Supplementary Document for: Deep Non-Line-Of-Sight Reconstruction

Javier Grau Chopite
University of Bonn
jgraucho@uni-bonn.de

Matthias B. Hullin
University of Bonn
hullin@cs.uni-bonn.de

Michael Wand
University of Mainz
wandm@uni-mainz.de

Julian Iseringhausen
University of Bonn
iseringhausen@cs.uni-bonn.de

Abstract

In this document, we provide additional analyses and quantitative evidence to support the discussions in the main paper.

1. Extracting depth maps from volume data

We thank the authors of [3] and [2] for providing code and data. One important observation we made on the provided material is that it can be hard to extract meaningful depth maps from volumetric solutions. In Figure 1, we show as an example the "Bike30" dataset [2] in 64×64 resolution, as reconstructed using the LCT method by O'Toole et al. [3] and f-k migration by Lindell et al. [2] (code provided by authors). The detailed depictions shown in the respective publications have often been cropped to a tight temporal window, without which the solution would be barely visible (Fig. 1).

In our experience, the quality of a depth map of this type of scene mainly hinges on proper foreground-background segmentation. In fact, for some other results contained in these two papers, the authors masked depth data using a ground-truth object silhouette.

For the depth maps for simpler scenes as shown in our main paper, we obtained a segmentation by thresholding a windowed, unfiltered max-intensity projection at 30% of its maximum value. The authors of [2] and [3] approved this in written personal conversation to be a fair representation of their work.

Note that we are including this information not to argue about either method's performance or (dis-)advantages, but merely to illustrate the difficulties in comparing volumetric and depth map-based solutions.

2. Dataset generation parameters

We generated depth maps with our depth scene strategy, i.e FlatNet and ShapeNet, by using the following parameters:

- total number of samples in dataset: 40,000 (36,000 for training)
- number of maximum models in depth scene = 5
- scale factor $0.5 \leq s \leq 1$
- rotations along arbitrary axis in range $-45 \leq \theta \leq 45$
- translation $-0.9 \leq t(x, y, z) \leq 0.9$ from the cubic volume $[-1, 1]^3$

where the parameter choices for transforming and picking the models are based on the criteria that each depth map must exhibit some depth variability (only-background depth should not populate the dataset). We found these numbers to be statistically sufficient for such requirements, but we do not claim they are optimal for training.

On the other hand, obtaining depth maps from the Redwood database required pre-processing choices that we found to have an impact on the final dataset statistics (foreground/background). In this regard, we considered the following heuristics: 1) training dataset should contain little amounts of noise possible, 2) rich class variability. To proceed, we uniformly sampled categories in order to reduce class imbalance. When addressing noise, we remove missing pixels and border artifacts. We found that, in general, this a difficult task to achieve across the entire dataset as captures not only possess different resolutions, but also different noise according to several scene factors (indoors, outdoors, motion, etc). We inpainted regions of invalid depth (depth ≤ 0) using diffusion. Then, we extracted square crops centered at the image center and downsampled them to the network's resolution. Finally, the extracted samples were scaled to match our $[-1, 1]^3$ convention.

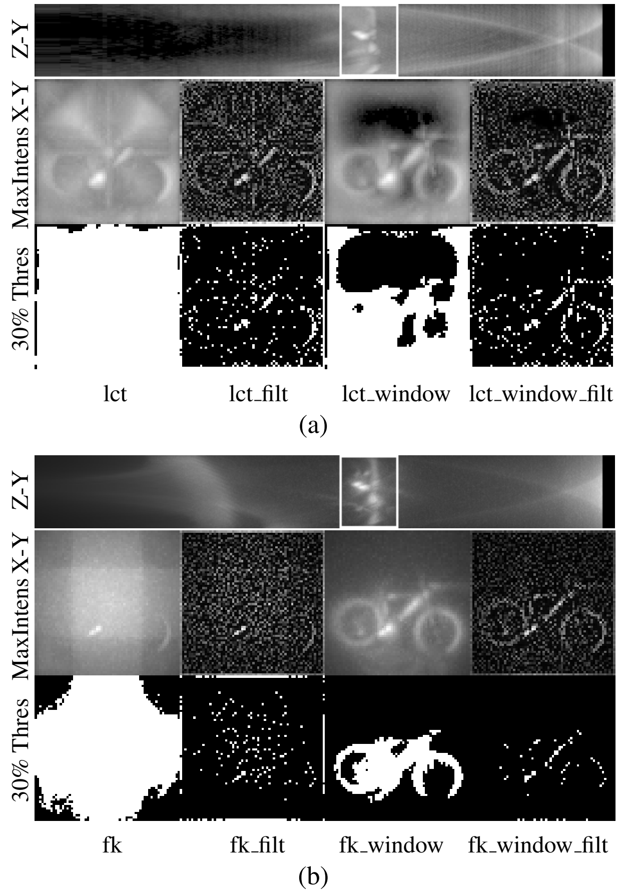

Figure 1. Reconstruction of a downsampled ($64 \times 64 \times 512$) version of the Bike30 dataset [2] using LCT (a) and f-k migration (b), shown in max-intensity projection as proposed by the authors of the respective works. At the top of each group is the reconstructed volume projected into the $z - y$ plane. Only a small portion of the volume (highlighted by a box) contains the target object. From the full volume and a temporally windowed version (suffix "_window"), we have obtained maximum-intensity projections (middle row) and attempted segmenting the object (bottom row) by setting a threshold of 30% on the projected image or its Laplacian-filtered version (suffix "_filt").

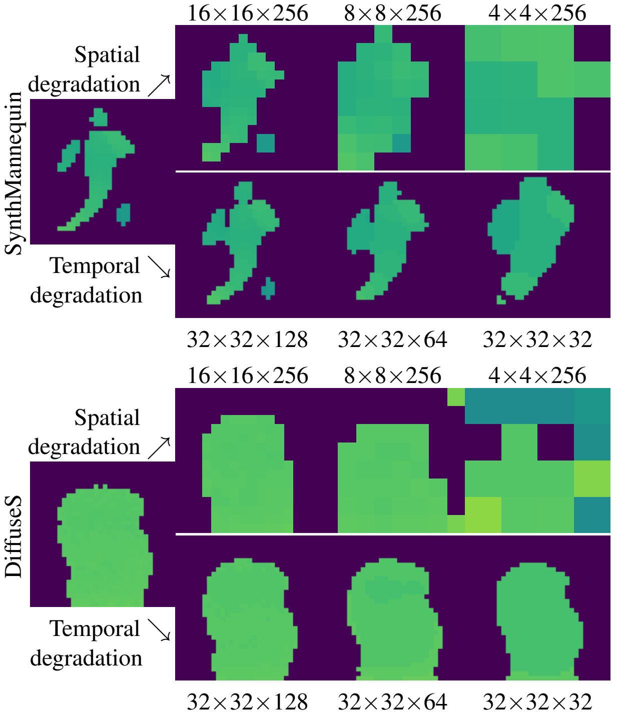

Figure 2. LCT reconstructions [3] on spatially and temporally downsampled data. Shown are a synthetic (top) and a real-world (bottom) example. Full resolution (far left) is $32 \times 32 \times 256$.

3. Spatial/temporal downsampling series for LCT reconstruction

We provide spatial and temporal decimation series for the light cone transform (Figure 2). The coarsening of the spatial dimension immediately maps to the output, since LCT operates at a fixed resolution and cannot upsample. In case of a temporal reduction, details wash out until they become unrecognizable.

4. Evaluating regressor choices

An important component of our pipeline is making architectural decisions suitable for treating the dimensionality mismatch when mapping transient volumes to 2D depth maps. As the target shape is globally encoded in the transient volume, a naive approach would be placing a dense network directly correlating inputs and outputs. This approach results prohibitively expensive as it adds 1 billion parameters for optimization. Hence, we resort to convolutional regressors given their success for segmentation tasks. While we build upon the work from [4, 1] to design our 3D/2D encoder-decoder, we find that numerous choices are available when computing depth values at the output end of the network. Authors in [2] opted for a multi-scale convolutional upsampler with by-pass connections that use the intensity image to alleviate gradient flow. However, for the NLOS case, intensity images computed from the transient volumes contain very little information about the target due to the presence of the diffuse wall. Thus, an upsampler network such as in [2] is not useful.

To build our regressor module, we considered fully-connected layers and traditional convolutional networks. We trained four networks on the ShapeNet dataset with different regressor modules:

- a 1×1 convolution to average incoming depth maps (Conv1×1),
- a 1×1 convolution followed by one dense layers (1-dense),

- a 1×1 convolution followed by two dense layers (2-dense),
- a 2D convolutional network with four layers and filters of decreasing size ((7,7), (5,5), (3,3), (2,2))

We evaluated these experiments by looking at the validation loss (mean squared error) on a test partition of $4,000$ pairs. Table 1 shows the final scores, where we see that dense layers exhibit superior performance than convolutional mechanisms. While 1-dense is slightly better than 2-dense, we chose the latter as it showed better performance on real experimental data.

Conv1×1	1-dense	2-dense	ConvNet
0.0465	0.0415	0.0419	0.0467

Table 1. Mean squared loss performance on validation set for different regressors

5. Supplementary Code and Data

We provide a package consisting of all our input datasets, the full ShapeNet-trained model used across most of the main paper, and basic Python/Keras code to make predictions. Due to its size, we could not upload the package to CMT. Instead, we share it via the Open Science Foundation: `https://osf.io/jmc7p/download`

References

[1] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 2

[2] David B. Lindell, Gordon Wetzstein, and Matthew O'Toole. Wave-based non-line-of-sight imaging using fast f-k migration. *ACM Trans. Graph. (SIGGRAPH)*, 38(4):116, 2019. 1, 2

[3] Matthew O'Toole, David B. Lindell, and Gordon Wetzstein. Confocal non-line-of-sight imaging based on the light-cone transform. *Nature*, 555(25489):338–341, 2018. 1, 2

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 2