Supplementary Material for ObjectMatch: Robust Registration using Canonical Object Correspondences

Can Gümeli Angela Dai Matthias Nießner

Technical University of Munich

In this supplementary material, we first show additional registration results in Section A and additional ablation studies in Section B. We then describe additional method and baseline details in Section C.

A. Additional Results

Additional Pair Results. In Figure 1, we show additional low-overlap view registration samples from ScanNet [4] validation and test images.

SLAM Reconstructions. In Figure 2, we show various scene reconstructions from TUM RGB-D [22] and ScanNet [4] using our camera pose estimates with a scalable volume integration [3, 25]. We show that our method obtains consistent and high-quality reconstructions in challenging low-frame-rate scenarios. We use 1 fps for TUM RGB-D (top row) and 1.5 fps for ScanNet (last 3 rows).

Runtime. On an RTX 3090, our method takes 0.63s for pairwise registration (except I/O, 0.05s SuperGlue) and can be made orders of magnitude faster by JIT and custom kernels for network inference and GN optimization.

Pairwise Evaluation on SuperGlue Test Split. We also evaluate our model on ScanNet test pairs introduced in SuperGlue [20]. While SuperGlue evaluation pairs are sampled with a bias towards higher-overlap pairs (only 19% of pairs have $< 30\%$ overlap, less than 4% of pairs for $< 10\%$ overlap), our method does not suffer but even improves pose recall on this data, as shown in Table 1.

Method	Recall by Overlap %	
	$< 30\%$	$\geq 30\%$
SG + BF [5, 6, 20]	77.39	98.19
Ours (w/ SG + BF)	**80.21**	**98.52**

Table 1. Registration recall by overlap in the SuperGlue [20] ScanNet [4] test pairs @ 15°, 30cm.

Motion Blur in Non-Expert Capture Video. To demonstrate the practical applicability of low FPS registration, in Fig. 3, we show a sample non-expert captured real video sequence with motion blur where two sharp frames are more than 20 frames apart.

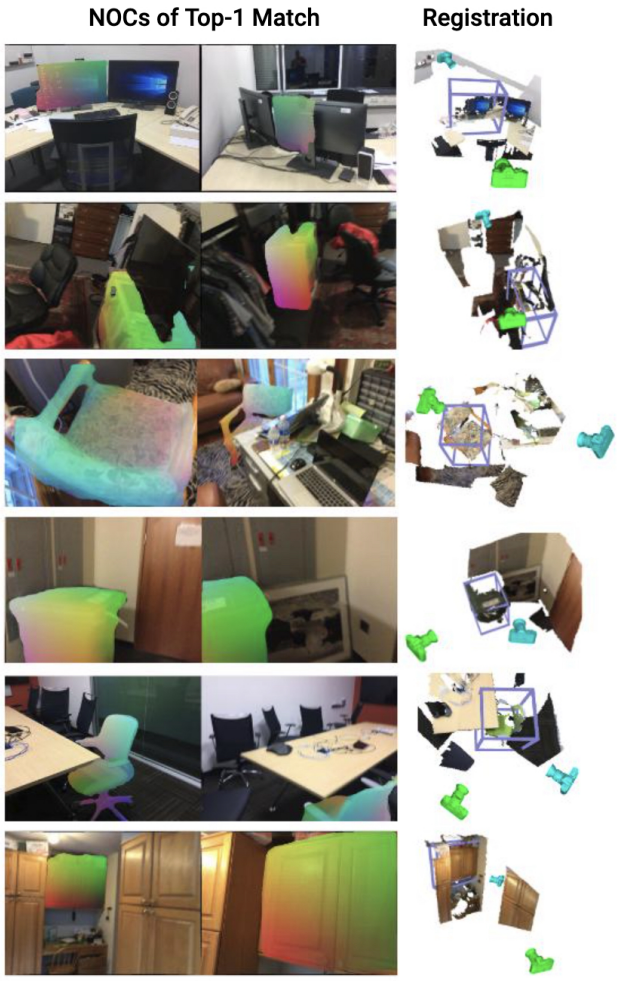

Figure 1. Additional low-overlap registration on ScanNet [4], where traditional feature matching fails. Predicted NOC correspondences are visualized, along with object box and camera poses of the left and right images in green and blue, respectively.

Figure 2. Example scene reconstructions from TUM RGB-D [22] and ScanNet [4] with optimized camera trajectories. Interpolation of camera color from blue to red represents the temporal order of cameras. Geometric reconstructions are obtained using scalable TSDF volume integration [3, 25].

Low FPS Discussion. Lower FPS registration can enable various applications, allowing for more compute budget for other tasks, or enabling the selection of low motion blur frames for reconstruction during fast capture. On the other hand, our approach remains beneficial in higher FPS scenarios, for instance improving ATE RMSE from 5.41 of SG+BF to 5.22 (ours) at 10fps on shorter ScanNet scene718_00, and 21.67 to 16.37 at 3fps on longer ScanNet scene430_00. To handle longer sequences and higher frame rates, we need to incorporate additional hierarchy levels, which we do not employ in this work for a simpler yet fair comparison between different feature-matching and object constraints.

Input Modality	Class Avg.	Instance Avg.
Color	37.33	41.31
Color + Depth	46.46	54.34
Color + Depth + Normal	**48.92**	**55.09**

Table 2. Multi-modal inputs for object recognition. We evaluate Scan2CAD [1] alignment accuracy over ScanNet25k validation images [4]. Adding jet-colored depth input [7] improves performance significantly. Normal input also offers a notable improvement, particularly in category average, helping infrequently-seen categories to generalize better.

Method	≤ 10	(10, 30)	≥ 30
GeoTrans	22.08	64.27	93.11
SG + BF	24.03	73.45	95.95
Ours (Color only)	37.01	77.84	96.50
Ours (Color + Depth only)	42.21	80.64	**97.16**
Ours (final)	**45.45**	**81.44**	**97.16**

Table 3. Pose recall @ 30cm, 15° by overlap % range. Ours refers to Ours (w/ SG + BF), with color, depth, and normal inputs.

B. Ablation Studies

Effect of Multi-modal Inputs on Object Recognition. To measure the effect of multi-modal (color, depth, normals) learning in object recognition, we use a set of per-frame object alignment accuracy metrics over the whole ScanNet25k [4] validation set. We measure Scan2CAD [1] alignment accuracy over object poses as well as standard 2D recognition metrics, restricting the number of possible objects from a category per image instead of per scene. We show the results in Table 2. The local alignments are obtained using the input depths and predicted NOCs. Our multi-modal approach shows notable benefits over color features only.

Effect of Multi-modal Inputs on Pairwise Registration. In Table 3, we measure the effect of multi-modal inputs in the final registration task, using our best method combined with SG+BF. Multi-modal inputs especially benefit in the low-overlap scenario, achieving a registration recall improvement from 37.01% in the color-only case to 45% when using depth and normal inputs.

Effect of Object Identification Context Size. To assess the effect of context size (i.e., the scaling factor of the detected bounding box) on object matching, we evaluate the effect of context size top-1, top-2, and top-3 correctness of best-matching objects between ScanNet [4] validation pairs that have at least 1 shared object. top-2 and top-3 refer to up

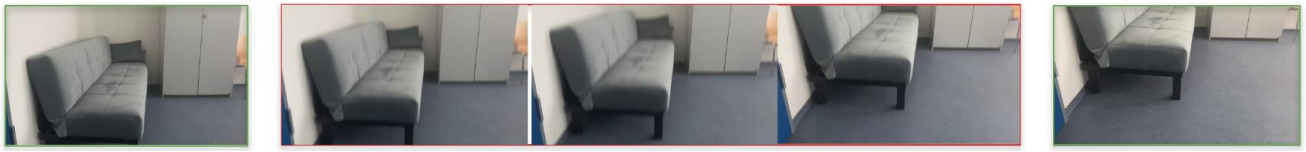

Figure 3. A self-captured sequence of frames with a smartphone. We show that two non-blurry frames on the left and right are apart by $>$ 20 frames, where motion blur is measured using the variance-of-Laplacian thresholding. We show three sample frames in between that are affected by the motion blur. Hence, low FPS registration would have real-world use in scenarios where frames not affected by motion blur must be considered for registration.

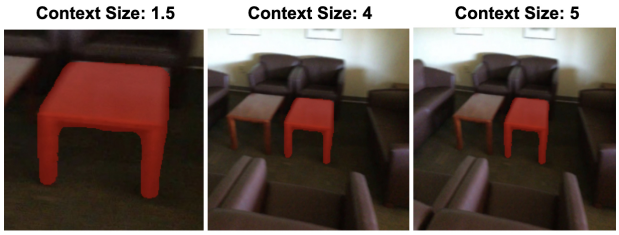

Figure 4. Crops of object regions in different context sizes, where object mask is magnified in red. A larger context size captures more details regarding the whole image while smaller context sizes capture more object details.

Context Size	top-1	top-2	top-3
4	92.86	84.30	**81.27**
5	**93.41**	**85.40**	80.72
6	92.03	84.57	79.89

Table 4. We evaluate top-1, top-2, and top-3 object matching accuracy using different context sizes. Context size is used to scale the detected object box for ROI-cropping for object identification. We show a context size of 5 is a sweet spot for a robust top-1 and top-2 object matching, while a smaller context size is better for top-3 matching.

to 2 and 3, since all pairs may not have that many objects. We consider three different context sizes: 4, 5, and 6. These context sizes c are used to scale the detected object boxes B as cB when used to crop image regions for object identification. We show the effect of context size in Figure 4 and Table 4, and find that $c = 5$ produces the most robust performance.

Effect of Background Context in Object Identification. Without encoding the background and training our identification model that only uses foreground object crops without bounding box scaling, top-1 object identification accuracy drops from 93.41% to 91.21%.

Top-1 vs. Top-2 Filtering. In Table 5, we evaluate the registration recall at different overlap levels for top-1 and top-2 matching object selection. To be selected, objects must be from the same category with mean embedding distance < 0.05. We do not report top-3 matching, since it produces

	Recall by Overlap %		
	≤ 10	$(10, 30)$	≥ 30
Top-1	**52.38**	**81.05**	**99.42**
Top-2	42.86	77.89	**99.42**

Table 5. Recall at 15°, 30cm by overlap percentage. We compare top-1 and top-2 selections of objects in our best method variants, Ours (w/ SG + BF). We show top-1 matching is more robust, especially in the low-overlap cases.

identical results to top-2 due to our other outlier removal strategies. In low-overlap cases, top-1 matching offers additional robustness, while with high overlap the two strategies perform identically. This is expected since in high-overlap frames, identical objects and their contexts look more similar to each other. On the other hand, in the low-overlap regime, there is a higher chance of having ambiguities since different objects may look similar due to viewpoints, and usually, there is a smaller number of matching objects compared to high-overlap cases. Therefore, having only the best-matching object increases the robustness by reducing the chance of error.

Effect of Symmetry Filtering in Pairwise Registration. In Table 6, we show the effect of filtering out the symmetric objects in our method, with and without keypoint constraints. While symmetry filtering helps in both cases, the effect is greater when keypoint constraints are not used, since keypoint constraints help to resolve potential ambiguities caused by the rotational symmetries.

C. Method and Baseline Details

C.1. Data Preparation

We use ScanNet [4] RGB-D frames along with CAD model annotations from Scan2CAD [1] to provide supervision for object NOCs. To train the Mask R-CNN [9]-based NOC prediction network, we use ScanNet400k, a subset of the 2.5m ScanNet RGB-D frames defined by ROCA [4, 8]. To supervise object class categories and their 9-DoF poses, we use the Scan2CAD CAD alignment labels. Different from ROCA, we match the alignment labels to ScanNet's own instance labels, thereby obtaining NOCs and object

Method	Pose Recall
Ours (w/o keypoint, w/o sym flt.)	68.28
Ours (w/o keypoint)	**70.87**
Ours (w/ SG + BF, w/o sym flt.)	86.73
Ours (w/ SG + BF)	**87.38**

Table 6. Pose recall @ 15°, 30cm results with and without filtering out rotationally symmetric objects, where the filtering is enabled by rotational symmetry classification trained alongside NOC prediction. We show that symmetry filtering helps more in the object-only case (w/o keypoints), while still maintaining some improvement when our method is combined with keypoints. This shows that the keypoint constraints help to resolve symmetry ambiguities in object constraints.

labels via an inverse projection of RGB-D depth measurements instead of renderings of CAD models.

To train the object identification network, we sample triplets of objects; each triplet is sampled from the same scene. To ensure wide baseline coverage, we take the positive samples that are at least 100 frames apart. We use the predictions matched with the ground-truth labels to obtain object crops, matched using the Hungarian algorithm over predicted boxes.

C.2. Architecture

Additional Mask-RCNN Backbone Details. We use the weights and configuration of Mask-RCNN-R50-FPN-3x [9, 10, 12, 13, 18] model from Detectron2 [23] as initialization. Our method predicts 32x32 masks instead of 28x28, to increase the resolution of our object correspondences; therefore, our method pools 16x16 feature grids for region proposals. Due to its fully convolutional nature, the default mask head pre-trained on COCO [13] can still be used for initialization. We use a batch size of 4 images with 128 region proposals each for training and fine-tune each layer of the backbone, except the first 2 layers.

NOC Prediction Head. We use a fully-convolutional network that predicts NOCs for every object. We use the same $16 \times 16 \times 256$ pooled features as in the mask prediction. The feature map is first processed by four 3×3 convolutions with channel sizes of 256. The resulting feature map is then upsampled to $32 \times 32 \times 256$ using a single 3×3 convolution that maps the channel size to 1024, followed by a pixel shuffle operator [16, 21]. The upsampled feature map is further processed using a single 3×3 convolutions and two 1×1 convolutions all with hidden sizes 256 modeling a shared MLP. The output is obtained via a final 1×1 convolution that projects the feature map to the desired output channel size of 3. We use ReLU activations for each layer except the output, and a padding of 1 for all 3×3 convolutions. Only NOC values of foreground pixels are considered for training and inference, using the values of ground-truth and predicted segmentation masks, respectively.

Scale Regression Head. We use a fully-connected network (MLP) for regressing 3D anisotropic object scales. We use the same 16×16 feature map as in the NOC head. For efficiency, the network input is first downsampled to $8 \times 8 \times 256$, using a single 5×5 convolution with a stride of 2, and then flattened. Then, we apply two fully-connected layers with a hidden size of 1024. For the 3D scale output, we use a per-category affine layer, similar to the scale prediction head of Vid2CAD [15] and ROCA [8]. That is, the final layer regresses 3 scale values for each of the 9 categories and selects the correct category using the object classification. This enables learning category-specific weights and biases that model the different scale statistics of different categories, e.g., tables being much larger than trash bins or three-person sofas having a different aspect ratio than chairs. All hidden layers use ReLU activations.

Symmetry Classification Head. We use an MLP that is identical to the scale regression head except for the final layer dimension. We use a per-class affine output since the symmetry statistics of each category tend to differ, e.g., trash cans and tables are more often symmetric than chairs.

Object Identification Network. We use a metric learning approach for identifying objects across frames. Our model backbone is built from an ImageNet [18]-pre-trained ResNet18 [10] architecture provided by Torchvision [16]. We use each layer except the final linear layer, and replicate the backbone for both background and foreground in each input modality (i.e., color and depth), without parameter sharing. We concatenate the foreground and background features, sum the concatenated color and depth features, and feed the result to an output network that applies a 2×2 max-pooling, a 2×2 convolution that doubles the feature channels, followed by a global max pooling, and an output linear layer that produces a 1024-dimensional embedding. We use ReLU activations for every hidden layer. We train the network using an initial learning rate of 1e-4, which is decreased by 10 when no improvement has occurred in the last two validation steps; validation is run every 5k iterations, evaluating top-2 matching accuracy between validation image pairs.

Input Depth and Normal Preprocessing. We use multi-modal networks [7] to process the depth inputs for both NOC prediction and object identification. For NOC prediction, we normalize the depth inputs assuming a maximum depth of 10 meters (sufficient for indoor rooms) and then color the depths using inverse jet coloring from Matplotlib [11] such that red represents near and blue represents far since orange/yellow tones are more common than blue tones in real indoor images. We also estimate normals from depths, using bilateral filtering followed by a nearest neighbor downsampling for depth, followed by a smooth normal

estimation using 5×5 Sobel filters. We observe that using normals with half the size of the depth map improves computational efficiency and model accuracy, likely due to their smoothness. Normals are directly colored to RGB by mapping the $[-1, 1]$ range to $[0, 255]$ range. For object identification, we also use inverse jet-colored depths. However, we normalize the depths using the maximum depth value observed in the image rather than using the absolute maximum depth.

C.3. Energy Optimization on RGB-D Sequences

Optimization for Temporal Sequence Registration. Due to the temporal nature of sequence data, we apply adjusted thresholds from the pairwise registration scenario. In filtering consecutive frames, we use a 30cm instead of a 20cm threshold in BundleFusion Kabsch filtering [5] threshold for keypoint matches to ensure consecutive frames can be registered, with other outlier removal thresholds remaining the same. For non-consecutive frames that may contain potential loop closures, we make the constraints stricter, using a 15cm threshold for BundleFusion Kabsch filtering. We also use a 0.04 threshold for object matching to ensure further robustness in loop closures.

Loop Closure Outlier Rejection. We apply various filters to accept or reject loop closures, i.e., frame pair matches that are not consecutive. We only apply loop closure filters for ScanNet data [4], since TUM RGB-D [22] scenes are relatively small.

When using loop closures with objects, we only accept loop closures where objects' optimized depths (z-dimension of the translation) are below 2.15m in at least one of the frames in a frame pair. We also ensure the z-dimension of the translation is positive in at least one of the matching frames. To apply such translation filtering, we transform the optimized global object poses to the local object poses using the optimized camera pose. We also filter out degenerate object optimizations by filtering out any object with an optimized scale dimension less than 0.05.

For loop closure without objects, we apply a global translation filtering by rejecting loop closures whose optimized relative camera translation is too large. That is, we allow a maximum of 60cm translation in nearby loop closures (within the consecutive 20 frames) and 1.5m for loop closures that are farther away. We exclude loop closure edges that do not adhere to these constraints from the global pose graph optimization.

Pose Graph Optimization Details. We use the default global Gauss-Newton pose graph optimizer from Open3 [2, 25]. For ScanNet [4], we use a maximum correspondence distance of 0.1, an edge prune threshold of 0.45, and use a 100% preference rate for loop closures. For TUM RGB-D [22], we use loop closure preference of 35% with all other hyper-parameters the same.

Pose Graph Restructuring. We also make some adjustments to the standard pose graph structure of [2] to handle low-frame-rate scenarios more robustly. We make spatially distant consecutive edges uncertain, using a translation threshold of 40cm in TUM RGB-D [22] and 50cm in ScanNet [4]. We also make some non-consecutive edges certain if they have a translation of less than 4.5cm. This helps to overcome distant consecutive frames by assigning nearby loop closures as parents, helping significantly in performance.

C.4. Baselines

GeoTransformer. We evaluate the 3DMatch [24] pretrained Geometric Transformer [17], since re-training or fine-tuning on our ScanNet [4] pairs did not empirically help in terms of further generalization. We use the standard LGR optimizer to obtain results since it worked slightly better than RANSAC in our experiments. When combining with our method, we integrate the weighted feature matches to our Gauss-Newton optimization following a BundleFusion-style [5] outlier removal, similar to our handing of SuperGlue baseline [6, 20].

Redwood Optimization Details. We use the best-performing pose-graph hyperparameters for the Redwood (Global Registration) [2] method to obtain maximum robustness in different datasets of SLAM sequences. First, we cancel the graph restructuring process since it does not provide any benefits. For ScanNet [4], we use edge prune threshold 0.25 and max correspondence distance 1.0, with loop closure preference of 100% for long sequences (>115 frames) and 30% for other sequences. For TUM RGB-D, we change the edge prune threshold to 0.5 and the loop closure preference rate to 50%. The baseline initially uses FPFH [19] features with RANSAC, but it falls back to ICP for odometry edges with < 0.5 convergence score.

BundleFusion SIFT Optimization Details. We use the best-performing hyperparameters for the SIFT + BF [5, 14] approach for sequence registration. We use a Procrustes threshold of 50cm for ScanNet and 30cm for TUM RGB-D, respectively, for odometry cases, and 30cm for ScanNet and 20cm for TUM RGB-D for loop closures. We use max correspondence distance and edge pruning threshold of 0.5 for ScanNet and 0.5 and 3 for TUM-RGBD. We also use graph re-structuring in ScanNet, using a 4cm threshold to make loop closure edges certain. For both TUM-RGBD and ScanNet, we use a 1m threshold in nearby frames and 50cm and 75cm thresholds in far-away frames to make them uncertain. We refer to Open3D [25] registration pipelines for further detail.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 2614–2623, 2019. 2, 3

[2] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015. 5

[3] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 1, 2

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2, 3, 5

[5] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 1, 5

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018. 1, 5

[7] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 681–687. IEEE, 2015. 2, 4

[8] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: Robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4022–4031, 2022. 3, 4

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3, 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[11] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. 4

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[14] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 5

[15] Kevis-Kokitsi Maninis, Stefan Popov, Matthias Niesser, and Vittorio Ferrari. Vid2cad: Cad model alignment using multi-view constraints from videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 4

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4

[17] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11143–11152, 2022. 5

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4

[19] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009. 5

[20] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 1, 5

[21] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 4

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 1, 2, 5

[23] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4

[24] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017. 5

[25] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 1, 2, 5