction from a different view point.

5 3D from a Single Image

Problem setting. Given a single RGB image of an object, the aim is to reconstruct the 3D shape and optionally the 3D pose of the object depicted in it. Shapes are commonly represented as occupancy grids [23, 2, 7, 11, 32, 34, 35], point clouds [9, 17], or implicitly [22, 20, 6]. Poses are predicted relative to the camera, commonly as an explicit transformation [10, 13], or as part of the scene volume [23]. Single image 3D reconstruction is a highly ambiguous and under-constrained vision problem as the techniques have to reason about the complete 3D shape of the object from a single 2D projection of that object in a single environment.

The distinctiveness of NAVI. Commonly used real-world datasets for single image 3D reconstruction such as Pix3D [29] and Pascal3D+ [33] are category-specific with objects of some common categories such as chairs, cars etc. As a result, simple recognition based 3D model retrieval techniques can already perform well on such class-specific datasets [30]. In contrast, NAVI objects are category-agnostic and provide a unique opportunity to evaluate the 3D geometric understanding capabilities of the techniques. Another key issue with the most existing real-world datasets is that the 3D shapes are only approximate (either nearest CAD models or reconstructed using SfM), whereas NAVI provides near-perfect 3D shape GT and alignments allowing for more accurate evaluations of 3D reconstructions.

NAVI dataset and metrics. In the experiments, we use all NAVI images, from both multiview and in-the-wild collection types. We split the images into train and test using three strategies: randomly (S_i), randomly along the object they depict (S_o), and along objects and environments (S_b). There are no images in common between the test and train splits in all three strategies. In addition, the set of object instances depicted on the images in S_o and S_b is disjoint between the train and test splits. Finally, the backgrounds against which objects are photographed are dissimilar between the train and test splits of S_b. We use the official NAVI splits for S_i and S_o and we rely on capture location to split the dataset for S_b. S_i and S_o follow the practice of [23, 10] for measuring performance in datasets with real images. Splitting along images in S_i allows the model to learn about the geometry of all objects in the dataset and to apply this knowledge to images it hasn't seen in the test split. S_o presents a harder scenario, as the model has to reconstruct unseen geometry on unseen images. S_b poses an additional challenge, as the model can no longer rely on a familiar background on the test set.

Experiment setup. For our preliminary experiments, we use CoReNet [23] as a representative single-object reconstruction method. CoReNet predicts volumetric binary occupancy on a regular grid inside a given 3D box in front of the camera. The geometry and the pose of the object can be extracted from the grid using Marching Cubes [14]. We establish baselines by training and evaluating CoReNet on the NAVI dataset. We evaluate CoReNet in two scenarios: 1. We provide CoReNet with access to the object's GT pose at test time (B_g), using the mechanism described in the paper [23] for the Pix3D dataset and; 2. We ask CoReNet to reconstruct occupancy inside a fixed $3m \times 3m \times 3m$ bounding box, placed $1m$ in front of the camera (B_f). Scenario B_g is essentially equivalent to shape prediction. Scenario B_f combines shape and pose prediction. It is much harder than B_g, since the model has to resolve the depth/scale ambiguity from a single image only. It is more feasible in NAVI, as NAVI has a single prominent object in each image and the pose is given in metric space.

	images (S_i)	objects (S_o)	backgrounds (S_b)
GT pose provided (B_g)	66.6%	46.3%	46.2%
Fixed grid (B_f)	49.7%	13.1%	–

Table 4: **IoU performance of CoReNet on the NAVI dataset, under different settings**. Columns indicate the way data is split into train and test, rows – whether the model has access to the ground-truth pose at test time.

5.1 Analysis

We train CoReNet models on 5 combinations of split strategy and pose prediction. In all cases, we start from a model pre-trained on synthetic data (h_7 from the CoReNet paper), and we train for 15 epochs. To evaluate, we measure intersection-over-union (IoU) between the predicted and ground truth occupancy grids. Table 4 summarizes the results, Figure A8 shows sample reconstructions.

Analysis across different splits. Models trained on S_i outperform those trained on S_o by a large margin (+20.3% for B_g, +36.6% for B_f). As expected, splitting randomly along images allows the model to learn about object geometry and to apply this to the similar objects in the test set. This is also confirmed visually (Figure A8). Reconstructed objects re-project correctly over the input images in all scenarios, but they contain large errors in unobserved regions for S_o. The difference between models trained on S_o and S_b is negligible, indicating that learning about geometry is more important than background.

Analysis with and without pose prediction. Comparing models with access to the ground-truth pose (B_g) to those without (B_f), shows that performance falls modestly for models trained on S_i (from 66.6% to 49.7%) and significantly for models trained on S_o (from 46.3% to 13.1%). Learning about model geometry and most importantly about object size becomes essential for B_f, as the model has no other means to resolve the depth/scale ambiguity. Visually, reconstructed objects re-project correctly over the input images, but looking from below reveals that they are smaller/larger than the ground truth and they are reconstructed at the wrong depth (Figure A8).

References

[1] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[2] W. Bian, Z. Wang, K. Li, and V. A. Prisacariu. Ray-onet: Efficient 3d reconstruction rrom A single RGB image. In *British Machine Vision Conference (BMVC)*, page 377, 2021.

[3] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. P. Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[4] M. Boss, V. Jampani, R. Braun, C. Liu, J. T. Barron, and H. P. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[5] M. Boss, A. Engelhardt, A. Kar, Y. Li, D. Sun, J. T. Barron, H. P. Lensch, and V. Jampani. SAMURAI: Shape and material from unconstrained real-world arbitrary image collections. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[6] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016.

[8] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 224–236, 2018.

[9] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] J. J. Georgia Gkioxari, Jitendra Malik. Mesh r-cnn. *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[11] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision (ECCV)*, 2016.

[12] Z. Kuang, K. Olszewski, M. Chai, Z. Huang, P. Achlioptas, and S. Tulyakov. Neroic: Neural rendering of objects from online image collections. *ACM Transactions on Graphics*, 41(4), jul 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530177. URL `https://doi.org/10.1145/3528223.3530177`.

[13] W. Kuo, A. Angelova, T.-Y. Lin, and A. Dai. Mask2CAD: 3D shape prediction by learning to segment and retrieve. In *European Conference on Computer Vision (ECCV)*, 2020.

[14] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares. Efficient implementation of marching cubes' cases with topological guarantees. *J. Graphics, GPU, & Game Tools*, 8(2):1–15, 2003.

[15] M. Livingston, A. Gregory, and W. Culbertson. Camera control in three dimensions with a two-dimensional input device. *Journal of Graphics Tools*, 5, 01 2000.

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

[17] P. Mandikal, N. K. L., M. Agarwal, and V. B. Radhakrishnan. 3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. In *British Machine Vision Conference (BMVC)*, 2018.

[18] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. NeRF in the Wild: Neural radiance fields for Unconstrained photo collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[19] Q. Meng, A. Chen, H. Luo, M. Wu, H. Su, L. Xu, X. He, and J. Yu. GNeRF: GAN-based neural radiance field without posed camera. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[20] L. M. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (SIGGRAPH)*, 41(4), July 2022. doi: 10.1145/3528223.3530127. URL `https://doi.org/10.1145/3528223.3530127`.

[22] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] S. Popov, P. Bauszat, and V. Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *ECCV*, pages 366–383. Springer International Publishing, 2020.

[24] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski. Kornia: An open source differentiable computer vision library for pytorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3674–3683, 2020.

[25] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *International Conference on 3-D Digital Imaging and Modeling*, 2001. doi: 10.1109/IM.2001.924423.

[26] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4938–4947, 2020.

[27] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou. Loftr: Detector-free local feature matching with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8922–8931, 2021.

[29] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3D: Dataset and methods for single-image 3D shape modeling. In *CVPR*, 2018.

[30] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox. What do single-view 3d reconstruction networks learn? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3405–3414, 2019.

[31] D. A. Williams. three.js. URL `https://threejs.org/`.

[32] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[33] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014.

[34] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[35] H. Xie, H. Yao, S. Zhang, S. Zhou, and W. Sun. Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *International Journal on Computer Vision (IJCV)*, 2020.

[36] C.-H. Yao, W.-C. Hung, Y. Li, M. Rubinstein, M.-H. Yang, and V. Jampani. Lassie: Learning articulated shape from sparse image ensemble via 3d part discovery. In *NeurIPS*, 2022.

[37] C.-H. Yao, A. Raj, W.-C. Hung, Y. Li, M. Rubinstein, M.-H. Yang, and V. Jampani. Artic3d: Learning robust articulated 3d shapes from noisy web image collections. 2023.

[38] J. Y. Zhang, G. Yang, S. Tulsiani, and D. Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[39] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *European Conference on Computer Vision (ECCV)*, 2016.