Supplementary Material for ROCA: Robust CAD Model Retrieval and Alignment from a Single Image

Can Gümeli Angela Dai Matthias Nießner

Technical University of Munich

1. Additional Results

1.1. Qualitative Evaluation on iPhone Images

We apply our approach trained on ScanNet+Scan2CAD to images taken with an iPhone 11 (rear camera) of various office environments. As shown in Figure 1, our method achieves high-quality alignment results in complex scenes.

1.2. Unconstrained Retrieval

Following previous research [2, 3, 13], we retrieve CAD models from a given scene pool in our main results. In Figure 2, we consider unconstrained retrieval where CAD models can be retrieved from the whole database of CAD models that appear in the training data. Note that the CAD models that only appear in the validation set are omitted.

In the unconstrained setting, the number of unique CAD models from benchmark categories is $\approx$ 2300. Similar to the standard approach, we pre-compute CAD embeddings prior to inference. Using per-category nearest-neighbor lookup from this large set, our method operates at 59 milliseconds per image at inference ($\approx$ 17 frames per second). This is a 2-frame performance drop compared to the $\approx$ 19 frames per second achieved with constrained CAD model pools. Even when retrieving from the large set of more than 2000 CAD models, our method still achieves interactive frame rates and has the potential for real-time applications.

2. Further Implementation Details

2.1. Depth Estimation Head

To predict depth for an input image, we use the multi-scale future fusion (MFF) module from [9]. We adopt MFF to use FPN [12] features instead of ResNet [8] features directly, and omit the up-convolutional part for computational efficiency. Moreover, we use a pixel-shuffle [17] layer for a learnable, parametric up-sampling as opposed to the commonly used bilinear interpolation performed only in post-processing [9, 11].

Figure 3 diagrams our depth prediction head, and its sub-components are given in Figure 4.

To optimize depth estimation, we use the reverse Huber

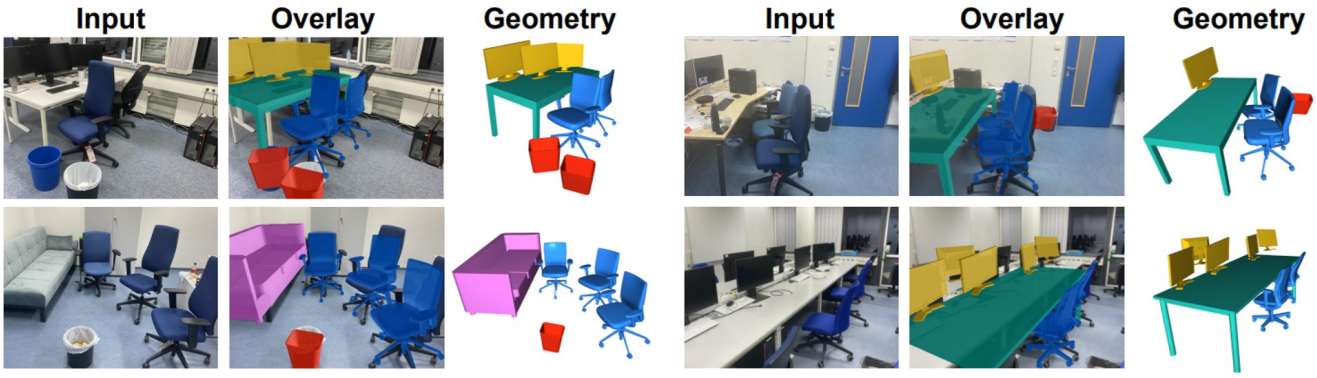

Figure 1. **Sample Alignments from Smartphone Images.** We take photos of a real work environment from an Iphone 11 rear camera. Our method achieves high quality alignments to complex scenes with multiple objects.

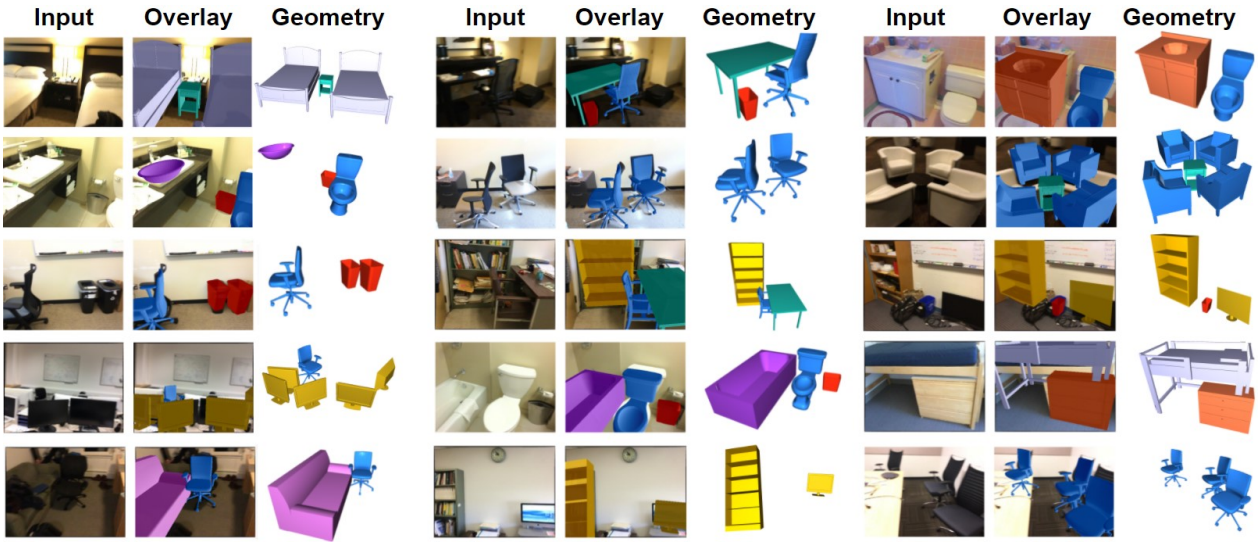

Figure 2. **Unconstrained Retrieval.** We show results from ScanNet validation set [5], where the candidate retrieval pool covers the full training set. Our method shows promising generalization capability in this challenging setup.

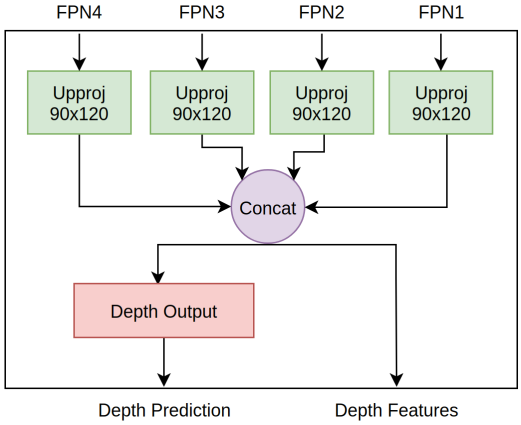

Figure 3. **Overview of Depth Prediction**. Adopted from [11], up-projection (Upproj) layers up-samples of to a given spatial resolution. The resulting multi-scale features are combined via concatenation and used in depth estimation and alignment pipeline.

(berHu loss) [11], whose definition is

$$\text{berHu}(x) = \begin{cases} |x| & |x| \leq c, \\ \frac{x^2+c^2}{2c} & |x| > c, \end{cases} \tag{1}$$

where c is set adaptively to $\frac{1}{5}$th of the loss values in a grid. This loss allows low-error predictions to get more accurate by down-weighting the outliers during training.

2.2. Retrieval Network

We show the retrieval part of our architecture in Figure 5, which is inspired by the joint scan-CAD embedding of [4].

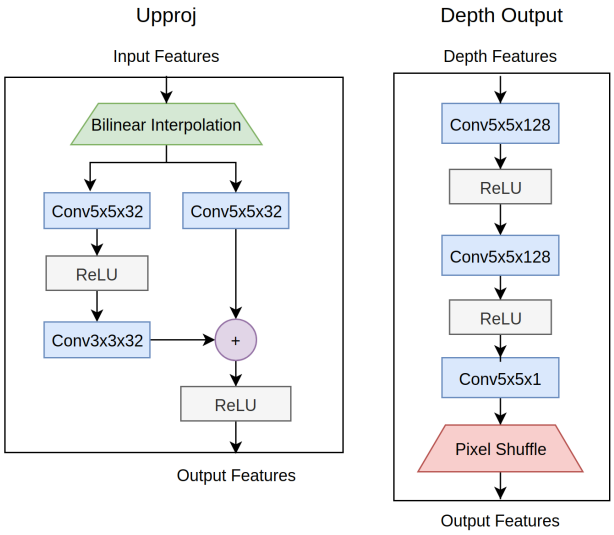

Figure 4. **Depth head modules**. Convolutions are described with their filter width, filter height, and output channels, respectively. Pixel shuffle layers up-sample spatial dimension by 4x4 (16 channels). The bilinear interpolation in Upproj layer adaptively rescales the input features to quarter of the image, 90x120 in out case, resolution irrespective of the input size. We use 32-channel feature maps at each up-projection following [9].

2.3. Data Preparation

We render Scan2CAD alignment labels [2] over ScanNet images [5] using a simple rasterization pipeline [1]. Before rendering, alignment labels are projected to the image camera coordinate systems, using the inverse of the camera pose

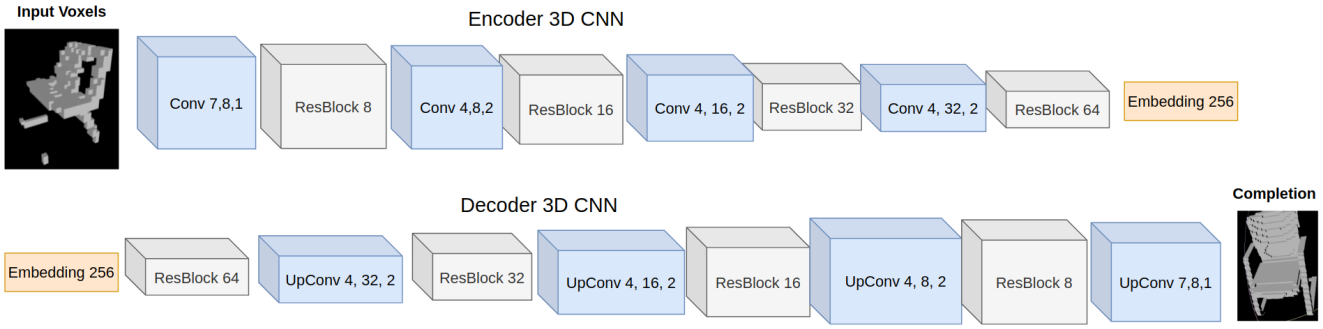

Figure 5. **NOC-based Retrieval of CADs to Images.** We use our predicted NOCs to enable 3D-based joint embedding of real-world observations and CAD models, by interpreting the NOCs as a voxelization in the canonical space. Conv and UpConv represent convolution and transpose convolution layers, with comma-separated values represent filter size, output channels, and stride, respectively. Each Conv and UpConv layer are followed by a ReLU activation. ResBlocks represent ResNet basic blocks [8] that use 3D convolutions. The result of the encoder and input for the decoder is a 256-dimensional embedding vector.

provided by the ScanNet [5].

Following Mask2CAD [10], we filter out objects whose centers are outside the image frame. Furthermore, no ScanNet labels except for Scan2CAD alignments, camera poses and camera intrinsics are involved in the rendering pipeline. Thus, the rendering and supervision of our method is consistent with and directly comparable to the previous work [10, 13].

In the dense frame sampling experiment, we extracted images from ScanNet raw sensor stream and perform the same data processing.

2.4. Total3D Training Details

We re-train the pose estimation component of Total3D [14] on our data for comparison. Since the method relies on pre-computed object detections, we first train a ResNet50 Faster-rcnn [12, 18] initialized from ImageNet and COCO pretraining. Then, we match the pre-computed detections with our labels based on bounding box IOUs. Also using the pre-computed detections, we extract the relevant geometry features for the pose estimation pipeline [14].

Since the rotation component of total3d depends on the room layout that we do not have, we simply use the ground-truth rotation.

We use an SGD optimizer similar to our main training. We train for the total of 60k iterations, decaying learning rate at 40k, based on our tuning of the model.

2.5. Used Open-Source Libraries

We utilize various open source libraries for our model trainings and data pre-processing. Our model is implemented using PyTorch, Detectron2, and Pytorch3D [15, 16, 18], without declaring any custom low-level kernels outside of these libraries. We make our geometry visualizations using Open3D [19] and CAD voxelizations using Trimesh [6].

For the baseline methods without learned retrieval, we performed single sided Chamfer Distance lookup over point clouds sampled using the farthest points sampling implementation from PyTorch Cluster [7].

References

[1] Rasterization: a practical implementation. https://www.scratchapixel.com/lessons/3d-basic-rendering/rasterization-practical-implementation. Accessed: 2021-05-13. 2

[2] 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. 1, 2

[3] Armen Avetisyan, Angela Dai, and Matthias Nießner. End-to-end cad model retrieval and 9dof alignment in 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2551–2560, 2019. 1

[4] Manuel Dahnert, Angela Dai, Leonidas J Guibas, and Matthias Nießner. Joint embedding of 3d scan and cad objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8749–8758, 2019. 2

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

[6] Dawson-Haggerty et al. trimesh. 3

[7] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 3

[8] 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. 1, 3

[9] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher

resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019. 1, 2

[10] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2cad: 3d shape prediction by learning to segment and retrieve. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 260–277. Springer, 2020. 3

[11] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016. 1, 2

[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. 1, 3

[13] Kevis-Kokitsi Maninis, Stefan Popov, Matthias Nießner, and Vittorio Ferrari. Vid2cad: Cad model alignment using multi-view constraints from videos. *arXiv preprint arXiv:2012.04641*, 2020. 1, 3

[14] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 55–64, 2020. 3

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 3

[16] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 3

[17] 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. 1

[18] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3

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