SPARC: Sparse Render-and-Compare for CAD model alignment in a single RGB Image - Supplementary Material

Florian Langer
fml35@cam.ac.uk
Gwangbin Bae
gb585@cam.ac.uk
Ignas Budvytis
ib255@cam.ac.uk
Roberto Cipolla
rc10001@cam.ac.uk

Department of Engineering
University of Cambridge
Cambridge, UK

We provide additional information for various aspects of our main work. In Sec. 1 we present alignment accuracies evaluated with the orignal (wrong) evaluation code and the corrected one. In Sec. 2 we demonstrate the effectiveness of sparse sampling at different image resolutions. In Sec. 3 and Sec. 4 we give additional information for training our pose prediction network, SPARC-Net, and the networks used for depth and surface normal estimation. In Sec. 5 we highlight issues when trying to align differently shaped CAD models with each other and the resulting systematic offsets that appear in the predictions of the normalised object coordinates. We quantitatively support this section by ablating our system with ROCA [7] predictions in Sec. 6. Limitations and possibility for future works are discussed in Sec. 7. In Sec. 8 we visualise pose predictions for inaccurate bounding box predictions and finally in Sec. 9 we explain a released video showing extra qualitative results on ScanNet [6].

1 Correction to Evaluation Script

Scan2CAD [2] proposed to consider a CAD alignment correct if the object class prediction is correct, the translation error is less than 20 cm, the rotation error is less than 20 degrees and the scale ratio is less than 20 %. We found that there was a bug in the original evaluation code which was subsequently used to evaluate ROCA [7] and Mask2CAD [9]. When computing the scale ratio the formula $s_{\text{error}} = |\sum_{i=x,y,z}(s_i^{\text{pred}}/s_i^{\text{gt}} - 1)|$ was used instead of $s_{\text{error}} = \sum_{i=x,y,z}|(s_i^{\text{pred}}/s_i^{\text{gt}}) - 1|$ which allowed scale errors in different directions to cancel each other out. We correct for this mistake and reevaluate [7, 9]. The accuracy computed with the evaluation code containing the mistake are presented in Table 1 and their corrected counterparts are presented in Table 2.

Method	bathtub	bed	bin	bkshlf	cabinet	chair	display	sofa	table	**class**	**instance**
Mask2CAD-b5 [9]	8.3	2.9	25.9	3.8	5.4	30.9	17.3	5.3	7.1	11.9	17.9
ROCA [8]	22.5	10.0	29.3	14.2	15.8	41.0	30.4	15.9	14.6	21.5	27.4
Ours	26.7	25.7	26.7	17.5	23.8	52.6	22.5	32.7	17.7	27.3	33.9
Ours + ROCA rot init	27.5	30.0	41.4	17.5	23.5	53.7	26.2	32.7	22.6	30.6	36.8

Table 1: **Alignment accuracy with original (incorrect) treatment of scale predictions** on ScanNet [6].

Method	bathtub	bed	bin	bkshlf	cabinet	chair	display	sofa	table	**class**	**instance**
Mask2CAD-b5 [9]	7.5	2.9	24.6	1.4	5.0	29.9	13.1	5.3	5.6	10.6	16.7
ROCA [8]	20.8	8.6	26.3	9.0	13.1	39.9	24.6	10.6	12.7	18.4	25.0
Ours	25.8	25.7	24.6	14.2	20.8	51.5	17.8	28.3	15.4	24.9	31.8
Ours + ROCA rot init	25.0	30.0	36.2	14.2	19.2	52.3	20.4	28.3	20.1	27.3	34.1

Table 2: **Alignment accuracy with correct treatment of scale predictions** on ScanNet [6].

2 Investigating Sparse Sampling for Different Image Resolutions

We investigate the effect of sparse sampling at different image resolutions. Particularly, we test the claim that using less than 1% of the available pixels can give accurate pose estimates through render-and-compare. We test 3 image resolutions with $(\text{width}, \text{height})$ set to $(240, 180)$, $(480, 360)$ and $(640, 480)$. Compared to the main experiment we use a simplified setup where we do not explicitly include 2D image information from the reprojected pixel locations (i.e. $N_{\text{reproj}} = 0$) or context image information from outside the bounding box ($N_{\text{context}} = 0$). We set $(N_{\text{CAD}} = 100, N_{\text{bbox}} = 330)$, $(N_{\text{CAD}} = 500, N_{\text{bbox}} = 1200) =$ and $(N_{\text{CAD}} = 1000, N_{\text{bbox}} = 2000)$ for the three image resolutions respectively. These values were chosen such that the sum of N_{CAD} and N_{bbox} is just less than 1% of the total number of pixels at the given resolution. Here we find that even at the smallest resolution we obtain an average instance accuracy of 30.7% (compared to 31.8% for the main experiment). For the resolutions $(480, 360)$ and $(640, 480)$ we obtain average instance accuracies of 30.2% and 30.8% respectively. Those results show that sparse sampling is indeed effective at various image resolutions and even for small image sizes sampling less than 1% of the available pixels can give accurate pose estimates.

3 Additional Train and Test Information

Train data generation. We generate per-image CAD model pose annotations for all images in ScanNet25k [6] (which is the original ScanNet[6] dataset sampled for every 100th frame) by transforming CAD model pose annotations from [2] from ScanNet [6] world coordinates into camera coordinates. ScanNet25k [6] contains ca. 20K train images from 1200 different scenes. Note that for a given image we only train our CAD model to align CAD models whose center is reprojected into the image (such as to avoid training on objects that are barely visible). Further, we filter objects which do not have at least 50% of their reprojected depth values within 30 cm of the ground truth depth values. This avoids training on objects that are hidden behind walls or in other way strongly occluded.

Pose sampling at train time. When initialising poses for training we uniformly sample

translations $\mathbf{T}$ to be between 1 and 5 m along the Z-axis and sample X and Y by uniformly sampling pixel coordinates in the predicted bounding box and mulitplying the corresponding pixel bearing by the sampled Z to obtain X and Y values. Scale values $\mathbf{S}$ are sampled uniformly in the range of all observed CAD model scales for the detected category. 25% of train examples are uniformly sampled in random rotation (any azimuthal angle, 20° tilt range and 40° elevation range around $\mathbf{R}^{\text{gt}}$) to learn to classify rotations, whereas 75% of train examples are sampled in the correct rotation bin (90° azimuthal angle range, 20° tilt range and 40° elevation range around $\mathbf{R}^{\text{gt}}$) to learn to regress pose offsets.

At test time the pose prediction network is used to iteratively refine its own predictions. During training it is therefore crucial to not just generate random CAD model poses, but also poses based on the networks predictions. This ensures that the poses sampled during training are as similar to the ones the network sees at test time as possible. For every image and corresponding CAD model annotation we therefore use a randomly initialised pose as well as the two subsequent refinements predicted by the network as training data. Specifically, for a batch of N train examples, each containing image information and CAD model information sampled in a random initial pose, we predict the pose updates and apply losses. Based on the predicted pose updates, we update the CAD model poses for every train example and recompute the inputs. The recomputed inputs are fed through the network again, pose updates are predicted and losses applied. This process is repeated once more, after which new training images with new CAD models in random initial poses are sampled as the next training examples.

Test details. At test time $\mathbf{R}$ is initialised with four rotations at 0, 90, 180 and 270 degrees around the vertical axis, 0 degrees tilt and 20 elevation angle (such that the camera is looking slightly down at an object that is standing straight upright). $\mathbf{T}$ is initialised to have $z = 3$ m and x and y such that the reprojected $\mathbf{T}$ lies at the bounding box center. The scale $\mathbf{S}$ is initialised with the median value of all CAD models for the given category. For the rotation with the highest classification score c we predict $(\Delta\mathbf{T}, \Delta\mathbf{R}, \Delta\mathbf{S})$ and iteratively refine the pose 3 times. Note that our pose prediction network is very robust to poor initialisation for scale and translation (see the video explained in Sec. 9) but can not reorient CAD models if their are initialised within the wrong 90° rotation bin.

4 Details for Training Surface Normal and Depth Networks

For both surface normal $\mathbf{N}_{\text{Img}}$ and depth estimation $\mathbf{D}_{\text{Img}}$, we use a light-weight convolutional encoder-decoder architecture from [1]. For both tasks, we predict the per-pixel probability distribution for the output and supervise the network by minimizing the negative log-likelihood (NLL) of the ground truth. For surface normal, we parameterize the distribution using the Angular vonMF distribution, proposed in [2], while we parameterise the depth distribution with a gaussian distribution. After training, we discard the uncertainty and only use the predicted mean values. We use ground truth surface normals provided by [3] and ground truth depth as provided by ScanNet [6], respecting the train/test split. For depth we train on all two million available train images, while for surface normals we train on all images for which [3] provide annotations and that are within the set of train images which results in ca. 200K train images. We train both networks for ten epochs using the AdamW optimiser [11] and schedule the learning rate using 1cycle policy [12] with $lr_{\max} = 3.5 \times 10^{-4}$ (same

as [5]). We use a batch size of four for both surface normal and depth training.

5 Aligning CAD Models for Normalised Object Coordinates

ROCA [7] relies on predicting Normalised Object Coordinates (NOCs) for each pixel of the detected objects. NOCs are 3D object coordinates in a canonical frame in which objects have been aligned. However, here we demonstrate that aligning different object shapes with each other is not trivial. Figure 1 shows that even for two very similar shapes different alignments are possible depending on which object parts one wishes to align. This means that NOCs learned for one shape do not generalise well to other shapes. When attempting to predict NOCs we observe that ROCA [7] often predicts NOCs with a systematic offset (see Figure 2). Here we show the NOCs predicted overlayed in the canonical object frame. One can see that the 3D coordinates predicted are often systematically above or below the actual object. This means that even though the reprojected NOCs roughly match the objects in the image the corresponding object alignment is very wrong.

6 Ablating SPARC with ROCA Predictions

The previous section showed that ambiguities in aligning different shapes with each other can lead to systematic offsets when predicting NOCs. A second issue with ROCA [7] is that object scale is directly regressed which is inaccurate and can produce very wrong scale estimates subsequently leading to poor translation estimates. We demonstrate both of these issues quantitatively by replacing either the scale, translation or rotation prediction of our system with ROCA's [7] prediction (see Table 3, lower half). The top half of the table is a copy of our main results table and is for reference only. Note that the row "Ours + ROCA rot init" uses the ROCA rotation predictions as an initialisation which is subsequently refined by our own predictions. In contrast the rows "Ours + ROCA rotation/translation/scale" show results when replacing the respective predictions with ROCA's predictions and keeping them fixed during the refinement process. We observe a noticeable drop in alignment accuracy when replacing our translation or scale predictions with ROCA's confirming the issues presented above. ROCA's rotation predictions are largely unaffected by inaccurate scale predictions and systematic offsets in NOCS and we observe that they perform similarly to ours.

7 Limitations

This section lists limitations of the current approach which we plan to address in future works.

Dense depth and surface normal predictions. Currently our method uses dense depth and surface normal estimates that are precomputed. If depth and surface normals are computed in realtime, predicting them sparsely only for relevant pixel coordinates may reduce inference time. Further for real applications our method could make use of additional sensory

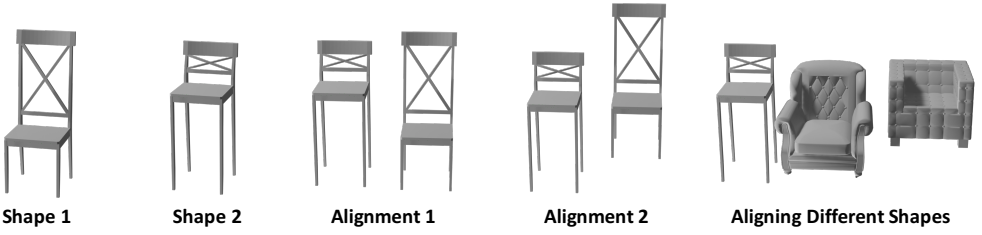

Figure 1: **Issues with shape alignment.** When trying attempting to align Shape 1 and Shape 2 with each other different alignments are possible. Alignment 1 aligns the top and the bottom of the chairs. This ensures that both shapes fit into the same normalised 3D bounding box and is the alignment used for NOCs. Alignment 2 aligns the chairs by their seating area. While now both shapes do not lie in the same normalised 3D bounding box, their seating areas align which is useful when trying to predict their coordinates. The point is that for different shapes different alignments are possible depending on which object parts one wishes to align. The more different the shapes are the harder it is to find some global alignment that aligns all different object parts with each other. This means that NOCs learned for one shape do not generalise well to NOCs learned for other shapes.

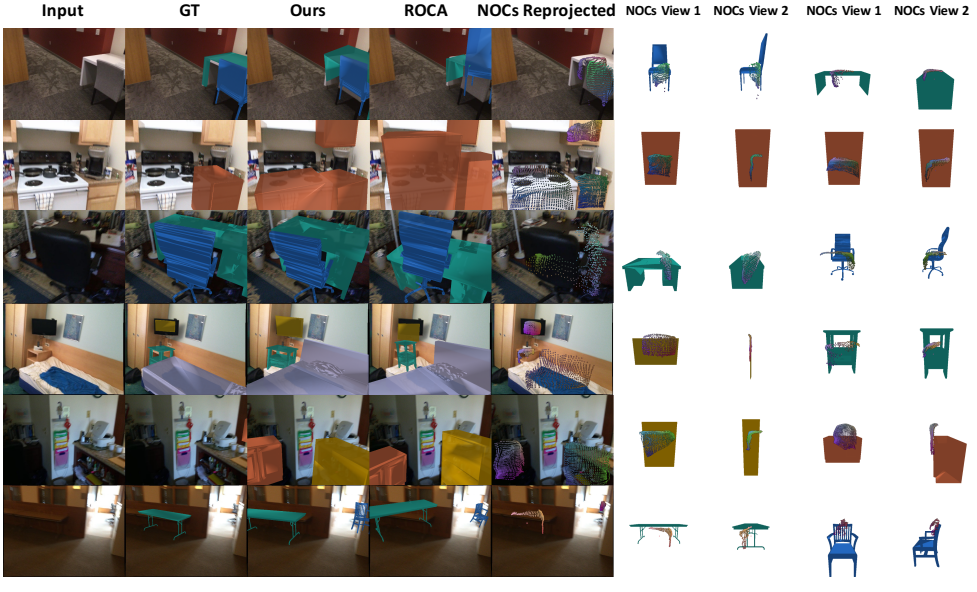

Figure 2: **Visualisation of ROCA's NOCs predictions.** For different inputs we show the GT CAD model alignment, our prediction and ROCA's prediction. Further we show the estimated NOCs overlayed in the canonical object frame from different views for relevant objects. We also show the NOCs reprojected back into the image under the predicted pose. One can observe that the predicted NOCs are often systematically offset from the actual 3D shape. Therefore even though the NOCs reprojected under the estimated pose roughly match the images the corresponding CAD alignments can be very wrong.

Method	bathtub	bed	bin	bkshlf	cabinet	chair	display	sofa	table	**class**	**instance**
Number of Instances #	120	70	232	212	260	1093	191	113	553	9	2844
ROCA [7]	20.8	8.6	26.3	9.0	13.1	39.9	24.6	10.6	12.7	18.4	25.0
Ours	25.8	25.7	24.6	14.2	20.8	51.5	17.8	28.3	15.4	24.9	31.8
Ours + ROCA rot init	25.0	30.0	36.2	14.2	19.2	52.3	20.4	28.3	20.1	27.3	34.1
Ours + ROCA rotation	19.2	22.9	29.3	13.2	20.0	46.1	16.8	27.4	15.9	23.4	29.6
Ours + ROCA translation	20.8	10.0	22.8	9.0	11.5	40.7	14.1	17.7	11.4	17.6	24.2
Ours + ROCA scale	1.7	24.3	22.8	6.6	12.7	43.9	17.3	21.2	9.6	17.8	24.9

Table 3: **Ablation of SPARC-Net with ROCA predictions.** The top half repeats the results presented in the main paper. The lower half shows results when we replace one of our predictions (rotation, translation or scale) with ROCAs [7] prediction. Note that those predictions are not refined with our own predictions. Row 3 "Ours + ROCA rot init" in contrast uses ROCA's rotation prediction as an initialisation which is subsequently refined with our own predictions.

information such as LiDAR which provides naturally sparse depth inputs that can easily replace predicted depth values in our pipeline. Further for video applications event cameras [10] may be of particular interest as these are extremely fast and energy efficient as they only react to changes in light intensity, therefore providing sparse image data containing object edge information that is crucial for 3D shape estimation or CAD model alignment.

Rotation predictions. Currently our handling of rotation is not elegant as at test time it requires four extra forward passes through our network to determine the initialisation of the rotation. This aspect could be improved by reprojecting the four different rotation initialisation simultaneously while at the same time predicting pose updates for all of them, but then only applying those pose updates to the rotation initialisation with the highest estimated probability.

CAD model refinements. Currently, our approach is limited to refining an initial object pose, but not the initial object shape. In general all retrieval-based approaches for shape estimation are limited by the availability of a fitting CAD model. However, even with growing CAD model databases it is unrealistic that every object in the real world will have a precisely fitting CAD model in the database. Therefore it is important to deform a retrieved CAD model to better fit an observed object. This could be nicely achieved with the presented framework by in addition to 9 DoF pose updates predicting N-dim shape updates where N are the numbers of parameters of some shape transformation. One possibiliy for such a shape transformation function are neural cages [13].

Joint shape and pose predictions. Currently for reconstructing scenes every object is treated individually. However, this neglects important information contained within object-object relationships. These can contain information about the pose (e.g. two tables that are standing next to each other are likely to be aligned with each other) or the shape (e.g. chairs around a table are likely to have the same shape which can be a very important signal when dealing with strong occlusion.). Taking into account such information (e.g. by modelling it as a scene-graph [3] will further improve our shape and pose predictions.

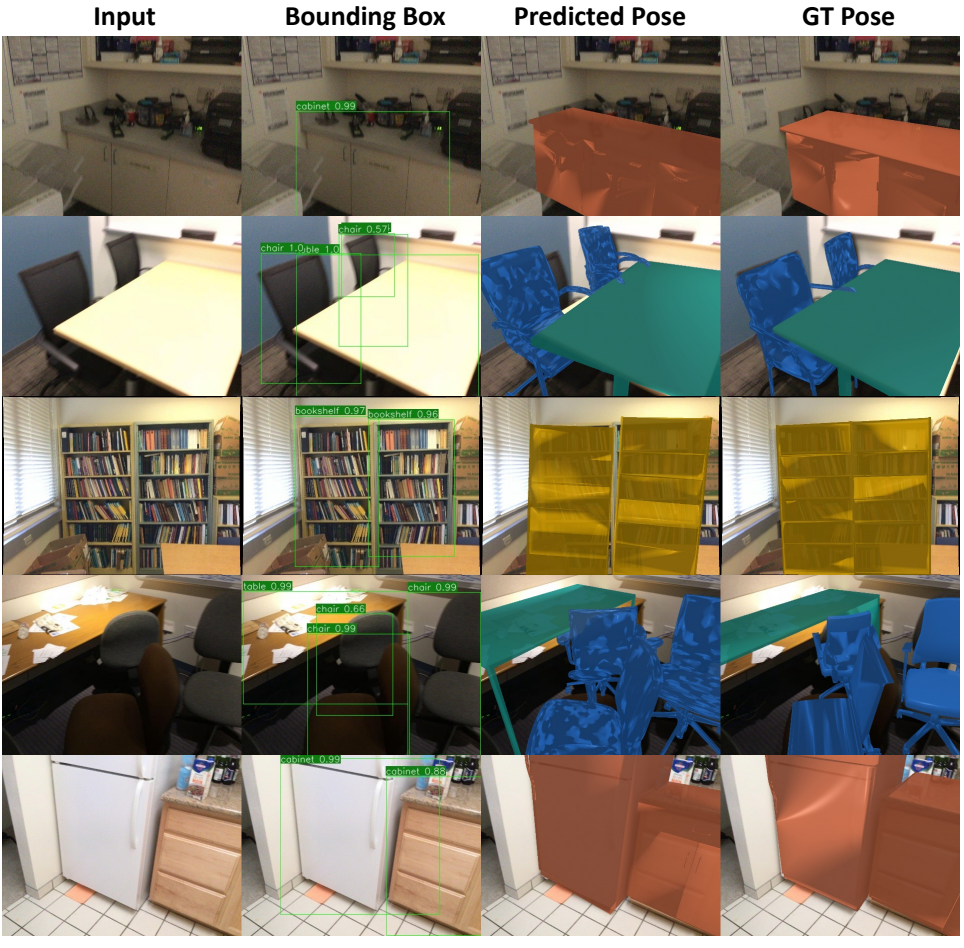

Figure 3: **Predicted alignments for inaccurate and partial bounding box detections.**

8 Visualisations for Inaccurate and Partial Bounding Box Detections

In Fig. 3 we show additional qualitative visualisations when the bounding box predictions are inaccurate. In row 1 one can see that even though the bounding box prediction only extends over a part of the kitchen cupboard SPARC-Net correctly predicts a shape for the entire cupboard (similarly in row 3 and row 5). Furthermore, both the predicted poses for the chair in the front in row 2 and the table in row 4 result in their reprojected shapes reaching outside of the predicted bounding box and therefore leading to better poses compared to if the poses had been confined to lie within the bounding box.

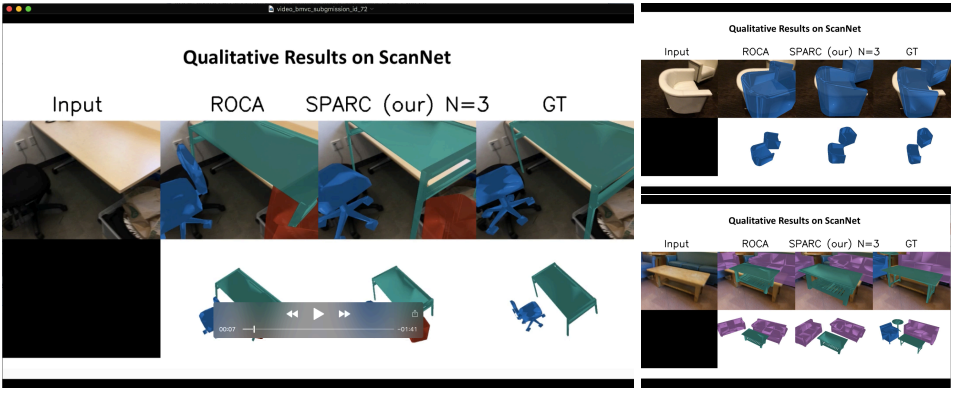

Figure 4: **Visualisation Video.** We visualise intermediate refinement steps on ScanNet [6] in a video (https://youtu.be/eVVW__0QGnM).

9 Visualisation Video

In the visualisation video (https://youtu.be/eVVW__0QGnM) one can see nicely that our pose prediction network, SPARC-Net, is very robust to rough initialisation and able to significantly translate, rotate and scale CAD models to fit the objects observed in the image. The results demonstrate the advantage of an iterative procedure: the first refinement is usually a large pose update, transforming the often very bad initialisation to roughly match the pose of the object in the image. The second and third refinement in contrast are smaller pose updates that really align the CAD model with the objects in the image.

Note that for some images in the video the number of ground truth CAD models and the number of CAD models for which the pose is predicted do not correspond exactly, either due to missing 2D object detections or because of missing ground truth annotation as [2] did not provide exhaustive CAD model annotation for ScanNet [6].

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

[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 *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019.

[3] Armen Avetisyan, Tatiana Khanova, Christopher Choy, Denver Dash, Angela Dai, and Matthias Nießner. Scenecad: Predicting object alignments and layouts in rgb-d scans. In *The European Conference on Computer Vision (ECCV)*, August 2020.

[4] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13137–13146, 2021.

[5] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2842–2851, June 2022.

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[7] Can Gümeli, Angela Dai, and Matthias Nießner. Roca: Robust cad model retrieval and alignment from a single image. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2022.

[8] Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas J Guibas. Framenet: Learning local canonical frames of 3d surfaces from a single rgb image. In *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[9] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2CAD: 3D Shape Prediction by Learning to Segment and Retrieve. In *Proc. 16th European Conference on Computer Vision*, Glasgow, UK, August 2020.

[10] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128$\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008. doi: 10.1109/JSSC.2007.914337.

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

[12] Leslie Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates. *ICLR*, 2018.

[13] Wang Yifan, Noam Aigerman, Vladimir G. Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. Neural cages for detail-preserving 3d deformations. In *CVPR*, 2020.