Deep Non-Line-Of-Sight Reconstruction

Javier Grau Chopite* Matthias B. Hullin† Michael Wand‡ Julian Iseringhausen§

Abstract

The recent years have seen a surge of interest in methods for imaging beyond the direct line of sight. The most prominent techniques rely on time-resolved optical impulse responses, obtained by illuminating a diffuse wall with an ultrashort light pulse and observing multi-bounce indirect reflections with an ultrafast time-resolved imager. Reconstruction of geometry from such data, however, is a complex non-linear inverse problem that comes with substantial computational demands. In this paper, we employ convolutional feed-forward networks for solving the reconstruction problem efficiently while maintaining good reconstruction quality. Specifically, we devise a tailored autoencoder architecture, trained end-to-end, that maps transient images directly to a depth-map representation. Training is done using a recent, very efficient transient renderer for three-bounce indirect light transport that enables the quick generation of large amounts of training data for the network. We examine the performance of our method on a variety of synthetic and experimental datasets and its dependency on the choice of training data and augmentation strategies, as well as architectural features. We demonstrate that our feed-forward network, even if trained solely on synthetic data, is able to obtain results competitive with previous, model-based optimization methods, while being orders of magnitude faster.

1. Introduction

Over the last decades, the reconstruction of object shapes by means of time-of-flight measurements has matured to a highly pervasive and impactful technology. While it can already be challenging to image objects that are directly visible, recent years have seen researchers successfully demonstrating the even harder task of reconstructing targets hidden beyond the direct line of sight. In this setting, which we call the non-line-of-sight (NLoS) reconstruction problem, the loss of temporal and directional information caused by multiple bounces across more or less diffuse surfaces means that the problem has to be approached in a drastically different manner. Among the techniques proposed in literature for reconstructing object geometry and reflectance from three-bounce indirect measurements are combinatorial approaches [18], parametric model fits using simple, highly efficient forward models [15, 18, 19, 27, 29, 34], backprojection not unlike what is used in computed tomography [20, 35], inverse filtering approaches [12, 28] and space-time diffraction integrals [23, 24]. For the purpose of this paper, we focus on such active-illumination, time-of-flight input despite the fact that steady-state or even purely passive approaches have also been demonstrated [2, 4, 17, 19].

Representation learning methods based on deep neural networks have become a powerful and widely adopted tool for representing complex mappings between high-dimensional (function) spaces of natural signals by learning from examples. In recent work, this includes convolutional regressors for direct time-of-flight imaging [22]. Using scale-space representations with short-cuts across matching scales has turned out to be a particularly successful recipe [31]. Generative convolutional feed-forward networks have also been used for the creation of 3D data, including 2D-to-3D translations [7, 21, 38]. Despite this widespread success, no such approaches have so far been applied to the NLoS reconstruction problem. We assume that this can be attributed to the practical difficulty of generating large amounts of NLoS training data in laboratories or natural scenarios. A key step is therefore the development of a simulator that strikes a good balance between computational cost and physical predictiveness so as to keep the training problem tractable while yielding generalizable results.

In this paper, we build upon a highly efficient renderer that has already been used to implement a relatively slow analysis-by-synthesis scheme for NLoS reconstruction [15]. Here, we use this renderer as a source for large amounts of training data for the NLoS reconstruction problem. We complement this existing forward model with the following additional contributions which are needed to arrive at a full deep learning solution:

- We propose an end-to-end regressor that computes depth maps from 3D space-time impulse responses.
- In order to train our network, we propose sampling strategies to generate a wide gamut of representative NLoS datasets.
- We show that our dataset representation allows for parameter-free target extraction.
- We propose a fast approximate model for real-world time-of-flight histogramming setups that turns the rendered, clean datasets into data that is closer to what can

*University of Bonn / jgraucho@uni-bonn.de
†University of Bonn / hullin@cs.uni-bonn.de
‡University of Mainz / wandm@uni-mainz.de
§University of Bonn / iseringhausen@cs.uni-bonn.de

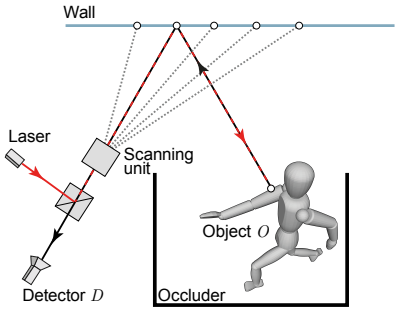

Figure 1. Non-line-of-sight setting. The target O hidden behind an occluder is probed by pointing a laser source toward a wall. Indirect reflections travel through the hidden scene and are then picked up by a sensing device D. The arrangement depicted here is the coaxial setting introduced by O'Toole et al. [28].

be expected from real-world devices. The model, while not as refined and detailed as existing SPAD models [13], can be applied as an augmentation step during training.

- We test the performance of our system on synthetic as well as experimental input data.

Despite the simplicity of its components, we show that our system, while never trained on any real-world input, can make meaningful predictions on experimental data. This holds even when scenes contain retroreflective objects that violate the assumptions of our (purely diffuse) forward model. To our knowledge, this constitutes the first time that a deep learning approach has been successfully demonstrated for the NLoS reconstruction problem, with the additional benefit of being parameter-free. With prediction times of 20 milliseconds, our method is also among the fastest techniques available.

2. Deep Non-Line-of-Sight Reconstruction

The purpose of our work is to devise a deep learning approach to the NLoS reconstruction problem. In this section, we introduce the interfaces to input/output data and use them to motivate a suitable architecture.

2.1. Problem Statement

A typical sensing setup is shown in Fig. 1. A laser source projects a very short pulse of light (often modelled as a Dirac pulse) to a location on a wall, from where it is reflected into the occluded part of the scene and, after interaction with the hidden target, picked up from the wall by a time-resolved detector. We formulate the operation of such a system in terms of the *forward model* T, which comprises the full image formation including light sources, propagation and detection. The non-line-of-sight reconstruction problem relates to an *inverse* mapping, where the measurement I is given and the hidden target O is to be estimated:

$$O = T^{-1}(I). \tag{1}$$

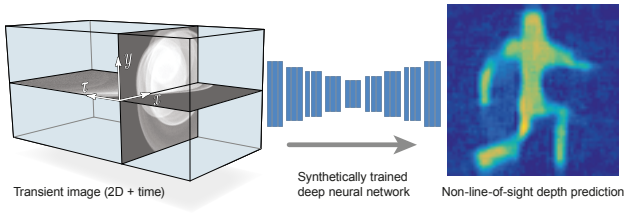

Figure 2. Schematic overview of the reconstruction pipeline. Input is provided in the form of a transient data volume. A convolutional neural network maps this data to a 2D depth map.

This mapping T^{-1} is what we aim to encode in a deep learning approach.

2.2. Input: Transient Image

Depending on the implementation, a setup as described above could produce data in various forms ranging from a timestamped serial stream of individual photon detection events to histograms that are densely sampled in space (x,y) and time (t). For the purpose of this work, we assume a fully sampled space-time cube of time-resolved intensity measurements. Such a 3D volume of intensity data I_{xyt}, also called a *transient image* [33, 35], will therefore serve as the input to our system.

2.3. Output: Predicting Depth Maps

Another central design decision relates to the representation of scene geometry. Among the candidates are volumetric (e.g., a box filled with scattering densities) and surface representations, with the latter offering a more compact encoding of opaque scenes. Since deep learning of generative models using irregularly structured geometry data is still an open problem [14, 1, 26], we decide to represent the scene as a depth map (a 2D map of range values). This choice of *output* comes with the sacrifice of not being able to reflect certain types of self-occlusion [11]. Yet, the representation is intuitively understandable and proves reasonably versatile for a wide range of test scenes as shown, for instance, in a recent optimization framework [34].

2.4. State of the Art in Deep Learning Architectures

Given these design choices regarding the input and output interfaces, the resulting high-level data flow is illustrated in Fig. 2. Modelling such inverse problems using end-to-end trained convolutional regressors has proven successful for a wide variety of computer vision tasks on dense data. In classical vision problems like image segmentation, a 1-to-1 mapping of an input image to per-pixel object class weights is desired. One possible architecture is an autoencoder which compresses the input to fewer degrees of freedom, then unpacks it to the original format [8]. An approach to learning input-to-output mappings consists in tailoring architectural features of such network and train in a supervised way [36]. Such a neural regressor, as we

will use in our approach, is naturally limited to the entropy of the input data. If further details should be hallucinated, generative statistical modeling has to be used, for example through variational autoencoders or generative adversarial networks [9, 30]. We do not follow this path, as we seek objective rather than visually pleasing reconstructions.

Our setting is characterized by a dimensionality mismatch: from a 3-dimensional input (two spatial dimensions plus time of flight), we want to infer a 2-dimensional output (a depth map). While there can be some correlation between spatial dimensions in the input and the output [28], this does not necessarily hold in all sensing geometries. Similar problems have for example been studied in deep network-based reconstruction of 3D images from 2D views, e.g. [7, 8, 21, 32, 38]. Where most prior work tries to infer 3D output from 2D input, our challenge is to go from the 3D transient image to 2D depth map output and hence more closely related to work on SPAD data denoising [22]. While sophisticated translation mechanisms are possible, such as sparse point clouds and differentiable rendering [21], a simple solution is to just use fully connected networks as translators [8]. Our architecture combines convolutional layers with fully connected columns in time with a final decoder for optimal reconstruction quality at reasonable cost. To our knowledge, no prior work has proposed an end-to-end deep NLoS reconstruction pipeline.

2.5. An Architecture for NLoS sensing

Several options are available for retrieving shapes from time-of-flight data (see supplemental material). Given their success in segmentation tasks, we resort to convolutional regressors [31, 6] to build a 3D/2D encoder-decoder network. Our guiding principle consists in extracting spatio-temporal descriptors using 3D convolutions, and subsequently decoding these into 2D features for the further target estimation. In our experiments we found that regressing depth values with convolutional layers provides only coarse estimations of the target. For the direct-line-of-sight setting, authors in [23] successfully used bypass connections that use the intensity image to inform the upsampler network. 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. Our empirical studies showed that dense layers can be a good solution, as they exhibit superior performance (10%) than convolutions at the regression stage. A detailed overview of our deep neural network is shown in Figure 3. We identify three main components within our network, which we motivate as follows:

3D/2D U-Net. At the entry level of the network, we construct an encoder-decoder network that follows the basic design principles of U-Net [31, 6]. On the encoder side, the network contracts the 3D transient image into 2D feature vectors, with the temporal dimension fully contracted at the bottleneck. The decoder upsamples 2D maps from the contracted vectors. Skip connections from the encoder to the decoder enhance gradient propagation and scale separation.

Upsampler network. Once the U-Net has again reached the spatial resolution of the input, we stack an upsampling network that expands incoming maps into higher spatial resolutions. This network employs the same upsampling mechanism as the decoder, namely an up-convolution followed by two 2D convolution + ReLU layers.

Regressor network. When the upsampler has reached a specified target resolution, we stack a regressor network that computes depth images from incoming feature maps. It consists of a 1×1 convolution that contracts incoming channels, and two fully-connected layers.

Throughout the entire pipeline, we use convolutional layers of one channel. We start with convolutions of size $3\times3\times9$ and gradually diminish the temporal size by 2 during encoding. We chose average pooling over max-pooling at this stage as this showed better validation performance on small batches. Upsampling during decoding is done by keeping channels constant while replicating the activation values to the target resolution, followed by a 2×2 (up-)convolution. Finally, skip connections are implemented as average pooling layers along the entire time domain followed by concatenations onto the decoder.

3. NLoS Datasets

Success of a deep learning approach to the NLoS reconstruction problem vitally depends on the availability of large amounts of suitable training data, i.e., pairs of transient images and corresponding depth maps. Collecting such data in the real world can be ruled out, as experimental setups developed so far require considerable amounts of calibration, automation and budget, and setting up thousands of different scenes by hand would be infeasible. We therefore resort to purely synthetic data to train our neural networks. This implies a number of challenges:

- A gamut of possible hidden targets needs to be defined, sampled and cast to depth maps
- For every scene, a transient image must be synthesized in a way that reflects how real-world light transport as well as the real-world measurement devices behave.
- Design and modeling choices should account for efficient dataset synthesis, as large amounts of data are necessary.

3.1. Generating Deep NLoS Datasets

We define a *NLoS dataset* as the collection of pairs $(I^1, O^1), ...(I^N, O^N)$, where O^i represents the depth map of a hidden target and I^i the corresponding transient image. In the following, we describe our sampling and generation strategies to collect such datasets.

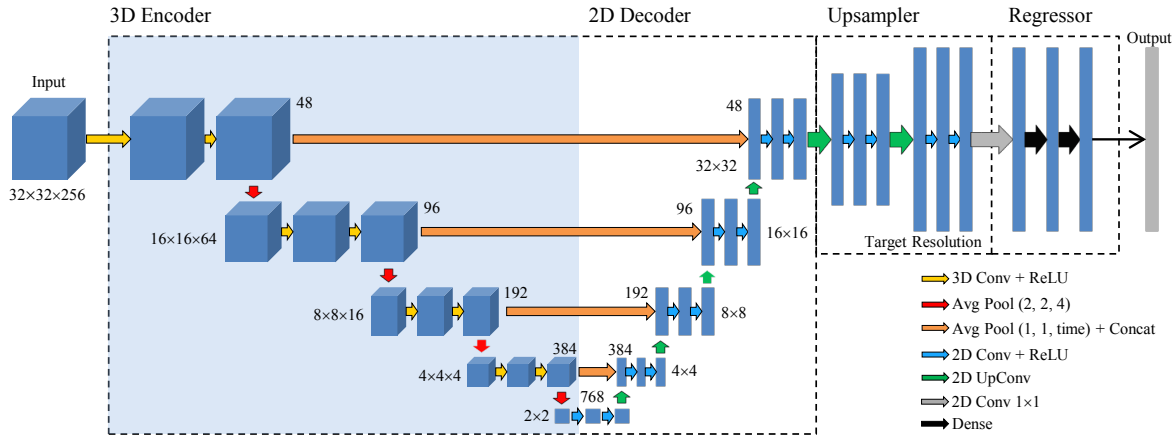

Figure 3. Our functional model for the NLoS reconstruction task. Time-of-flight volumes are analyzed by a 3D→2D encoder-decoder in the first stage. Resulting features are upsampled to a specified target resolution and further decoded into the object's representation, in this case a depth image of the hidden region.

3.1.1 Depth Map Generation

We collect depth maps from two different sources:

- **Depth maps of 3D objects**: given a database of 3D models, we generate a *depth scene* by placing one or more models inside a bounding volume. Then, we obtain the depth map by orthographic projection. Each model is positioned in the scene with a random affine transformation.
- **Samples of measured data**: An alternative approach to generating depth scenes consists in directly sampling depth maps from available RGB-D sequences.

Following either of these options, we generate datasets of target geometries $(O^1, \dots, O^N)$, represented as depth maps with fixed resolution. The depth map is transformed to the bounding volume $[-1, 1]^3$, where $z \leq -1$ represents the background region and $-1 < z \leq 1$ foreground geometry.

3.1.2 Transient rendering

The next step of our data generation consists in rendering transient images of the targets O^i for a fixed sensor D. Several techniques have been developed in recent years for rendering transient light transport [16, 25, 33, 34, 37]. Due to its accuracy and efficiency, we opt for the technique introduced by Iseringhausen et al. [15], which can be adapted to various capture geometries including the coaxial setting [28]. As the renderer accumulates light contributions per triangle of a meshed object, we triangulate our depth samples and feed them to the renderer, where these are further transformed from their native bounding volume to the bounding region of the hidden target. Prior to rendering, triangles with one or more background vertex, as well as triangles that are excessively stretched in z-direction (which form artificial walls connecting objects at different depths), are pruned out. The reflectance of all surfaces is assumed to be diffuse. The resulting transient rendering is stored in the NLoS dataset labeled with its corresponding depth map.

3.2. Sensor Model For NLoS Datasets

The rendering algorithm discussed above does not encompass all effects that should be covered in the image formation model T. Basic physical principles (photon statistics), as well as the hardware used to implement NLoS measurements in the real world, significantly contribute to the quality of the resulting signal. Since most publicly available real-world datasets have been acquired by single photon avalanche detectors (SPADs), we seek to extend our forward model (1) to include the most important features of this technology. While very advanced and accurate models for SPAD sensors exist [13], they require full knowledge of the relevant parameters of the setup and are expensive to evaluate. Since we are interested in feeding networks with tens of thousands of sensor-augmented samples during training, we opt for a strongly simplified model in favor of efficiency. Given a sample $I(x, y, t)$ generated by our renderer, we approximate the SPAD response as

$$I'(x, y, t) = \mathcal{P}(c * I(x, y, t) + b), \tag{2}$$

where c is a scale factor that converts unitless intensity values from the renderer to photon counts. We randomly sample c from an interval that is conservatively chosen by hand and contains the correct factor with high confidence. Including this variation in the training data encourages the regressor to become invariant to global changes in intensity. The global bias b accounts for the base level of dark counts expected in transient measurements [13]. To estimate it, we look at the early temporal bins of real-world measurements that precede the response from the scene. Again, we randomly sample this value from an adequately chosen interval to achieve invariance with respect to this global offset.

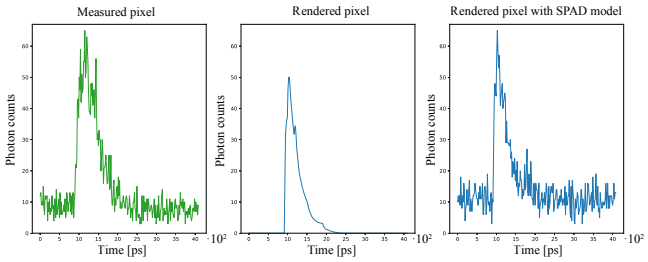

Figure 4. Comparison of a real pixel vs. our SPAD pixel model. Left: A real pixel measured with a SPAD sensor [28]. Center: A rendered pixel computed by using the forward model [15]. Right: Rendered pixel after applying our SPAD model.

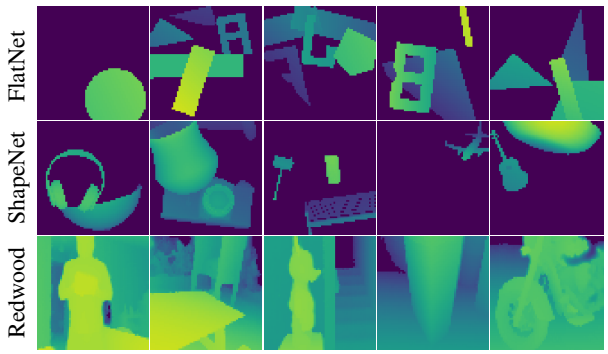

Figure 5. Examples of depth map datasets used for training each three of our models.

Finally, the function $\mathcal{P}$ samples a Poisson distribution to reflect photon counting statistics.

Figure 4 shows how our SPAD model approaches a rendered pixel to an experimental SPAD measurement. Although the model provides a coarse approximation, it captures the overall features of the real sensor while being easy to implement and very fast to compute. This enables its evaluation "on the fly" during training.

3.3. Training strategy

We implement our system in Keras with TensorFlow backend. To train our network, we backpropagate the mean squared difference of depth maps. We experimented with L_1 losses, but preliminary runs showed instabilities during training and/or convergence to poor local minima. Weights are initialized as in [10] and optimized using Adam with default settings and a learning rate of 0.0001. Trainings were performed on Nvidia GeForce RTX 2080 Ti over a runtime of 3–5 days, and models were regularized by means of early stopping. The long training times and the need for regularization are mostly due to the last stage of the pipeline with the large fully-connected layers, which contain most of the parameters. We opted for the (rather simple) early stopping at minimal validation error rather than drop-out due to better performance in our experiments.

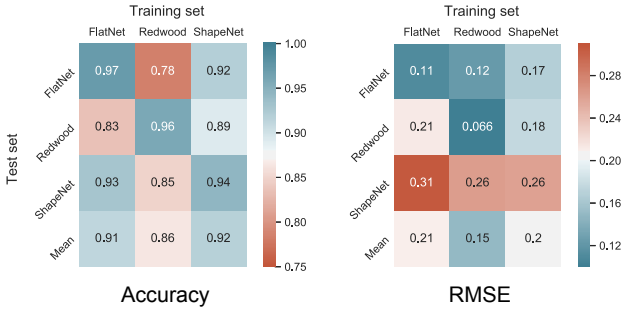

Figure 6. Average accuracy and RMS depth error across datasets. Models trained with synthetically generated depth maps (FlatNet and ShapeNet) exhibit better mean accuracy than the model trained on RGB-D data (Redwood), whereas the latter exhibits better at RMSE scores of the true positive pixels.

4. Results and evaluation

In the following, we evaluate our system on synthetic and real test cases. Input resolution is $32\times32\times256$, with 16 ps time bins. For the case of 64×64 outputs (Sections 4.1–4.5) the average prediction time was 23 ms.

4.1. Training models on synthetic data

We generate three NLoS datasets following the strategies proposed in Section 3.1.1: *FlatNet*, which uses strictly flat shapes as source models; *ShapeNet*, which samples models from the ShapeNetCore database [3]; *Redwood*, which extracts crops from real-world depth maps in the Redwood database [5]. We render 40,000 depth maps for each of these datasets (see Figure 5). Next, for each depth dataset we render the corresponding transient images. Datasets are partitioned into 36,000 training pairs and 4,000 test pairs. The input transient images are augmented using the SPAD sensor model during runtime.

To evaluate the prediction performance we consider two important aspects of the reconstruction task. On one hand, we note that an evaluation metric must account for the quality of foreground/background classification. For the remaining studies of this paper, we use the classification accuracy:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{3}$$

combining rates for missing geometry (false negatives/FN), excess (false positives/FP), correct foreground (true positives/TP) and background (true negatives/TN). In addition, we evaluate the depth error for true positives using a root-mean-square error (RMSE) metric. Figure 6 shows the cross-validated errors on the test partitions of each dataset. Note that, in average, the synthetically-trained models, FlatNet and ShapeNet, perform worst in the depth error sense than Redwood (see RMSE), but exhibit in turn superior classification performance (accuracy). Unlike FlatNet

and ShapeNet (with 71% and 78% background pixels, respectively), Redwood only contains 7% background pixels across the entire database. Such imbalance likely leads to bias and could therefore explain the observed behavior.

4.2. Evaluation on Synthetic Input

In addition to these large-scale datasets, we manually prepare three synthetic test cases: *SynthBunny*, *SynthMannequin* and *SynthDiffuseS*. Figure 7 shows predictions on these synthetic datasets that have been augmented by a plausible amount of sensor noise and bias. Far-away pixels in depth maps are masked as background ($z{=}{-}1$). This feature distinguishes our pipeline from current state-of-the-art methods proposed in [28] or [23], which require an additional threshold and possibly other parameters in order to perform such segmentation. As Figure 8 shows, the model trained on real depth images (Redwood) again tends to overestimate the foreground geometry. When comparing FlatNet with ShapeNet, the former overestimates less geometry for *SynthDiffuseS* and *SynthMannequin* than the latter, but it also misses slightly more. When testing on *SynthBunny*, we observe that ShapeNet captures the overall shape with reasonable RMSE while the FlatNet model misses even basic features of the shape, possibly due to not having been trained on curved shapes.

4.3. Evaluation on Real-World Experimental Input

Having trained and validated our models on synthetic data, we now test the performance of the ShapeNet model on input measured in the real world. We use data provided by O'Toole et al. [28] and Lindell et al. [23], which we downsample to our standard resolution. As baseline for our comparison, we use the light cone transform (LCT) [28] performed on the original datasets. Figure 9 shows how our network successfully retrieves global features of diffuse targets (*DiffuseS* and *Statue*) and, surprisingly, also those of partially specular objects with thin semi-transparent structures (*Bike*). This even holds for retroreflective targets (Figure 10), although our network was trained on purely diffuse input. The performance of our network on retroreflective and otherwise non-Lambertian input indicates that it is robust to mild deviations from the Lambertian ideal.

4.4. Sensor Model Evaluation

We now seek to evaluate our SPAD model, i.e., to assess whether simpler augmentations deliver satisfactory performance on synthetic and real cases. To this end, we trained two additional versions of the ShapeNet model that only include Poisson noise or no augmentation at all ("plain"). Figure 11 shows the impact of our sensor model on noise-degraded synthetic samples as well as on real-world input. While all models are able to predict visually recognizable renditions of *SynthDiffuseS*, they perform poorly for more complicated geometries *SynthBunny* and *SynthMannequin*. Combining Poisson and bias (Eq. 2) clearly brings the greatest benefit. On the real-world dataset *DiffuseS* [28], the difference turns out even more severe.

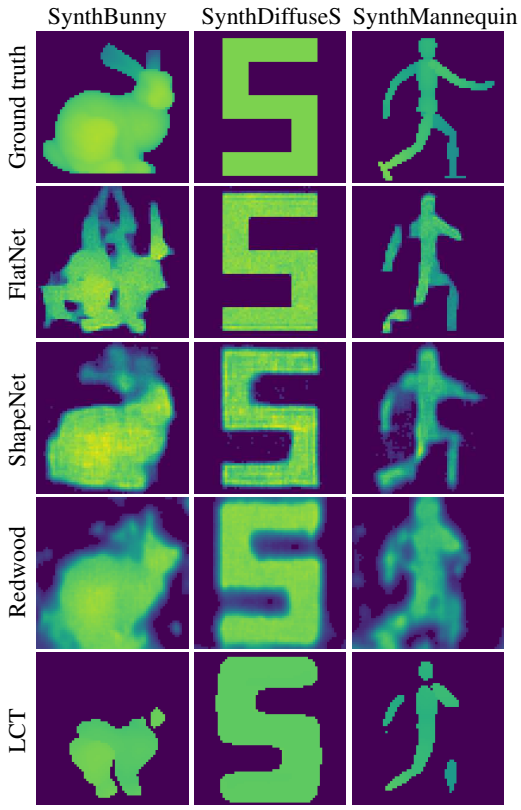

Figure 7. Qualitative visualization of three synthetic depth maps as reconstructed by our trained models and by LCT [28]. Inputs are contaminated with Poisson noise. In average, our models are able to retrieve the overall target shapes, which are comparable to state-of-the-art methods (LCT).

4.5. Extrapolating Real-World Challenges

We evaluate the performance of our pipeline in two adverse scenarios that are of high practical relevance for NLoS reconstruction: low light and low input resolution.

To assess robustness under low-light conditions (light paths involving three diffuse bounces tend to be extremely lossy), we increase the level of Poisson noise on our three synthetic test cases, and feed them to the ShapeNet-trained model (Figure 12). The predictions are stable even under severe noise degradation.

Another interesting aspect of any reconstruction scheme is to determine the critical input resolution that is required to recover a given target. We simulate this situation by block-wise averaging of values in the input data. This is not equivalent to training a suitably dimensioned network at native resolution. Yet, it allows us to use models trained in the full resolution to perform predictions on decimated input. We

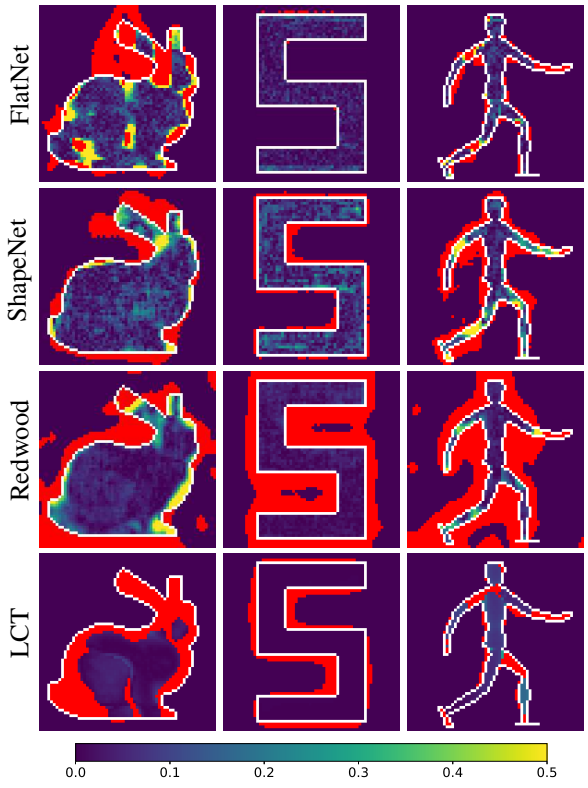

Figure 8. Quantitative assessment of synthetic depth maps from figure 7 in absolute difference values (per pixel) and accuracy. The trained models miss less geometry (red regions inside contour) than LCT, as evidenced in the SynthBunny and SynthMannequin cases, but also tend to overestimate more geometry (red regions outside contour). Although absolute difference values are better in LCT, our models still perform within acceptable margins.

independently decimated the spatial and temporal dimensions by factors of 2, 4 and 8. Figure 13 shows the behavior for a synthetic and a real test case. We note that one level of spatial reduction (factor 2×2) leads to acceptable results, whereas even a fourfold reduction of the temporal resolution yields virtually indistinguishable results for both the synthetic and the real test case. In the supplementary document, we perform a similar series for LCT reconstruction.

4.6. Higher-Resolution Reconstructions

We also explored the question of whether we can train models for higher resolution targets with the proposed pipeline. Like in the previous case, we stack layers until the output size is matched (two, for the case 128×128). As fully-connected layers grow quickly with the output size, using multiple fully-connected regressor layers becomes increasingly expensive. This constitutes an inherent limitation of our architecture. For the specific case of 128×128 outputs, we can still use one fully-connected layer, which retains good quality (see supplementary material). We trained on ShapeNet depth maps of 128×128 while keeping the input resolution unchanged. Figure 14 shows 128×128 predictions performed on the real measurements. By quali-

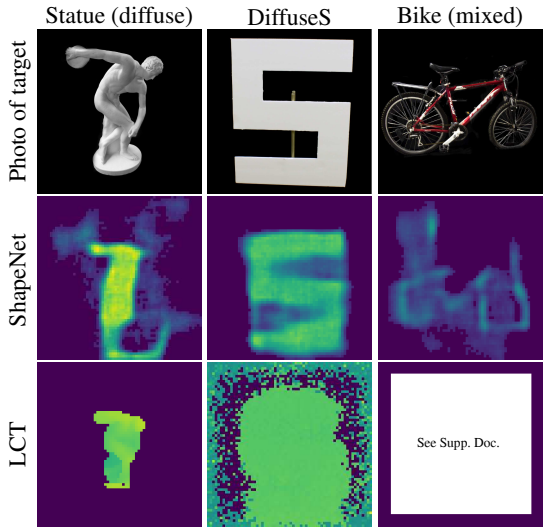

Figure 9. Predictions on three different scenes captured by [28] and [23] using ShapeNet and LCT. Our method allows for depth map segmentation without ambiguity. On the other hand, although LCT is able to better reconstruct complex targets (see *Statue*), extracting depth maps from the resulting volumes for the *Bike* and *DiffuseS* can be hard. The *Bike* scene is partially specular and thus deviates from our our diffuse assumption.

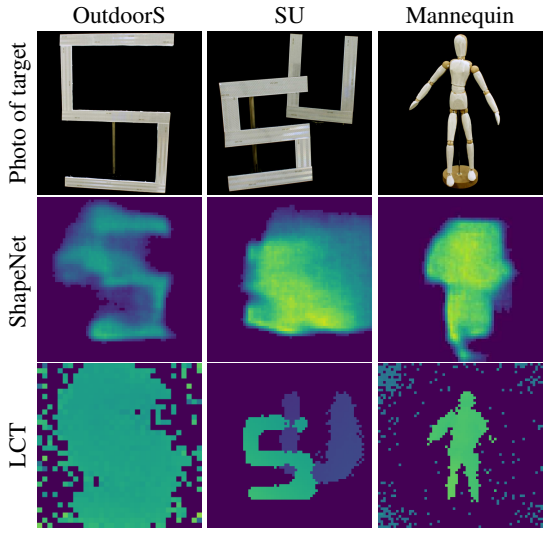

Figure 10. Predictions of three *retroreflective* scenes by O'Toole et al. [28] (top) using ShapeNet (middle row) and LCT (bottom row). Retroreflective targets constitute a failure case of our method. Yet, our model is able to reconstruct the *OutdoorS* target (left) enough to make it more recognizable than LCT. For the *Mannequin* case, although far too coarse, the network still captures global features of the general shape.

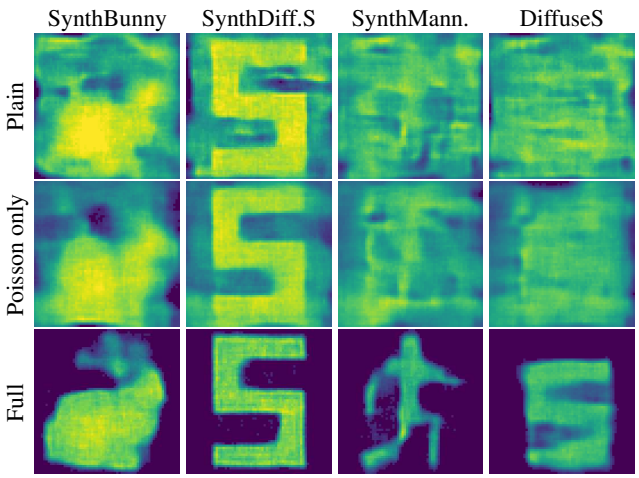

Figure 11. Adding sensor features to the forward model improves performance on noisy synthetic as well as real-world input. Only the full model with Poisson and bias recovers a recognizable depiction of the *DiffuseS* target.

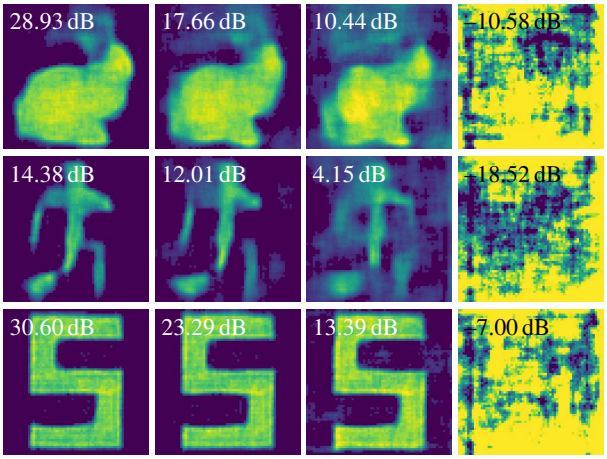

Figure 12. Predictions for input data with high amounts of added Poisson noise. The numbers indicate peak signal-to-noise ratio of the input. Note that, even for extreme levels of noise, our network is still able to retrieve relevant features of the targets.

tatively inspecting these results, we observe that both diffuse and non-diffuse predictions have benefited from these choices.

5. Discussion

In this study we show that deep learning is a promising approach for non-line-of-sight geometry reconstruction. Thanks to our sensor model, our network generalizes to real-world scenes, even though it is trained on purely synthetic data. We are also able to show that our approach performs well for input data with extreme amounts of shot noise. Thanks to its fast runtime, our approach might prove a competitive option for time-critical applications. In future work, we would like to extend our scene representation to three dimensions in order to be able to account for self-occlusion in the scene. Furthermore, it would be interesting to investigate on how to combine multiple measurements using a deep neural network.

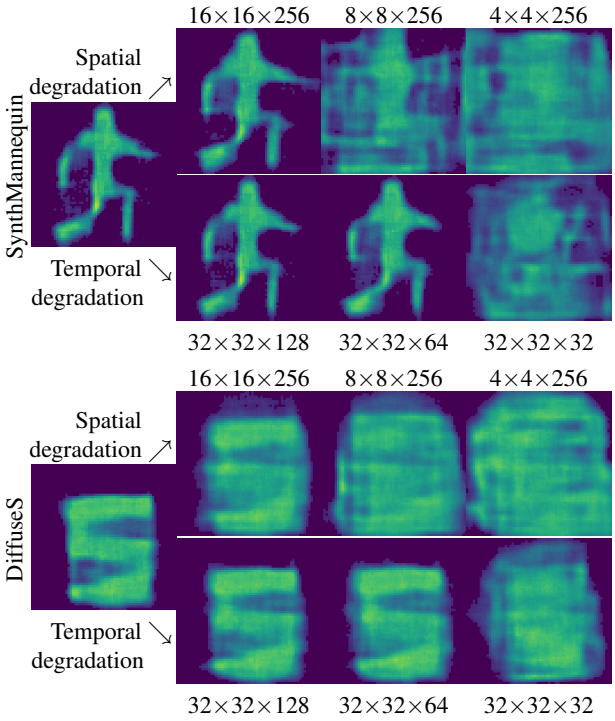

Figure 13. ShapeNet predictions on spatially and temporally downsampled data. Shown are a synthetic (top) and a real-world (bottom) example. In both cases, synthetic and real measurement, the trained model is robust to moderate amounts of decimation.

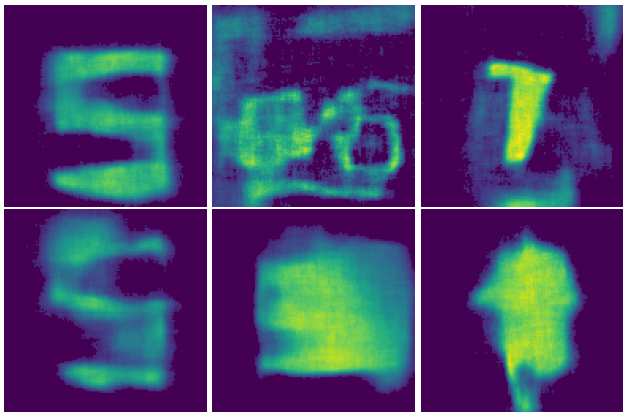

Figure 14. Multiple ShapeNet 128×128 predictions on real measurements. Top: Diffuse targets (*DiffuseS* and *Statue*) and partially specular (*Bike*). Bottom: Retroreflective targets.

Acknowledgements. This work was funded by the European Research Council (ERC Starting Grant "ECHO") and by the German Research Foundation (HU-2273/2-1).

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. 2018. 2

[2] Katherine L Bouman, Vickie Ye, Adam B Yedidia, Frédo Durand, Gregory W Wornell, Antonio Torralba, and William T Freeman. Turning corners into cameras: Principles and methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2270–2278, 2017. 1

[3] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 5

[4] Wenzheng Chen, Simon Daneau, Fahim Mannan, and Felix Heide. Steady-state non-line-of-sight imaging. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[5] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 5

[6] 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. 3

[7] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):692–705, April 2017. 1, 3

[8] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016. 2, 3

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 5

[11] Felix Heide, Matthew O'Toole, Kai Zang, David B. Lindell, Steven Diamond, and Gordon Wetzstein. Non-line-of-sight imaging with partial occluders and surface normals. *ACM Trans. Graph.*, 2019. 2

[12] Felix Heide, Lei Xiao, Wolfgang Heidrich, and Matthias B. Hullin. Diffuse mirrors: 3D reconstruction from diffuse indirect illumination using inexpensive time-of-flight sensors. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[13] Quercus Hernandez, Diego Gutierrez, and Adrian Jarabo. A computational model of a single-photon avalanche diode sensor for transient imaging. *arXiv preprint arXiv:1703.02635*, 2017. 2, 4

[14] Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G. Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Trans. Graph.*, 37(1):6:1–6:14, Nov. 2017. 2

[15] Julian Iseringhausen and Matthias B. Hullin. Non-line-of-sight reconstruction using efficient transient rendering. *ACM Trans. Graph. (TOG)*, 39(1), 2020. 1, 4, 5

[16] Adrian Jarabo, Julio Marco, Adolfo Muñoz, Raul Buisan, Wojciech Jarosz, and Diego Gutierrez. A framework for transient rendering. *ACM Transactions on Graphics (TOG)*, 33(6):177, 2014. 4

[17] Ori Katz, Eran Small, and Yaron Silberberg. Looking around corners and through thin turbid layers in real time with scattered incoherent light. *Nature Photonics*, 6(8):549–553, 2012. 1

[18] A. Kirmani, T. Hutchison, J. Davis, and R. Raskar. Looking around the corner using transient imaging. In *Proc. ICCV*, pages 159–166, 2009. 1

[19] Jonathan Klein, Christoph Peters, Jaime Martín, Martin Laurenzis, and Matthias B. Hullin. Tracking objects outside the line of sight using 2D intensity images. *Scientific Reports*, 6(32491), 2016. 1

[20] Martin Laurenzis and Andreas Velten. Nonline-of-sight laser gated viewing of scattered photons. *Optical Engineering*, 53(2):023102–023102, 2014. 1

[21] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1, 3

[22] David B. Lindell, Matthew O'Toole, and Gordon Wetzstein. Single-Photon 3D Imaging with Deep Sensor Fusion. *ACM Trans. Graph. (SIGGRAPH)*, (4), 2018. 1, 3

[23] 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, 3, 6, 7

[24] Xiaochun Liu, Ibón Guillén, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Adrian Jarabo, Diego Gutierrez, and Andreas Velten. Non-line-of-sight imaging using phasor-field virtual wave optics. *Nature*, 572(7771):620–623, 2019. 1

[25] Julio Marco, Wojciech Jarosz, Diego Gutierrez, and Adrian Jarabo. Transient photon beams. In *Spanish Computer Graphics Conference (CEIG)*. The Eurographics Association, June 2017. 4

[26] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[27] N. Naik, S. Zhao, A. Velten, R. Raskar, and K. Bala. Single view reflectance capture using multiplexed scattering and time-of-flight imaging. *ACM Trans. Graph.*, 30(6):171, 2011. 1

[28] 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, 3, 4, 5, 6, 7

[29] Adithya Kumar Pediredla, Mauro Buttafava, Alberto Tosi, Oliver Cossairt, and Ashok Veeraraghavan. Reconstructing rooms using photon echoes: A plane based model and reconstruction algorithm for looking around the corner. In *Computational Photography (ICCP), 2017 IEEE International Conference on*, pages 1–12. IEEE, 2017. 1

[30] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 1, 3

[32] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proc. CVPR*, 2019. 3

[33] Adam Smith, James Skorupski, and James Davis. Transient rendering. Technical Report UCSC-SOE-08-26, School of Engineering, University of California, Santa Cruz, 2008. 2, 4

[34] Chia-Yin Tsai, Aswin C Sankaranarayanan, and Ioannis Gkioulekas. Beyond volumetric albedo–a surface optimization framework for non-line-of-sight imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1545–1555, 2019. 1, 2, 4

[35] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M.G. Bawendi, and R. Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature Communications*, 3:745, 2012. 1, 2

[36] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016. 2

[37] Di Wu, Andreas Velten, Matthew O'Toole, Belen Masia, Amit Agrawal, Qionghai Dai, and Ramesh Raskar. Decomposing global light transport using time of flight imaging. *International Journal of Computer Vision*, 107(2):123–138, Apr 2014. 4

[38] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 3