Supplementary Materials: A Two-Streamed Network for Estimating Fine-Scaled Depth Maps from Single RGB Images

Jun Li[1,2], Reinhard Klein[1] and Angela Yao[1]
[1]University of Bonn, [2]National University of Defense Technology
lij, rk, yao@cs.uni-bonn.de

Network Architecture

We show a sketch of the architecture and detailed information about each layer in Figure 1.

3D Reconstruction Results

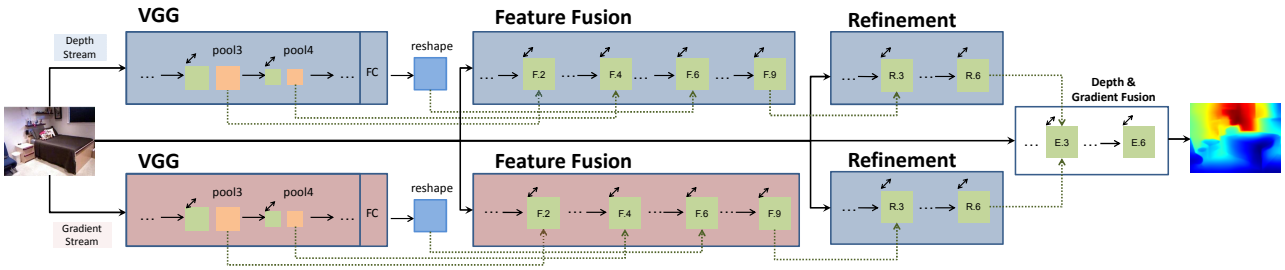
We show examples 3D reconstructions resulting from the depth estimates generated by our proposed method. We sort the 654 test images according to the RMS error and show 20 scenes each with the lowest (Figure 2,3), medium (Figure 4,5) and highest (Figure 6,7) error.

The accuracy of our method according to the RMS error aligns roughly with the depth range in the image. For example, most depths in 20 lowest error scenes are smaller than 6m; the medium error scenes have depths limited to approximately 8m, while the highest error scenes have very large depths which exceed 10m, which is also the limit of the Kinect sensor. However, even in the highest error scenes, we reconstruct plausible looking 3D scenes that preserve the overall structure. Resulting errors are rather associated with inaccuracies in the overall depth scale of the scene.

We compare our two fusion methods to the state-of-the-art ResNet-50 results from Laina *et al*. [1]. The numerical evaluation in [1] reports a higher accuracy than us, but we find their 3D projections to be distorted and suffer from more artifacts. Often, structures are not unidentifiable and the entire reconstructed 3D surface seems to suffer from grid-like artifacts, possibly due to their up-projection methodology. There is little difference in the 3D projections between our two fusion methods; detailing from the optimization are at times a bit sharper than the end-to-end training.

References

[1] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 1



	layers	V.1	V.2	V.3	V.4	V.5	V.6	V.7	skip V.1	skip V.2
VGG	size	113x152	56x76	28x38	14x19	7x9	1x1	55x75	28x38	14x19
	#chan	64	128	256	512	512	4125×D	D	64	64
	ker.sz	3x3	3x3	3x3	3x3	3x3	-	1x1	5x5	5x5
	l.rate	0.0001	0.0001	0.0001	0.0001	0.0001	0.1	0.1	0.01	0.01

	layers	F.1	F.2	F.3	F.4	F.5	F.6	F.7	F.8	F.9
Feature Fusion	size	55x75	55x75	55x75	55x75	55x75	55x75	55x75	55x75	55x75
	#chan	96+64	64	64+64	64	64+D	64	64	64	D
	ker.sz	9x9	5x5	5x5	5x5	5x5	5x5	5x5	5x5	5x5
	l.rate	0.001	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.001

	layers	R.1	R.2	R.3	R.4	R.5	R.6
Refinement	size	111x150	111x150	111x150	111x150	111x150	111x150
	#chan	96	64	64+D	64	64	D
	ker.sz	9x9	5x5	5x5	5x5	5x5	5x5
	l.rate	0.001	0.01	0.01	0.01	0.01	0.001

	layers	E.1	E.2	E.3	E.4	E.5	E.6	E.7
Depth & Gradient Fusion (End-to-End)	#chan	96	64	67	64	64	64	1
	ker.sz	9x9	5x5	5x5	5x5	5x5	5x5	5x5
	l.rate	0.001	0.01	0.01	0.01	0.01	0.01	0.001

Figure 1. Our depth estimation network architecture and layer details. D is the number of output channels in the prediction map; $D = 1$ for depth estimation and $D = 2$ for gradient estimation. *size* is the output map resolution of each layer or layer block. *#chan* is the channel size of output feature map. *ker.sz* is the filter size. *l.rate* is the learning rate.

Figure 2. 20 scenes with the lowest RMS error (average 0.166; part 1).

Figure 3. 20 scenes with lowest RMS error (average 0.166; part 2).

Figure 4. 20 scenes with medium RMS error (average 0.469; part 1).

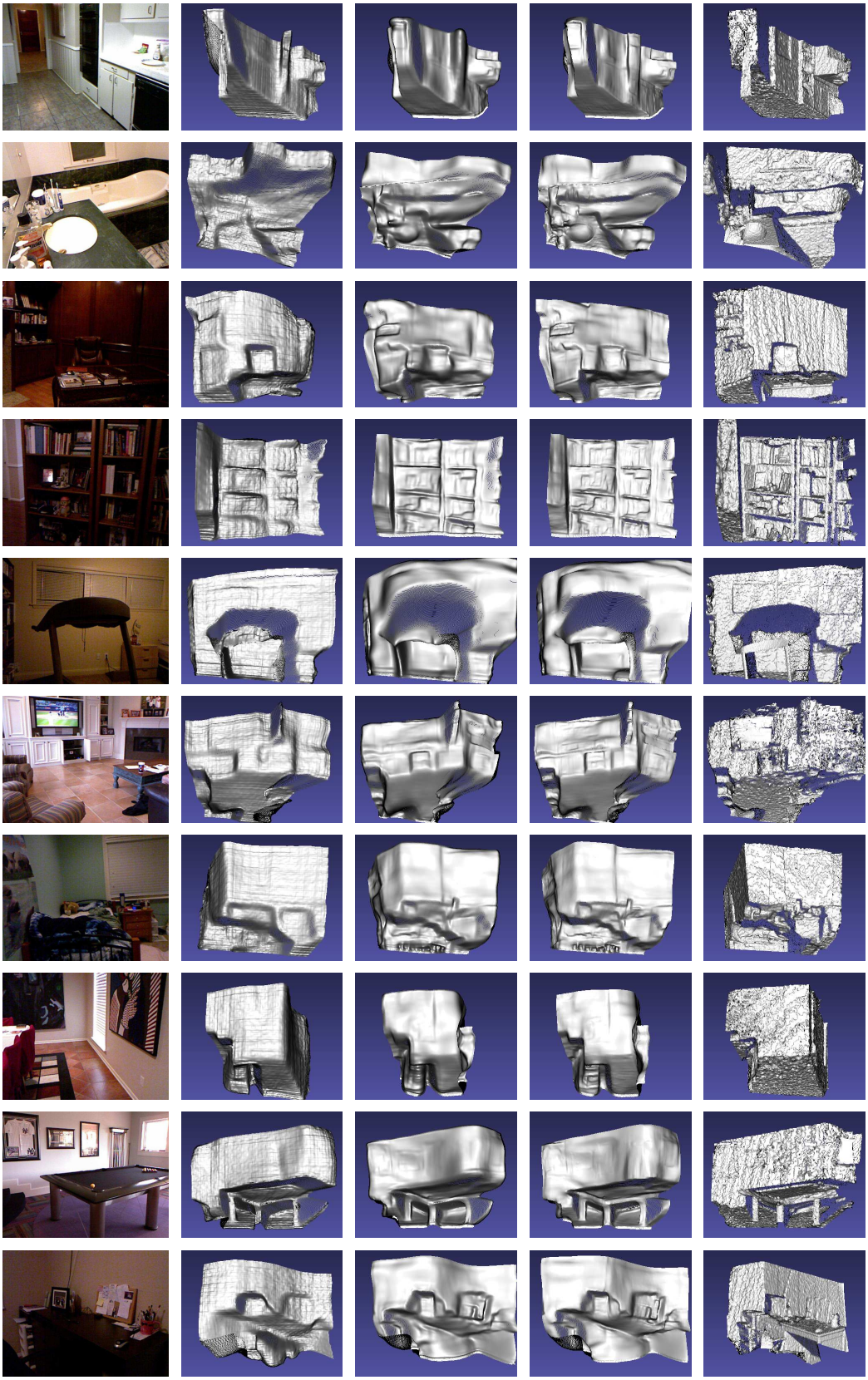


Figure 5. 20 scenes with medium RMS error (average 0.469; part 2).

Figure 6. 20 scenes with highest RMS error (average 1.50; part 1).

Figure 7. 20 scenes with the highest RMS error (average 1.50; part 2).