

# ViewSynth: Learning Local Features from Depth using View Synthesis

## - Supplementary Material

Jisan Mahmud<sup>1</sup>

jisan@cs.unc.edu

Rajat Vikram Singh<sup>2</sup>

singh.rajat@siemens.com

Peri Akiva<sup>3</sup>

peri.akiva@rutgers.edu

Spondon Kundu<sup>2</sup>

spondon.kundu@siemens.com

Kuan-Chuan Peng<sup>4</sup>

kpeng@merl.com

Jan-Michael Frahm<sup>1</sup>

jmf@cs.unc.edu

<sup>1</sup> University of North Carolina at Chapel Hill, Chapel Hill, NC

<sup>2</sup> Siemens Corporate Technology, Princeton, NJ

<sup>3</sup> Rutgers University, New Brunswick, NJ

<sup>4</sup> Mitsubishi Electric Research Laboratories, Cambridge, MA

## 1 Additional methodology details

### Fully connected and convolutional residual blocks

Figure 1 illustrates the residual fully connected block that we use in **GTE**, and the residual convolutional block that we use in **DSN**.

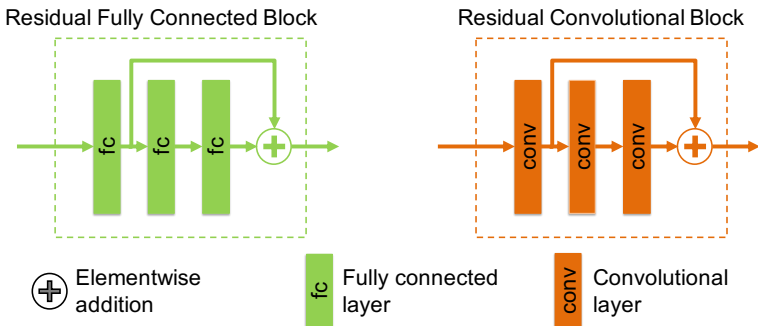


Figure 1: The architecture of residual fully connected block; and the residual convolutional block that are used in the ViewSynth framework.

## Mapping and warping Functions

Mapping grid  $M^{1 \rightarrow 2}$  introduced in Section 3.2 is computed using  $I'^{(2)}$  and camera parameters  $C^{(1)}$ ,  $C^{(2)}$ . We use intrinsic matrix  $K_n$ , and extrinsic matrix  $E_n = \begin{bmatrix} R_n & t_n \\ \mathbf{0} & 1 \end{bmatrix}$  defined by the camera parameters  $C^{(n)}$  for the  $n^{th}$  camera, to obtain-

$$M_{i,j}^{1 \rightarrow 2} = K_1 \left( R_1 R_2^{-1} \left( K_2^{-1} \left( I'^{(2)}(i, j) \times \begin{bmatrix} j \\ i \\ 1 \end{bmatrix} \right) - t_2 \right) + t_1 \right). \quad (1)$$

With that, we can find the warped feature representation projected into the target view-point. Given some feature representation  $X$ , the general form of the warp function is -

$$\text{Warp}(X; M_{i,j}^{1 \rightarrow 2}) = \begin{cases} \psi(X, M_{i,j}^{1 \rightarrow 2}), & \text{if } M_{i,j}^{1 \rightarrow 2} \in \pi(X), \\ \mathbf{0}, & \text{otherwise.} \end{cases} \quad (2)$$

Where  $\psi(X, M^{1 \rightarrow 2}(i, j))$  represents the bilinear sampling of  $X$  at location  $M_{i,j}^{1 \rightarrow 2}$ , and  $\pi(X)$  denotes the grid of spatial positions of  $X$ . Note that  $X$  can be a feature representation or a depth image.

## Composing $G^{1 \rightarrow 2}$

We compose a grid of transformation related parameters  $G^{1 \rightarrow 2} \in \mathbb{R}^{h \times w \times f'_t}$  and use it in the **GTE** (Section 3.2). The relative pose between  $C^{(1)}$  and  $C^{(2)}$  is  $E_2 E_1^{-1}$ .  $G^{1 \rightarrow 2}(i, j)$  is obtained by the following concatenation -

$$G_{i,j}^{1 \rightarrow 2} = \left[ \text{Warp} \left( I^{(1)}; M^{1 \rightarrow 2} \right) (i, j), \pi(X)(i, j), M_{i,j}^{1 \rightarrow 2}, E_2 E_1^{-1} \right] \quad (3)$$

$f'_t$  becomes 17 by concatenating these values.

## 2 Implementation details

ViewSynth framework jointly penalizes keypoint-descriptor loss  $L_{cm}$ , and view synthesis loss  $L_v$ . We use the joint loss function of  $L_{cm} + \alpha L_v$ , and use  $\alpha = 10$  throughout our experiments. At the start of the training, the backbone network is initialized with Imagenet [10] pre-trained weights of VGG16 [11]. Fully connected and convolutional layers in **VSM** are initialized with Kaiming uniform initializer [12]. We train all parameters of the network using Adam [13] optimizer with an initial learning rate of  $10^{-4}$ , and batch size of 4. The learning is further scheduled to drop by a factor of 10 when it plateaus over 30000 iterations. The training image pairs consist of  $640 \times 480$  resolution depth images for the MSR-7 [14] and TUM [15] datasets, and  $512 \times 424$  resolution depth images for the CoRBS dataset [16]. We use a batch size of 4, and only use depth image pairs that have at least 64 correct correspondences between them, to obtain stable gradient in each iteration. The network is trained until convergence (roughly 60 epochs). We train D2Net [17] and its variants in the same manner. ViewSynth can be trained with only a small time overhead compared to D2Net ( $\sim 1s$  vs  $\sim 0.87s$  per batch on a GTX 1080 GPU) with similar convergence time. Inference time is the same as D2Net since **VSM** is removed then.

Method	MMA <sub>0.1m</sub>
ISS [15] + SHOT [8]	23.0
ISS [15] + FPFH [9]	24.3
Harris3D [16] + FPFH [9]	37.4
Harris3D [16] + SHOT [8]	37.9
Harris3D [16] + 3DMatch [14]	38.2
Georgakis <i>et al.</i> [9]	41.2
D2Net [9]	Collapsed
R2D2 [9]	61.6
R2D2 <sub>s3</sub> [9]	50.1
mD2Net	45.7
D2Net <sub>L<sub>cm</sub></sub>	79.9
ViewSynth (ours)	<b>80.1</b>

Table 1: Keypoint mean matching accuracy (%) on the MSR-7 dataset for 10-frame-apart training, with 0.1m threshold. It demonstrates superior performance of ViewSynth over all baselines.

MMA Threshold	0.1m		0.25m		0.5m		0.1m		0.25m		0.5m		0.1m		0.25m		0.5m	
Frames Apart	10	30	10	30	10	30	10	30	10	30	10	30	10	30	10	30	10	30
Dataset	TUM						CoRBS						MSR-7					
D2Net	Collapsed						Collapsed						Collapsed					
mD2Net	8.72	3.62	20.48	12.60	30.89	21.33	17.10	13.93	29.83	28.13	44.61	42.10	45.69	45.02	61.31	59.55	71.48	69.25
R2D2 [9]	20.84	-	37.34	-	50.59	-	42.08	-	51.26	-	63.43	-	61.55	-	66.30	-	72.58	-
D2Net <sub>L<sub>cm</sub></sub>	33.38	23.93	53.19	45.82	68.93	61.25	56.73	51.53	71.24	66.65	80.35	75.47	79.87	80.35	<b>89.84</b>	90.30	93.30	93.41
ViewSynth (ours)	<b>34.75</b>	<b>35.63</b>	<b>59.45</b>	<b>57.39</b>	<b>77.02</b>	<b>73.65</b>	<b>67.30</b>	<b>52.69</b>	<b>72.43</b>	<b>69.25</b>	<b>81.76</b>	<b>79.16</b>	<b>80.10</b>	<b>80.56</b>	89.70	<b>90.72</b>	<b>93.37</b>	<b>94.19</b>

Table 2: Comparison of MMA on TUM, CoRBS, and MSR-7 datasets, trained on 10/30-frames-apart setting. Acronyms: mD2Net: modified D2Net; D2Net<sub>L<sub>cm</sub></sub>: D2Net with contrastive loss formulation; ViewSynth: D2Net<sub>L<sub>cm</sub></sub> +  $L_v$ , proposed method.

R2D2 [9] originally trains using images of  $192 \times 192$  resolution but the images of MSR-7, TUM and CORBS come in higher resolution. We resize the images in these datasets to  $256 \times 192$  for the R2D2 training to fit them in the GPU memory, while maintaining the aspect ratio of the images, and the image height used in the original paper.

### 3 Additional quantitative results

Table 1 shows that ViewSynth outperforms all competing methods on MSR-7 3D keypoint matching task. Table 2 shows that in different training settings of the MSR-7, TUM and CoRBS dataset, ViewSynth outperforms other baselines in most cases. For the 0.5m threshold, ViewSynth beats the baselines in all settings.

### 4 Additional qualitative results

We present the qualitative comparison between mD2net, D2Net<sub>L<sub>cm</sub></sub>, ViewSynth (D2Net<sub>L<sub>cm</sub></sub> +  $L_v$ ) in pairwise image matching task for MSR-7 [9], TUM RGBD-SLAM [16] and CoRBS [13] datasets. In all cases, the networks are trained on the respective datasets in 30-frames-apart setting. For pairwise matching, we extract 50 keypoints from each image, and match

the keypoints according to their descriptor similarity. We show the qualitative results on the MSR-7 [9] dataset in Figure 2, 3 and 4. Qualitative results on the TUM RGBD-SLAM [12] dataset are in Figure 5, 6, and 7. Qualitative results on the CoRBS [13] dataset are in Figure 8, 9 and 10. We only show the correct keypoint matches between the images. The higher the number of correct matches, the better the keypoint-descriptor set. We also qualitatively show the effectiveness of our proposed View Synthesis Module (VSM) (see Section 3.2 in the original paper) in Figure 11.

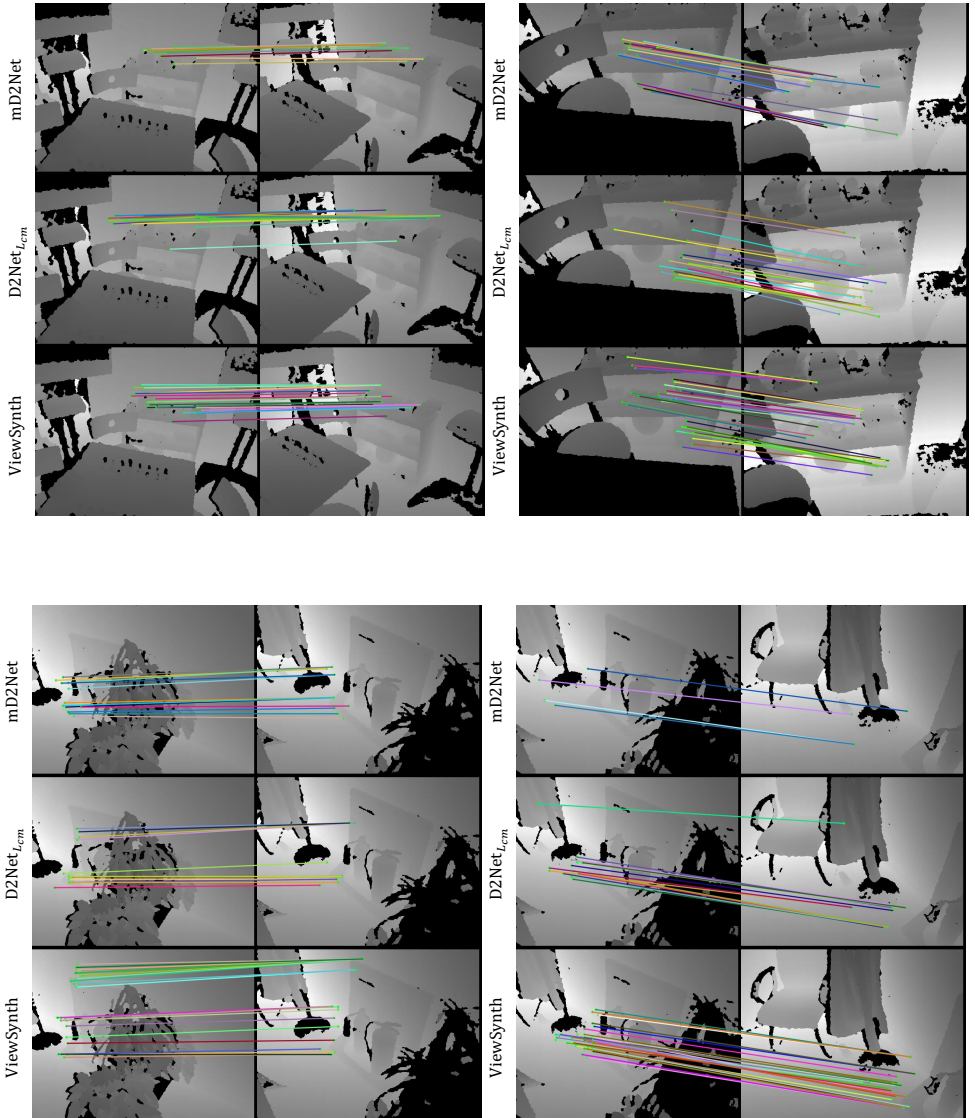


Figure 2: Pairwise keypoint matching on MSR-7 dataset. Top row: mD2Net, middle row: D2Net<sub>L<sub>cm</sub></sub>, bottom row: ViewSynth (D2Net<sub>L<sub>cm</sub></sub> +  $L_v$ ). ViewSynth obtains the highest number of correct matches between image pairs.

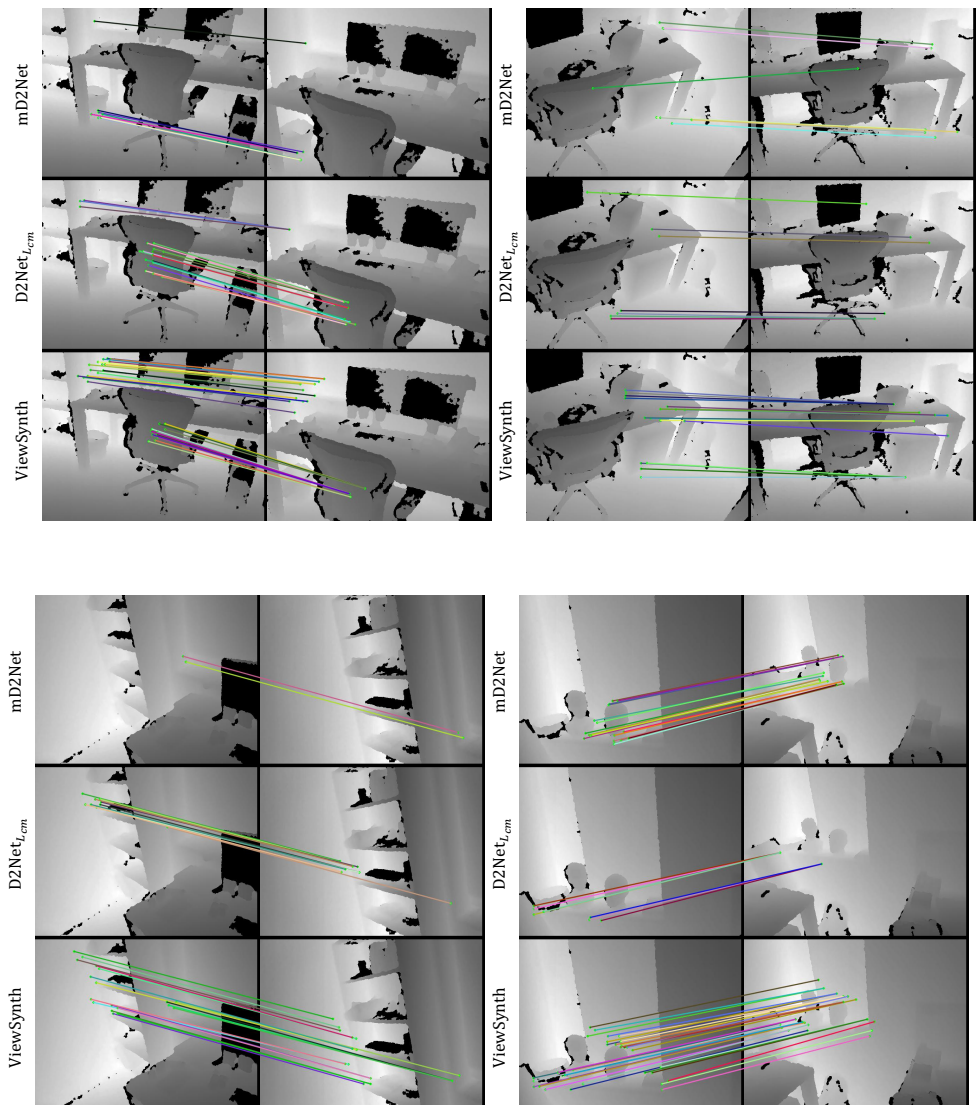


Figure 3: Pairwise keypoint matching on MSR-7 dataset. Top row: mD2Net, middle row: D2Net<sub>L<sub>cm</sub></sub>, bottom row: ViewSynth (D2Net<sub>L<sub>cm</sub></sub> +  $L_v$ ). ViewSynth obtains the highest number of correct matches between image pairs.

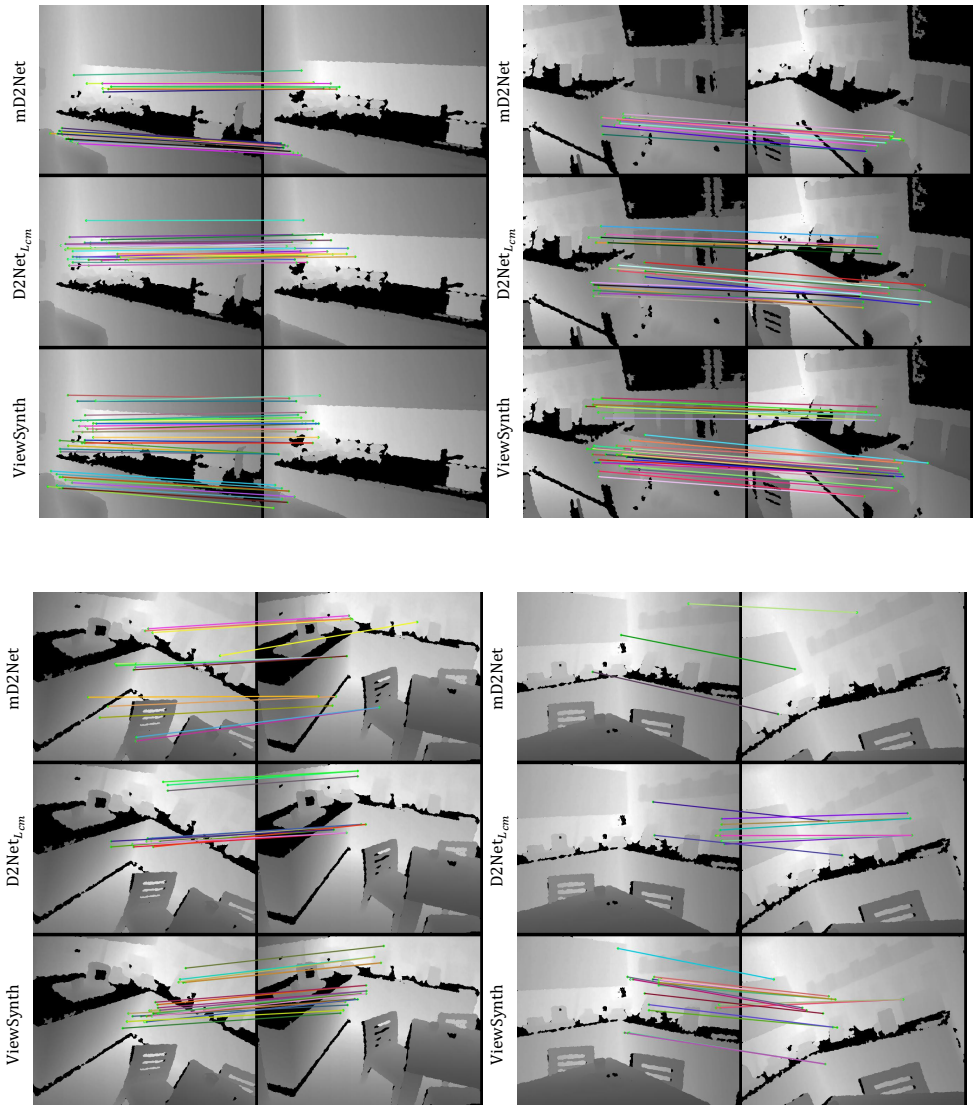


Figure 4: Pairwise keypoint matching on MSR-7 dataset. Top row: mD2Net, middle row: D2Net<sub>L<sub>cm</sub></sub>, bottom row: ViewSynth (D2Net<sub>L<sub>cm</sub></sub> +  $L_v$ ). ViewSynth obtains the highest number of correct matches between image pairs.



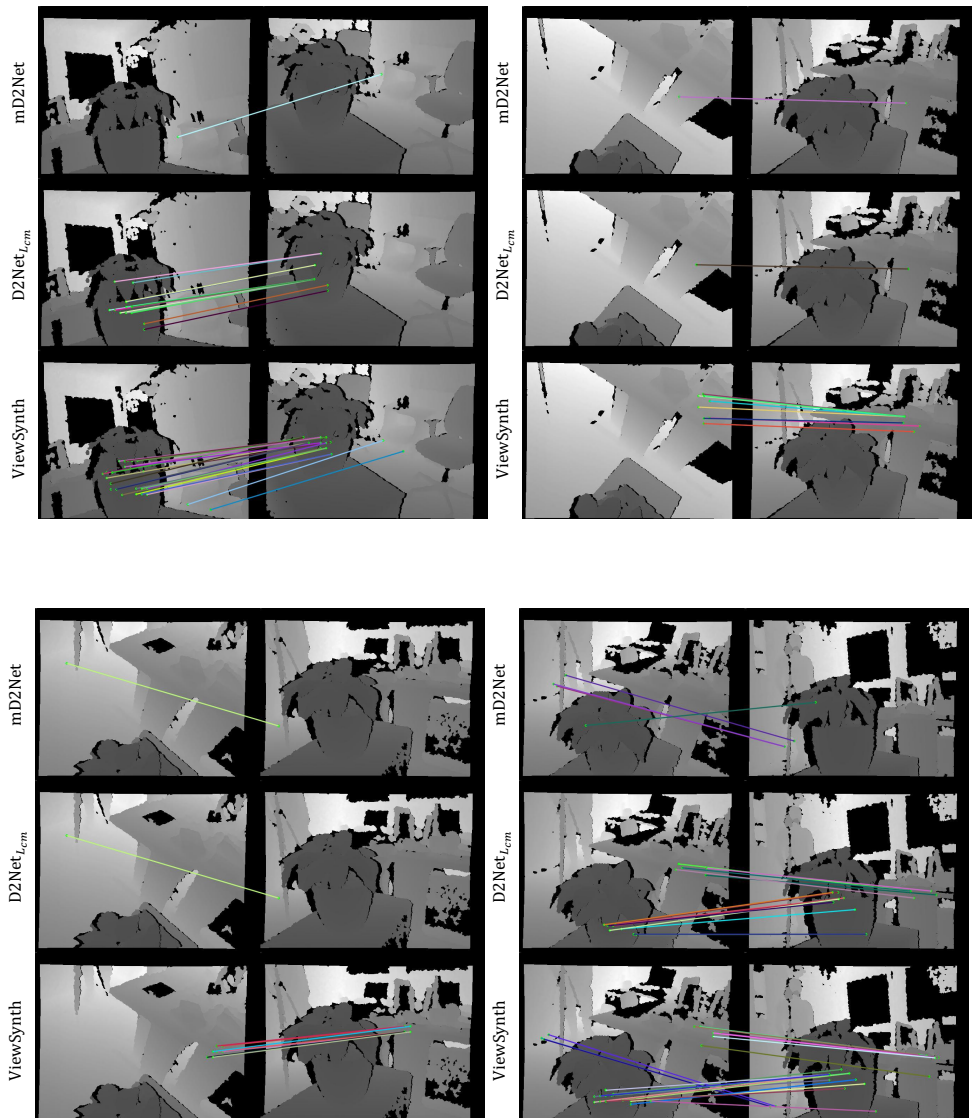


Figure 5: Pairwise keypoint matching on TUM dataset. Top row: mD2Net, middle row: D2Net<sub>L<sub>cm</sub></sub>, bottom row: ViewSynth (D2Net<sub>L<sub>cm</sub></sub> + L<sub>v</sub>). ViewSynth obtains the highest number of correct matches between image pairs.



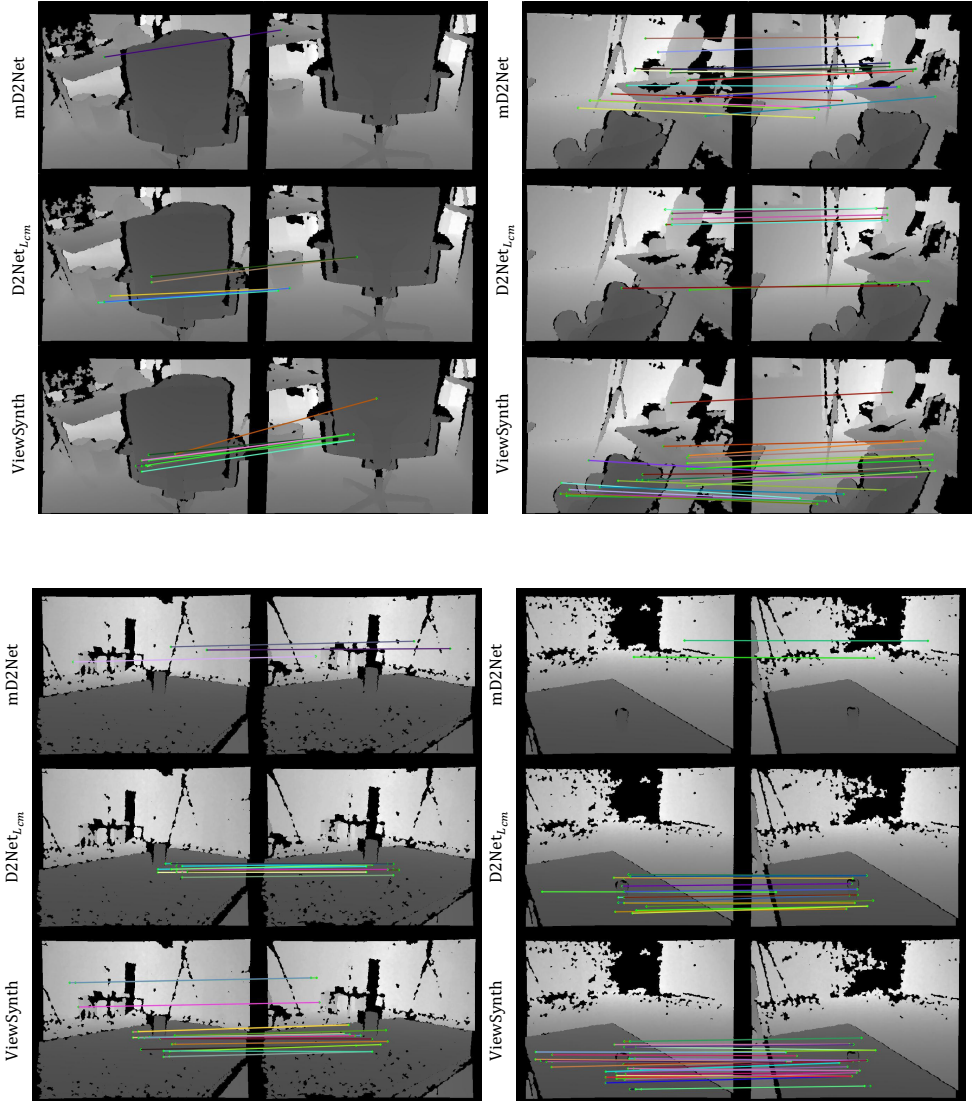


Figure 6: Pairwise keypoint matching on TUM dataset. Top row: mD2Net, middle row: D2Net<sub>L<sub>cm</sub></sub>, bottom row: ViewSynth (D2Net<sub>L<sub>cm</sub></sub> +  $L_v$ ). ViewSynth obtains the highest number of correct matches between image pairs.

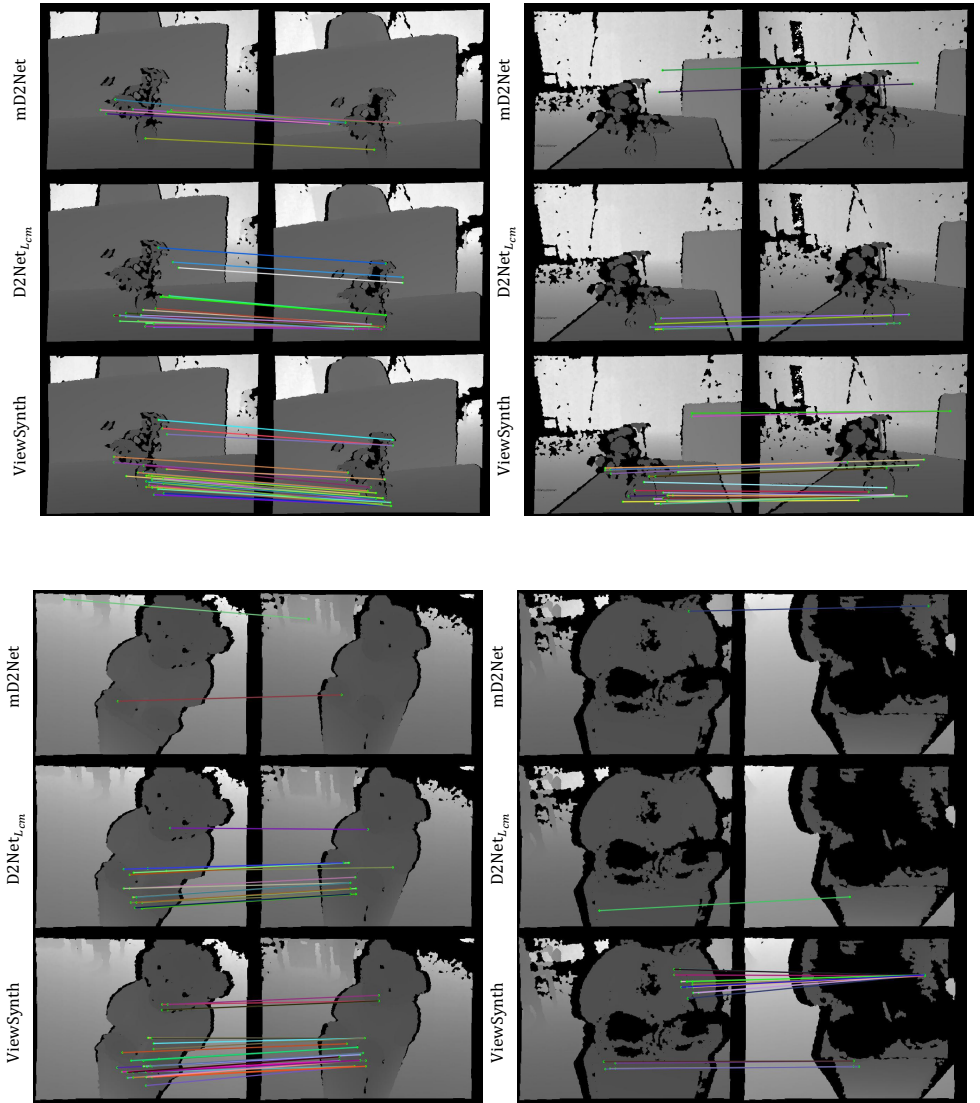


Figure 7: Pairwise keypoint matching on TUM dataset. Top row: mD2Net, middle row: D2Net<sub>L<sub>cm</sub></sub>, bottom row: ViewSynth (D2Net<sub>L<sub>cm</sub></sub> + L<sub>v</sub>). ViewSynth obtains the highest number of correct matches between image pairs.

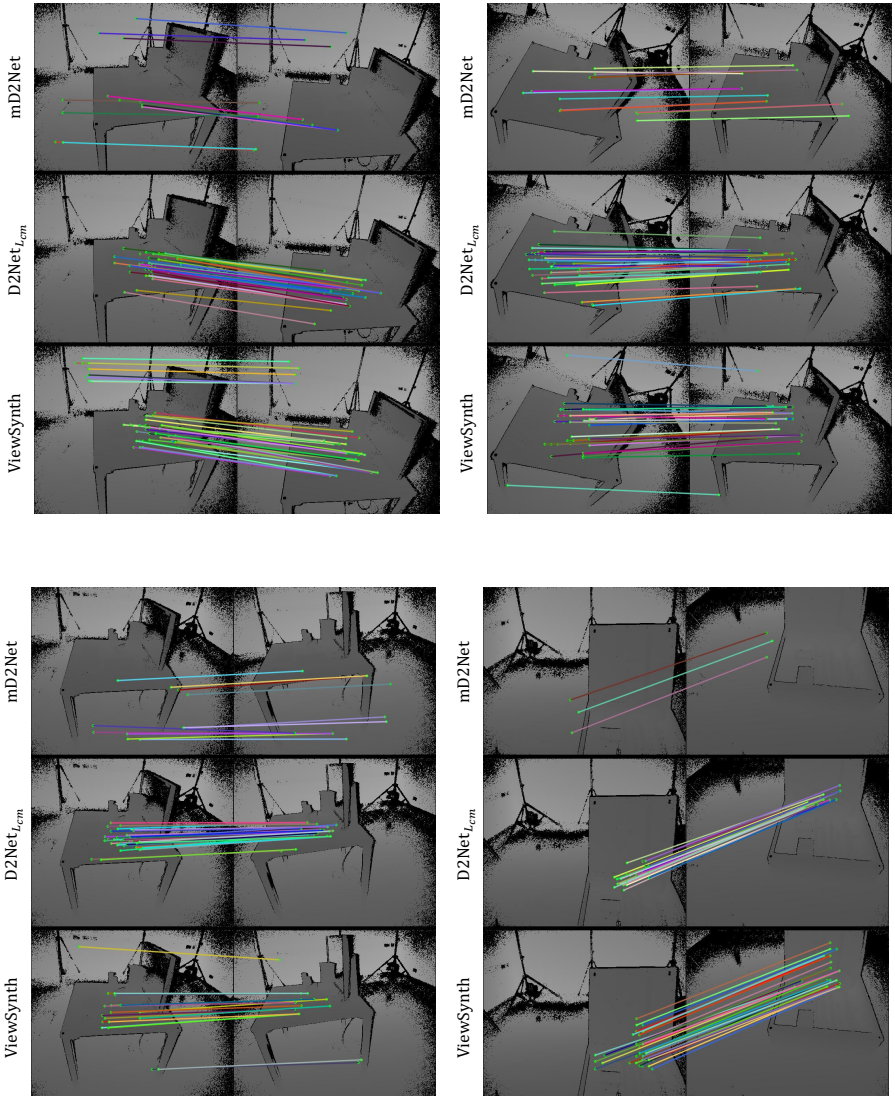


Figure 8: Pairwise keypoint matching on CoRBS dataset. Top row: *mD2Net*, middle row: *D2Net<sub>L<sub>cm</sub></sub>*, bottom row: *ViewSynth* (*D2Net<sub>L<sub>cm</sub></sub>* +  $L_v$ ). *ViewSynth* obtains the highest number of correct matches between image pairs.

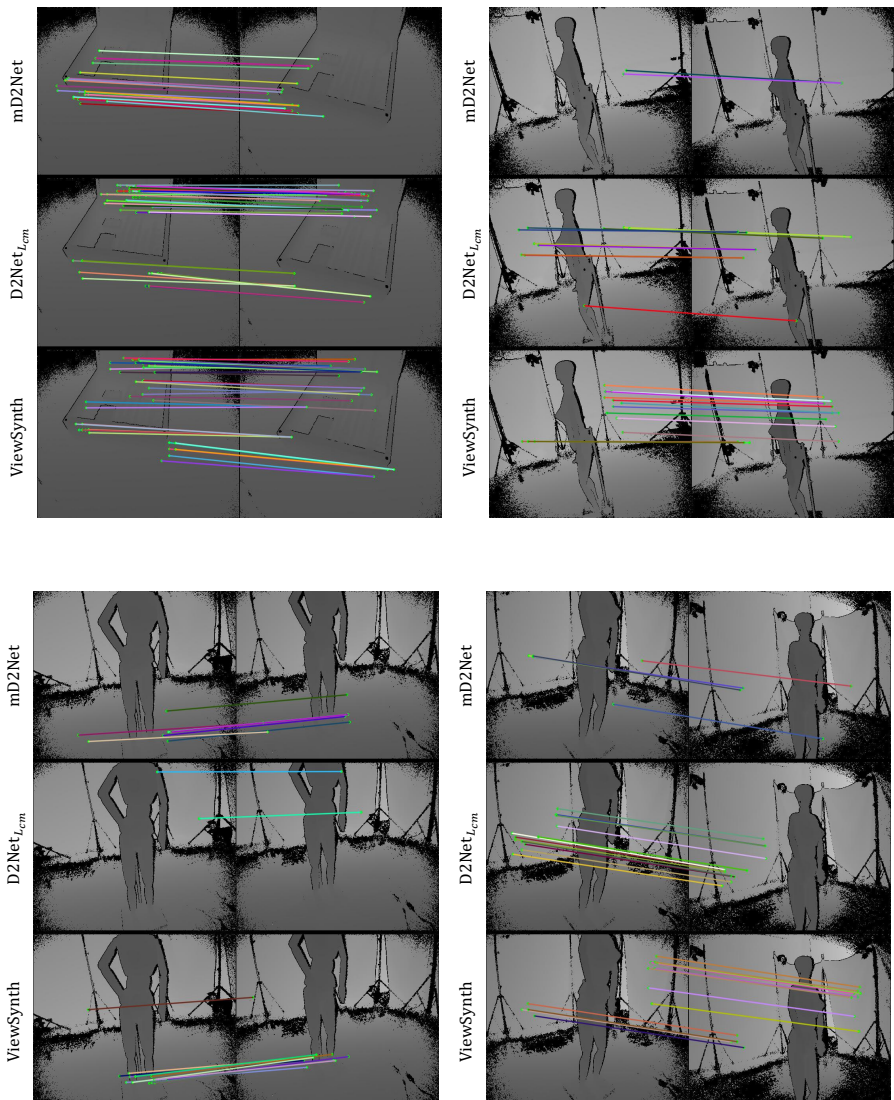


Figure 9: Pairwise keypoint matching on CoRBS dataset. Top row: mD2Net, middle row: D2Net<sub>L<sub>cm</sub></sub>, bottom row: ViewSynth (D2Net<sub>L<sub>cm</sub></sub> +  $L_v$ ). ViewSynth obtains the highest number of correct matches between image pairs.



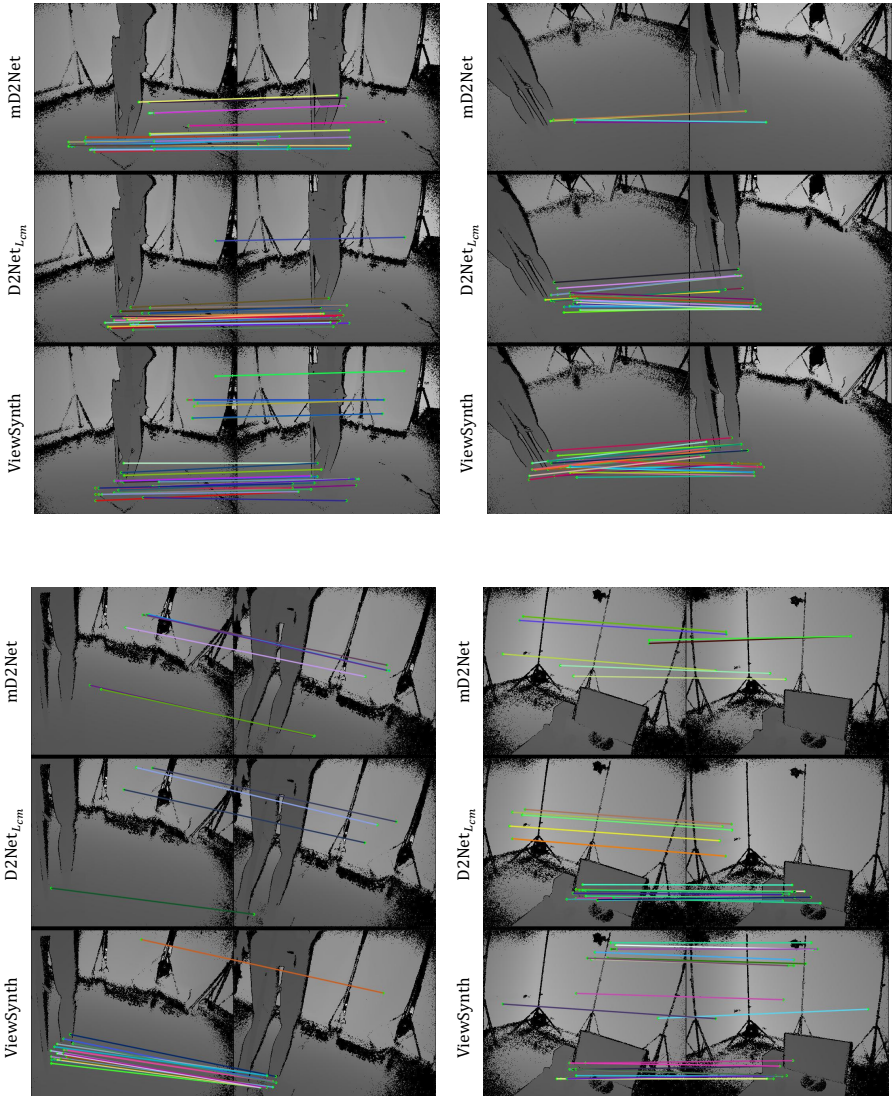


Figure 10: Pairwise keypoint matching on CoRBS dataset. Top row: mD2Net, middle row: D2Net<sub>L<sub>cm</sub></sub>, bottom row: ViewSynth (D2Net<sub>L<sub>cm</sub></sub> +  $L_v$ ). ViewSynth obtains the highest number of correct matches between image pairs.

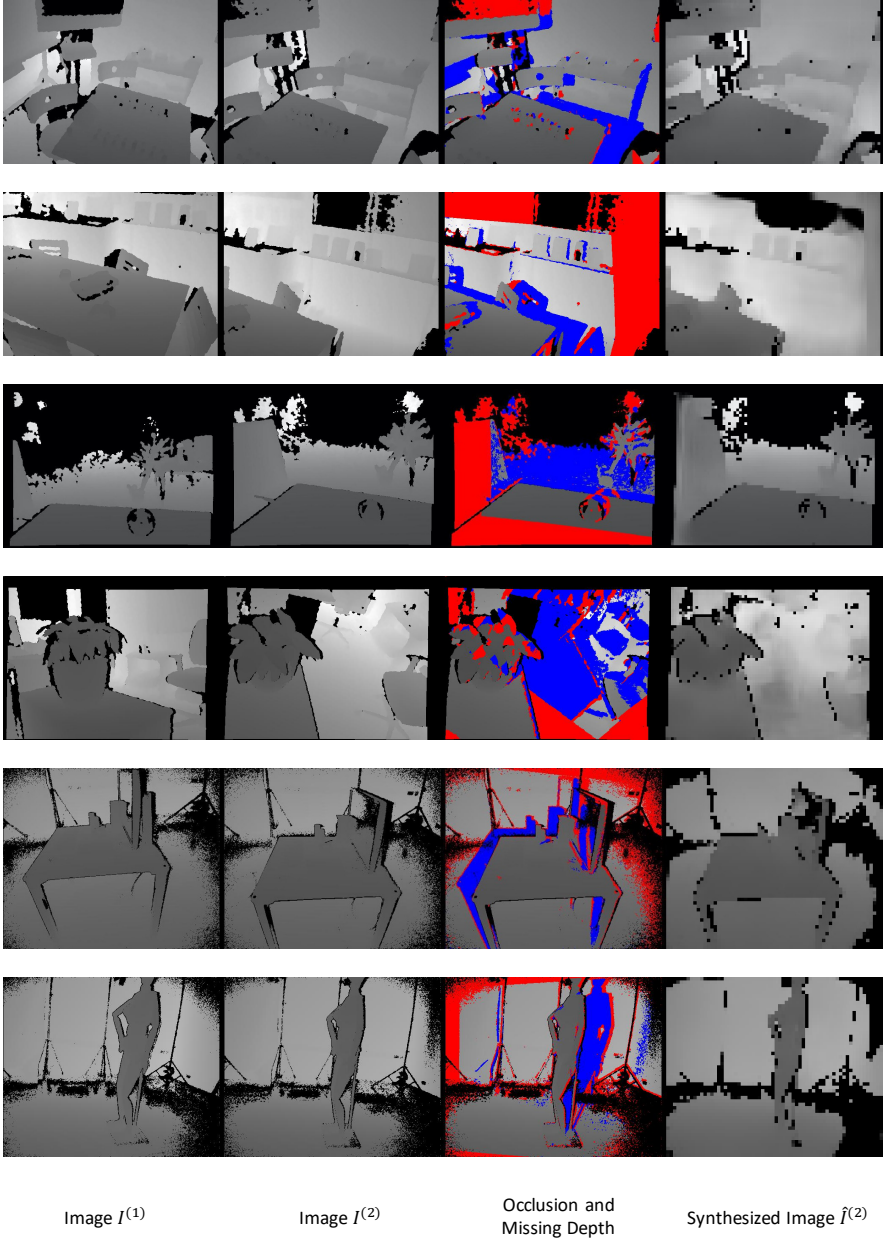


Figure 11: View synthesis examples using our ViewSynth framework. Top two rows: MSR-7, middle two rows: TUM, bottom two rows: CoRBS dataset. **Blue** highlighted area indicates the parts of  $I^{(2)}$  that are occluded in  $I^{(1)}$ . **Red** highlight indicates the change in pose between  $I^{(1)}$ ,  $I^{(2)}$  and the missing information in  $I^{(1)}$ .  $\hat{I}^{(2)}$  shows that **VSM** can synthesize the depth views in the **blue** occluded regions.

## References

- [1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [2] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.
- [3] Georgios Georgakis, Srikrishna Karanam, Ziyang Wu, Jan Ernst, and Jana Košecká. End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1965–1973, 2018.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.
- [5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [6] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yann Cabon, and Martin Humenberger. R2d2: Repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.
- [7] 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.
- [8] S. Salti, F Tombari, and L. Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125: 251 – 264, 2014. ISSN 1077-3142. doi: <https://doi.org/10.1016/j.cviu.2014.04.011>. URL <http://www.sciencedirect.com/science/article/pii/S1077314214000988>.
- [9] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, June 2013. doi: 10.1109/CVPR.2013.377.
- [10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [11] Ivan Sipiran and Benjamin Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963, 2011.
- [12] 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.



- [13] Oliver Wasenmüller, Marcel Meyer, and Didier Stricker. Corbs: Comprehensive rgb-d benchmark for slam using kinect v2. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–7. IEEE, 2016.
- [14] 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.
- [15] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 689–696, Sep. 2009. doi: 10.1109/ICCVW.2009.5457637.