Predicting Sharp and Accurate Occluding Contours in Monocular Depth Estimation using Displacement Maps

Supplementary Material

Michaël Ramamonjisoa* Yuming Du* Vincent Lepetit
LIGM, IMAGINE, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée France
{first.lastname}@enpc.fr https://michaelramamonjisoa.github.io/projects/DisplacementFields

1. Architecture Details

In Fig. 1, we detail the architecture of our network, which is composed of two encoders, one for Depth and an optional one for Guidance. We use a single decoder which combines the respective outputs of the Depth and Guidance decoders using residual blocks and skip connections. We give full details of each block in the following.

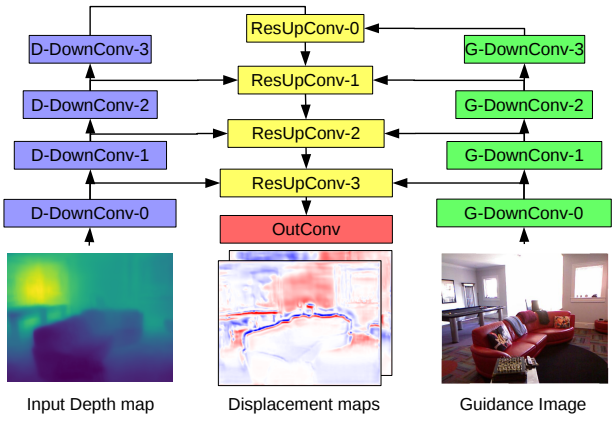

Figure 1. Detailed architecture of our displacement field prediction network. Details on the Depth and Guidance encoders are provided in Sections 1.1 and 1.2 respectively.

1.1. Depth Encoder

Our Depth is a standard encoder with a cascade of four down-convolutions, denoted *D-DownConv*. The D-DownConv blocks are composed of a convolution layer with 3x3 kernel, followed by a 2x2 MaxPooling, and a LeakyReLU [8] activation. The D-DownConv block convolution layers have respectively [32, 64, 128, 256] channels. They all use batch-normalization, are initialized using Xavier [3] initialization and a Leaky ReLU [8] activation.

1.2. Guidance Encoder

Our Guidance encoder is composed of a cascade four of down-convolutions as in [5], which we denote G-DownConv. It is identical to the D-DownConv block described in 1.1 except that it uses simple ReLU [4] activations. and batch normalization for the convolution layers. The convolution layers have respectively [32, 64, 128, 256] channels.

1.3. Displacement Field Decoder

The displacement field decoder is composed of a cascade of four ResUpConv blocks detailed in Fig 1.3.1, and a convolution block OutConv.

1.3.1 ResUpConv

The ResUpConv block is the main component of our decoder. It fuses depth and guidance features at multiple scales using skip connections. The block architecture is detailed in Fig 2. Guidance features are refined using a 3x3 residual convolution layer [5] denoted ResConv3x3. All blocks use batch-normalization, Leaky ReLU [8] activation and filters weights are initialized using Xavier initialization.

1.3.2 Output layers

The final output block *OutConv* is composed of two Conv3x3 layers with batch-normalization and ReLU [4] activation, followed by a simple 3x3 convolution layer without batch-normalization nor activation. The number of channels of those layers are respectively 32, 16, and 2. Weights are initialized using Xavier initialization.

* Authors with equal contribution.

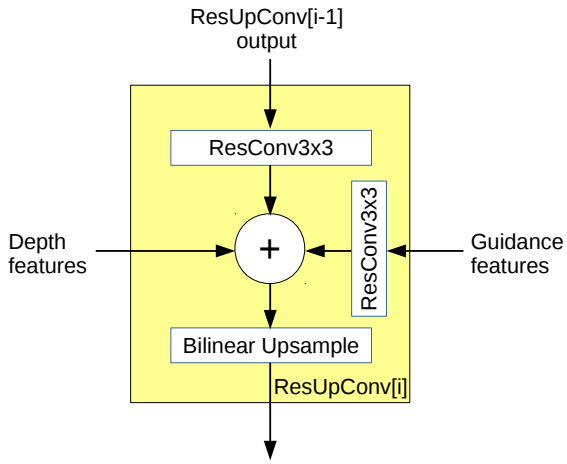

Figure 2. Details of the ResUpConv block used in our displacement field decoder.

2. Qualitative Results

2.1. Comparative Results on 2D Toy Problem

In Fig. 3, we show qualitative results on the 2D Toy Problem described in Section 3.3 of the main paper. One can see that residual update introduces severe artifacts around edges, producing large and spread out errors around them. Constrastingly, our proposed displacement update recovers sharp edges without degrading the rest of the image. To ensure fair comparison between residual and displacement update methods, identical CNN architectures were used for this experiment except for the last layer which predicts a 2-channel output for the displacement field instead of the 1D-channel output residual.

2.2. Comparative Results on NYUv2 Using Different MDE Methods

In Fig 4, we show qualitative results of our proposed refinement method on different MDE methods [1, 7, 2, 9, 11, 6] evaluated in the main paper. Our method always improves the sharpness of initial depth map prediction, without degrading the global depth reconstruction.

3. More NYUv2-OC++ Samples

In Fig. 5 we show several examples of our manually annotated 654 images NYUv2-OC++ dataset, which extends [9]. This dataset is based on the official test split of the popular NYUv2-Depth [10] depth estimation benchmark.

References

[1] D. Eigen, C. Puhrsch, and R. Fergus. Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014. 2, 4

[2] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2, 4

[3] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256, 01 2010. 1

[4] X. Glorot, A. Bordes, and Y. Bengio. Deep Sparse Rectifier Neural Networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[6] J. Jiao, Y. Cao, Y. Song, and R. W. H. Lau. Look Deeper into Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss. In *European Conference on Computer Vision*, 2018. 2, 4

[7] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper Depth Prediction with Fully Convolutional Residual Networks. In *International Conference on 3D Vision*, pages 239–248, 2016. 2, 4

[8] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 1

[9] M. Ramamonjisoa and V. Lepetit. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. *The IEEE International Conference on Computer Vision (ICCV) Workshops*, 2019. 2, 4

[10] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor Segmentation and Support Inference from RGBD Images. In *European Conference on Computer Vision*, 2012. 2, 4, 5

[11] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 4

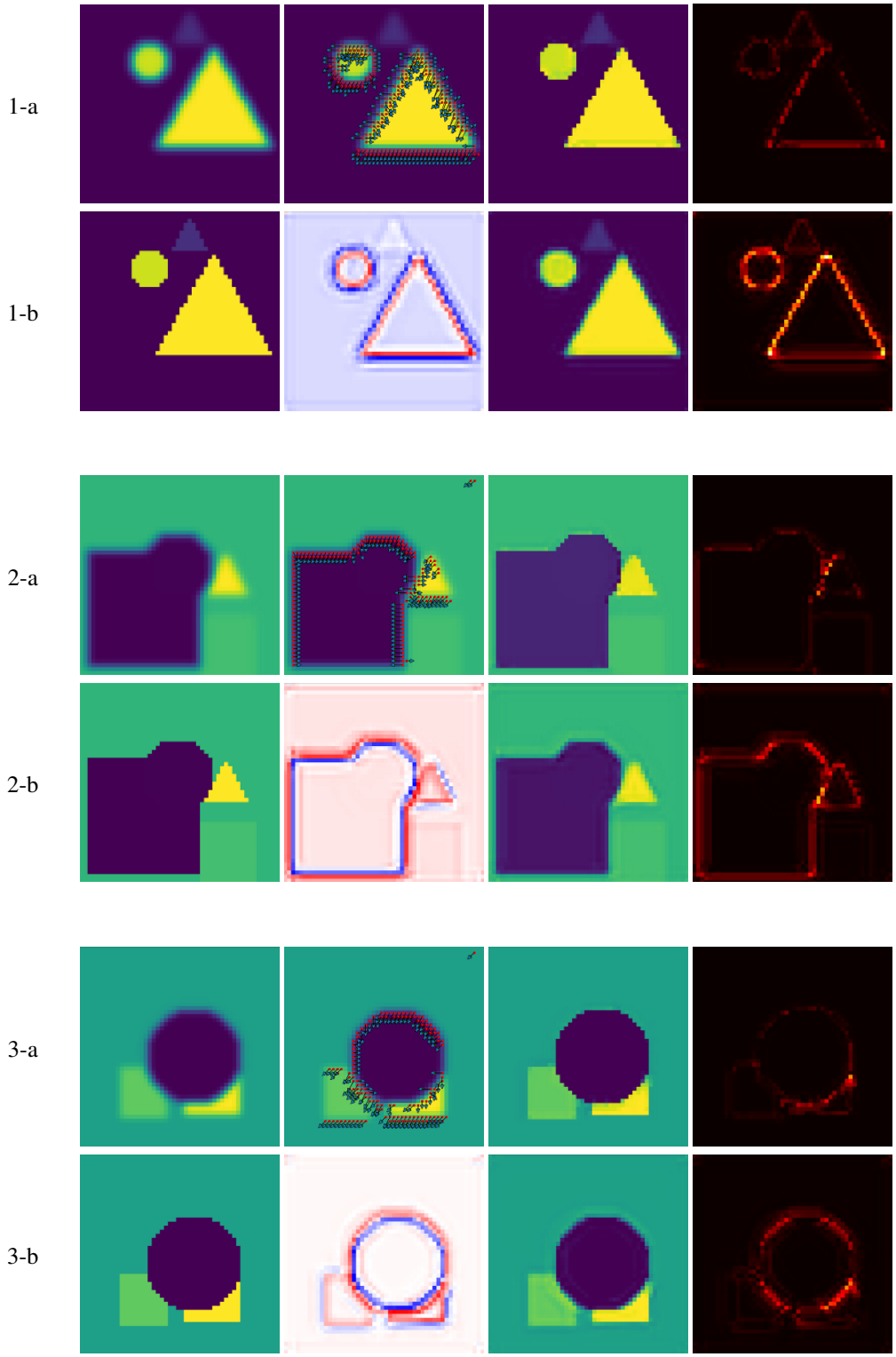

Figure 3. Comparison between residual and displacement learning on a toy image sharpening problem for three examples. A blurred input image $\tilde{I}$ is fed through a Convolutional Neural Network which learns to reconstruct the original clean image I. Lines (1,2,3)-a show from left to right: the input image, samples of the dense predicted displacement field, refinement result with displacement update, error map. Lines (1,2,3)-b show, from left to right: the ground truth image, the predicted residual (blue is negative, red is positive, white is zero), refinement result with residual update, error map. While introducing artifacts, residual learning also results in a spread out error map around edges.

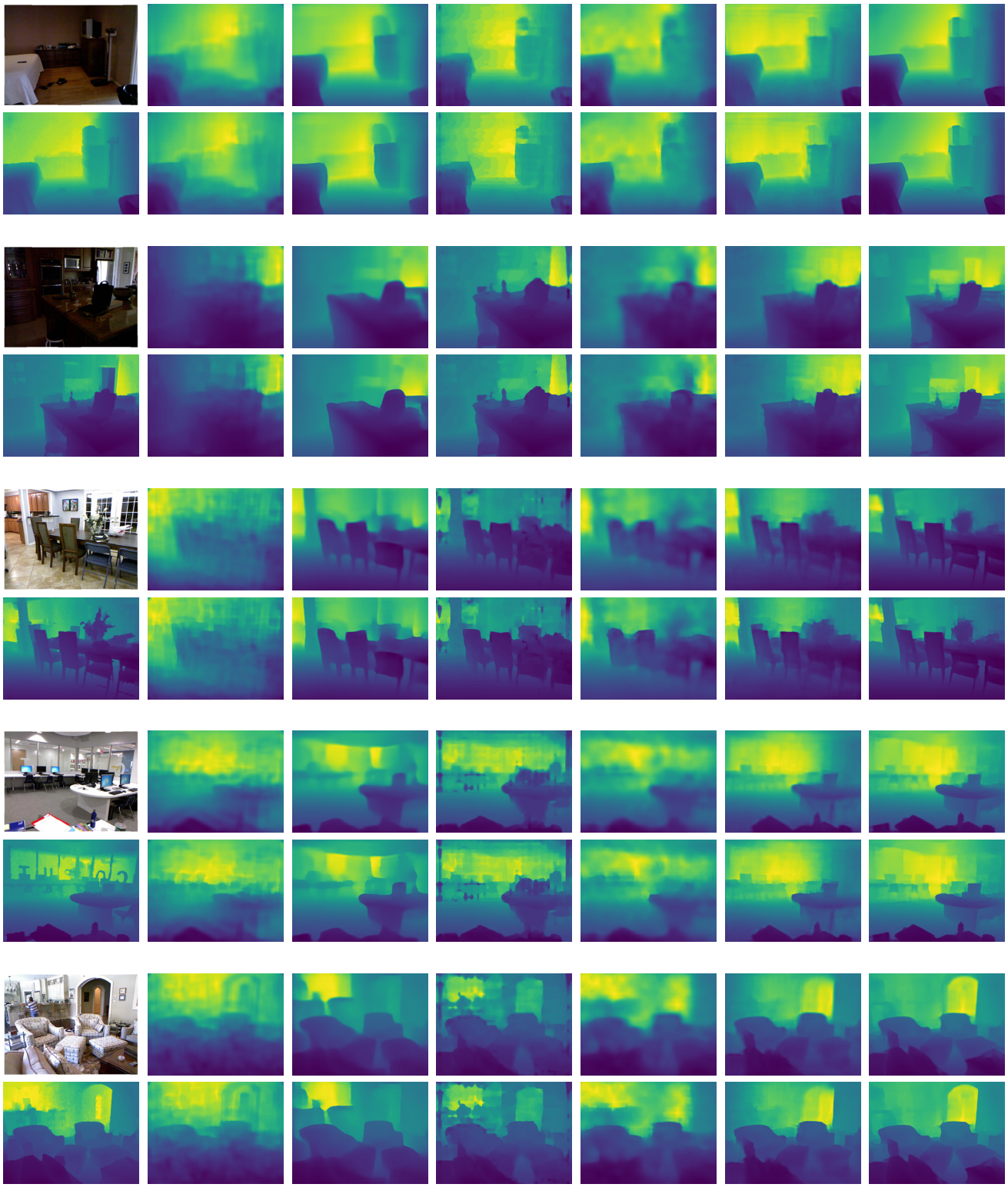

Figure 4. Refinement results using our method (best seen in color). Each example is represented on two rows, first row being the original predicted depth and second row being the refined depth. First column shows the RGB input images and associated ground truth depth from NYUv2 [10]. Following columns are refinement results for different methods we evaluated.

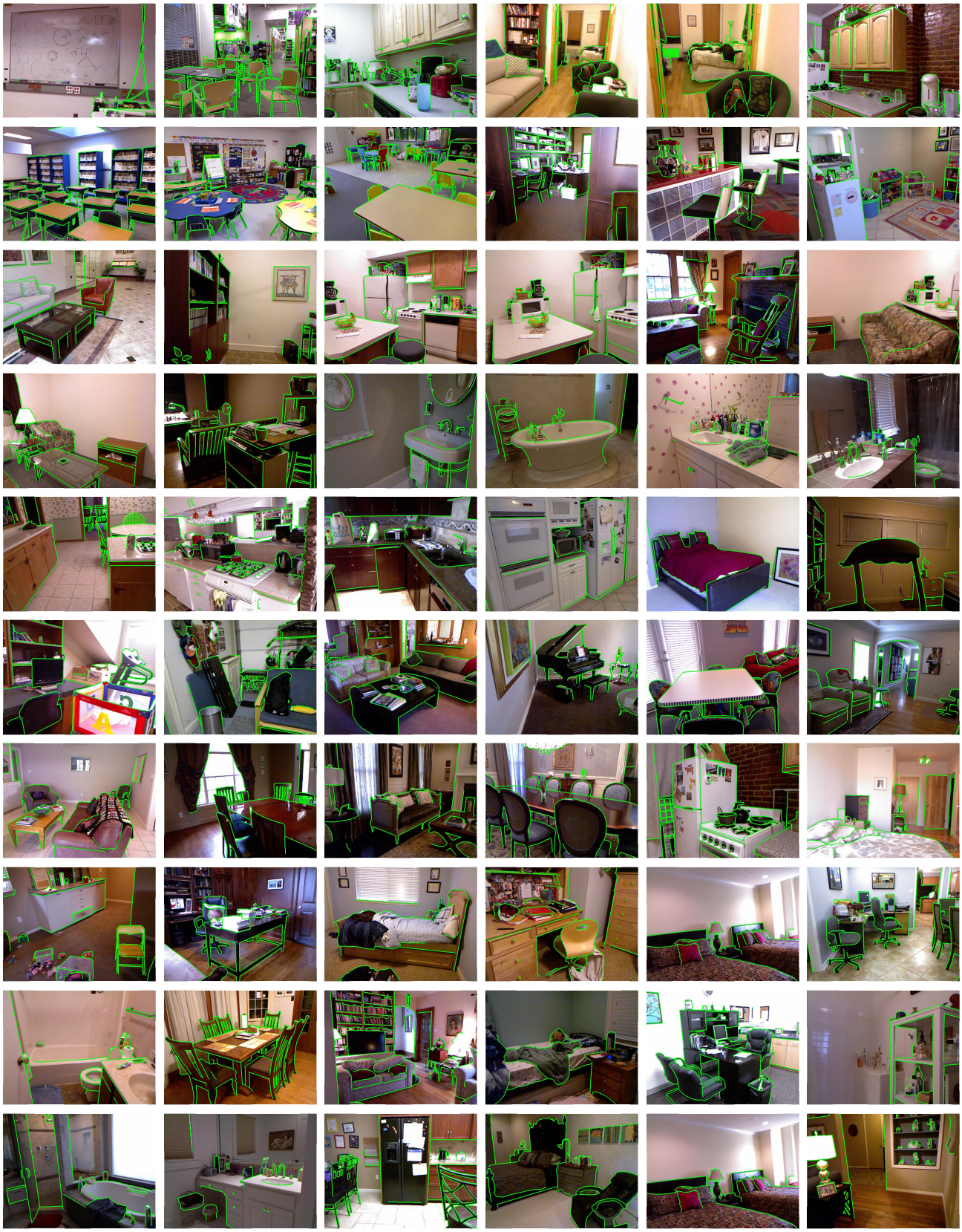

Figure 5. Samples taken from fine-grained manually annotated NYUv2-OC++, which add occlusion boundaries to the popular NYUv2-Depth [10] benchmark. We annotated the full official 654 images test set of NYUv2-Depth.